

# **Ejecución de un Plan de Mejora de Procesos de Desarrollo en un Entorno Académico**



**Vanessa Agredo Delgado**

**Claudia Sofía Córdoba Collazos**

Director: Esp. Pablo Augusto Magé Imbachí  
Codirector: PhD. Julio Ariel Hurtado

**Universidad del Cauca**

**Facultad de Ingeniería Electrónica y Telecomunicaciones**

**Departamento de Sistemas – Grupo de Investigación IDIS**

**Línea de Investigación Calidad del Software: Producto y Proceso**

**Popayán, Abril de 2013**

## Agradecimientos

En primer lugar agradecerle a Dios por darnos la fuerza y el empeño para culminar esta etapa de nuestras vidas, por cada bendición que permitió continuar e iluminar nuestro camino.

A cada uno de los miembros de nuestras familias, quienes a través del tiempo nos han brindado su apoyo incondicional y su valiosa compañía durante el transcurso necesario para alcanzar cada una de nuestras metas.

También agradecimientos al Esp. Pablo Magé y a nuestro Codirector PhD. Julio Ariel Hurtado por guiar cada una de las actividades necesarias para cumplir con este proyecto, por su dedicación, su tiempo, apoyo y ánimo que permitieron que todo se cumpliera a cabalidad.

A cada uno de nuestros educadores y amigos, quienes acompañaron durante el camino a esta meta con su apoyo, colaboración y solidaridad, que con sus conocimientos compartidos dejaron huella en nosotros y en la elaboración de este proyecto.

Para finalizar, nuestros agradecimientos a la Universidad del Cauca, institución que nos forjó como personas, brindándonos la oportunidad de realizar nuestros estudios de pregrado.

# Tabla de Contenido

<i>Agradecimientos</i> .....	<i>I</i>
<i>Tabla de Contenido</i> .....	<i>II</i>
<i>Lista de Figuras</i> .....	<i>VIII</i>
<i>Lista de Tablas</i> .....	<i>IX</i>
<b>Presentación</b> .....	<b>1</b>
<b>Capítulo 1 – Introducción</b> .....	<b>4</b>
1.1 PLANTEAMIENTO DEL PROBLEMA .....	4
1.2 OBJETIVOS .....	6
1.2.1 Objetivo General .....	6
1.2.2 Objetivos Específicos.....	6
1.3 RESULTADOS OBTENIDOS .....	6
<b>Capítulo 2 – Contexto Teórico</b> .....	<b>8</b>
2.1 DEFINICIONES / CONCEPTOS.....	8
2.1.1 PYMES.....	8
2.1.2 VSE (Very Small Entity) .....	8
2.2 ESTANDARES .....	9
2.2.1 ISO 12207 .....	9
2.2.2 ISO 15504 .....	10
2.2.3 ISO 29110 .....	11
2.2.4 CMMI – DEV .....	11
2.3 MODELOS RELACIONADOS.....	12
2.3.1 AGILE SPI – PROCESS .....	12
2.3.2 SIMEP – SW (Sistema Integral para la Mejora de los Procesos Software en Colombia)	
15	

2.3.3	MOPROSOFT (Modelo de Procesos para la Industria del Software).....	15
2.3.4	COMPETISOFT .....	16
2.4	TRABAJOS RELACIONADOS.....	18
2.4.1	Modelado y Propuesta de Mejora de Procesos de Desarrollo para un entorno académico .....	18
2.4.2	Aplicación de modelado de software y mejora de procesos en el ámbito académico. 18	
2.4.3	Implementación del Proceso Personal Software en un Ámbito académico .....	19
2.4.4	Una experiencia de mejora de procesos software aplicada a la educación .....	19
2.4.5	MUM - Proceso de Desarrollo de Software Modularizado, Unificado y Medible.....	20
2.4.6	Aplicación del modelo de procesos en un curso de Ingeniería del Software .....	21
2.4.7	Programa de mejora UAMISoft .....	22
2.5	ANÁLISIS DE TRABAJOS RELACIONADOS .....	22
<b>Capítulo 3 – Propuesta .....</b>		<b>23</b>
3.1	AGILE SPI-PROCESS PARA EL CONTEXTO ACADÉMICO: PROCEDIMIENTO FORMAL.....	23
3.1.1	Fases .....	24
3.1.1.1	Fase 1: Instalación .....	24
3.1.1.1.1	Actividades .....	24
3.1.1.2	Fase 2: Diagnóstico.....	25
3.1.1.2.1	Actividades .....	26
3.1.1.3	Fase 3: Formulación .....	27
3.1.1.3.1	Actividades .....	28
3.1.1.4	Fase 4: Mejora.....	28
3.1.1.4.1	Actividades .....	29
3.1.1.5	Fase 5: Revisión .....	29
3.1.1.5.1	Actividades .....	29
3.1.2	Disciplinas.....	30
3.1.2.1	Entrenamiento .....	30
3.1.2.2	Gestión del proyecto de Mejora .....	31
3.1.2.3	Análisis de Resultados .....	31
3.1.2.4	Evaluación .....	31
3.1.2.5	Diseño.....	32

3.1.2.6	Implantación .....	32
3.1.2.7	Gestión de la configuración del proceso .....	33
3.1.2.8	Aprendizaje .....	33
<b>Capítulo 4 – Evaluación: Estudio de Caso .....</b>		<b>35</b>
4.1	METODOLOGÍA.....	35
4.2	PREGUNTA.....	36
4.3	OBJETIVO.....	36
4.4	SELECCIÓN DEL ESTUDIO DE CASO.....	36
4.5	CONTEXTO .....	37
<b>Fase: Instalación .....</b>		<b>40</b>
<b>Fase: Diagnóstico .....</b>		<b>41</b>
4.8	ACTIVIDADES PARA EL DIAGNÓSTICO .....	41
4.8.1	Planificar.....	41
4.8.1.1	Objetivos del diagnóstico .....	41
4.8.1.2	Métricas.....	42
4.8.1.3	Indicadores de mejora.....	42
4.8.1.4	Fuentes de evaluación.....	43
4.8.2	Valorar.....	44
4.8.2.1	Formación de los grupos de trabajo .....	44
4.8.2.2	Tiempos registrados de las prácticas realizadas .....	44
4.8.2.3	Tiempos registrados de los requerimientos realizados .....	46
4.8.2.4	Forma de Trabajo .....	47
4.8.2.5	Notas parciales y totales de los estudiantes .....	48
4.8.2.6	Reporte de Asesorías.....	49
4.8.2.7	Reporte de Errores .....	51
4.8.2.8	Tiempo empleado para la creación de las prácticas .....	51
4.8.2.9	Tiempo empleado para la evaluación de las prácticas .....	52
4.8.3	Priorizar .....	52
4.8.3.1	Oportunidades de Mejora.....	53
4.8.4	Construir un plan general de mejora .....	57
4.8.4.1	Plan de Implementación .....	57

4.8.4.1.1 Carta Gantt del proyecto .....	57
4.8.4.1.2 Plan de Administración de Mejoras .....	57
4.8.4.1.3 Agenda del Plan de Control .....	57
4.8.4.1.4 Plan de Medición .....	58
4.8.5 Comunicar el plan general de mejora .....	58
<b>Fase: Formulación</b> .....	59
4.9 ACTIVIDADES PARA LA FORMULACIÓN.....	59
4.9.1 Planificar la iteración de mejora .....	59
4.9.2 Evaluar y analizar procesos .....	59
4.9.3 Diseñar procesos .....	60
4.9.3.1 Iteración 1 .....	60
4.9.4 Adoptar procesos .....	61
<b>Fase: Mejora</b> .....	62
4.10 RESULTADOS IMPLEMENTACIÓN DE MEJORA .....	62
4.10.1 Iteración 1 .....	62
4.10.1.1 Aplicación para generación de guías en PDF.....	62
4.10.1.2 Aplicación de una guía y demo de configuración de equipos.....	62
4.10.1.3 Tutorías para cada una de las tecnologías .....	63
4.10.1.3.1 Comandos en Linux.....	63
4.10.1.3.2 Compilación y ejecución de un programa en C .....	65
4.10.1.3.3 Conocimientos en lenguaje C .....	65
4.10.1.3.4 Conocimientos en lenguaje Java.....	65
4.10.1.4 Entrega de videos del desarrollo de prácticas .....	65
4.10.1.5 Repositorio de prácticas.....	65
4.10.1.6 Entrega de Servidor y Cliente de prueba .....	66
4.10.1.7 Entrega de documento de retroalimentación .....	66
4.10.2 Iteración 2 .....	66
4.10.2.1 Aplicación web que permita al docente ver tiempos de desarrollo de las prácticas y requerimientos.....	66
4.10.2.2 Aplicación del diseño de un modelo de secuencia .....	69
4.10.2.3 Definición de roles y responsabilidades durante el desarrollo .....	69
4.10.2.4 Estrategia para la administración de versiones .....	69

4.10.2.5	Identificación de tareas a realizar durante el requerimiento .....	69
4.10.2.6	Construcción y ejecución preliminar de casos de prueba.....	69
4.10.3	Iteración 3 .....	70
4.10.3.1	Scripts para la agilidad en la calificación de las prácticas .....	70
4.10.3.2	Formalización de la entrega con una lista de soporte de aceptación.....	70
4.11	ENCUESTA REALIZADA A LOS ESTUDIANTES DESPUÉS DE LA APLICACIÓN DE LAS MEJORAS	70
4.12	CRITERIOS PARA EVALUAR LA MEJORA.....	72
4.12.1	Notas parciales y totales de los estudiantes .....	72
4.12.2	Reporte de Asesorías .....	74
4.12.3	Reporte de Errores .....	75
4.13	INFORME DE IMPLEMENTACIÓN .....	76
4.13.1	Esfuerzo de implementación por fases .....	76
4.13.2	Información general de iteraciones .....	77
4.13.2.1	Iteración 1 .....	77
4.13.2.2	Iteración 2 .....	78
4.13.2.3	Iteración 3 .....	78
	<b>Análisis de Resultados</b> .....	79
4.14	MODELO DE PROCESOS REFINADO.....	79
4.14.1	Fase Refuerzos .....	79
4.14.1.1	Actividades de la Fase Refuerzos.....	80
4.14.2	Modificaciones en la Fase Prácticas .....	82
4.14.2.1	Tarea Diseñar.....	82
4.14.2.2	Tarea Desarrollar .....	83
4.14.3	Modificaciones en la Fase Proyectos .....	84
4.14.3.1	Tarea Desarrollar .....	84
4.15	EVALUACIÓN DE LA MEJORA.....	85
4.15.1	Evaluación de la Mejora desde la perspectiva de los Indicadores de Mejora definidos ...	85
4.15.2	Evaluación de la Mejora desde la perspectiva de las deficiencias encontradas .....	89
	<b>Resultados Estudio de Caso</b> .....	94
4.16	RESULTADOS .....	94
4.16.1	Nivel de Utilidad .....	94

4.16.2 Nivel de Aplicabilidad .....	95
4.17 ANÁLISIS DE RESULTADOS.....	96
<b>Capítulo 5 – Conclusiones, Recomendaciones y Trabajo Futuro .....</b>	<b>106</b>
5.1 CONCLUSIONES .....	106
5.2 RECOMENDACIONES .....	108
5.3 TRABAJO FUTURO .....	108
<b>Capítulo 6 – Bibliografía .....</b>	<b>110</b>

# Lista de Figuras

<b>Figura 1.</b> Niveles de Madurez de ISO 15504.....	10
<b>Figura 2.</b> Arquitectura conceptual de Agile SPI .....	12
<b>Figura 3.</b> Modelado bajo SPEM de las fases de Agile SPI-Process .....	13
<b>Figura 4.</b> Agile SPI-Process: Proceso de Mejora iterativo e incremental .....	14
<b>Figura 5.</b> Estructura del Modelo de Procesos de COMPETISOFT .....	17
<b>Figura 6.</b> Estructura de Factor.....	20
<b>Figura 7.</b> Agile SPI Contexto Académico.....	23
<b>Figura 8.</b> Fases y actividades de Agile SPI-Process Académico.....	24
<b>Figura 9.</b> Descripción gráfica del LSD .....	38
<b>Figura 10.</b> Tiempo (min) vs grupos de la práctica Proceso Básico de desarrollo con RPC - LSD II periodo 2011 .....	45
<b>Figura 11.</b> Tiempo (min) vs grupos Requerimiento RPC - LSD II periodo 2011 .....	47
<b>Figura 12.</b> Concurrencia vs Categoría de asesorías Práctica Proceso Básico de desarrollo con RMI - LSD II periodo 2011.....	50
<b>Figura 13.</b> Concurrencia vs Categoría de asesorías Práctica Paso de Parámetros con RMI - LSD II periodo 2011 .....	50
<b>Figura 14.</b> Concurrencia vs Categoría de asesorías Práctica Aplicaciones de tres capas con RMI - LSD II periodo 2011.....	50
<b>Figura 15.</b> Tiempo (min) vs grupos de la práctica Proceso Básico de desarrollo con RPC - LSD I periodo 2012.....	67
<b>Figura 16.</b> Tiempo (min) vs grupos Requerimiento RPC- LSD I periodo 2012.....	68
<b>Figura 17.</b> Concurrencia vs Categoría de asesorías Práctica Proceso Básico de desarrollo con RPC - LSD II periodo 2012.....	74
<b>Figura 18.</b> Concurrencia vs Categoría de asesorías Práctica Paso de Parámetros con RMI - LSD II periodo 2012 .....	74
<b>Figura 19.</b> Concurrencia vs Categoría de asesorías Práctica Paso de parámetros con RMI - LSD II periodo 2012.....	75
<b>Figura 20.</b> Concurrencia vs Categoría de asesorías Práctica Callback con RMI - LSD II periodo 2012.....	75
<b>Figura 21.</b> Modelo de Procesos LSD General .....	80
<b>Figura 22.</b> Diagrama de tareas de la actividad Actualizar tutorial .....	80
<b>Figura 23.</b> Diagrama de tareas de la actividad Preparar práctica tutorial.....	81
<b>Figura 24.</b> Diagrama de tareas de la actividad Apropiar tutorial .....	81
<b>Figura 25.</b> Diagrama de tareas de la actividad Ejecutar práctica tutorial .....	82
<b>Figura 26.</b> Diagrama de tareas de la actividad Preparar evaluación.....	82
<b>Figura 27.</b> Diagrama de tareas de la actividad Evaluar práctica tutorial .....	82
<b>Figura 28.</b> Diagrama de tareas de la actividad Ejecutar práctica.....	82
<b>Figura 29.</b> Diagrama de pasos para la tarea Diseñar - actividad Ejecutar práctica.....	83
<b>Figura 30.</b> Diagrama de pasos para la tarea Desarrollar - actividad Ejecutar práctica.....	83
<b>Figura 31.</b> Diagrama de tareas de la actividad Ejecutar proyecto.....	84
<b>Figura 32.</b> Diagrama de pasos para la tarea Desarrollar - actividad Ejecutar proyecto....	84

## Lista de Tablas

<b>Tabla 1.</b> Plantilla para identificar procesos. ....	26
<b>Tabla 2.</b> Plantilla para priorizar posibles mejoras.....	27
<b>Tabla 3.</b> Plantilla para especificar las oportunidades de mejora .....	28
<b>Tabla 4.</b> Plantilla de comparación de indicadores con resultados obtenidos.....	30
<b>Tabla 5.</b> Plantilla de comparación de indicadores con resultados obtenidos.....	30
<b>Tabla 6.</b> Indicadores y métricas del estudio de caso .....	39
<b>Tabla 7.</b> Registro de tiempos de la práctica Proceso Básico con RPC - LSD II periodo 2011 .....	45
<b>Tabla 8.</b> Registro de tiempos del Requerimiento RPC - LSD II periodo 2011 .....	46
<b>Tabla 9.</b> Oportunidades de mejora .....	56
<b>Tabla 10.</b> Actividad aplicación para generación de guías en PDF .....	61
<b>Tabla 11.</b> Registro de tiempos de la práctica Proceso Básico con RPC - LSD I periodo 2012 .....	67
<b>Tabla 12.</b> Registro de tiempos del Requerimiento RPC - LSD I periodo 2012 .....	68
<b>Tabla 13.</b> Esfuerzo de implementación por fases .....	76
<b>Tabla 14.</b> Información general de iteraciones .....	77
<b>Tabla 15.</b> Información de objetivos por fase .....	77
<b>Tabla 16.</b> Descripción formal de la tarea Recolectar información .....	80
<b>Tabla 17.</b> Descripción formal de la tarea Adaptar concepto LSD.....	81
<b>Tabla 18.</b> Descripción formal de la tarea Presentar tutorial.....	81
<b>Tabla 19.</b> Descripción formal del paso Definir rol – tarea Desarrollar .....	83
<b>Tabla 20.</b> Descripción formal del paso Programación por rol – tarea Desarrollar.....	84
<b>Tabla 21.</b> Descripción formal del paso Definir rol – tarea Desarrollar .....	85
<b>Tabla 22.</b> Descripción formal del paso Programación por rol – tarea Desarrollar.....	85
<b>Tabla 23.</b> Mejora desde la perspectiva de los Indicadores de Mejora definidos.....	88
<b>Tabla 24.</b> Mejora desde la perspectiva de las deficiencias encontradas.....	93
<b>Tabla 25.</b> Resumen de resultados encuesta docente percepción de utilidad.....	94
<b>Tabla 26.</b> Resumen de resultados de la percepción de los estudiantes sobre el nivel de utilidad de las herramientas de mejora. ....	95
<b>Tabla 27.</b> Resumen de resultados encuesta docente percepción de aplicabilidad.....	95
<b>Tabla 28.</b> Esfuerzo general del proyecto .....	96
<b>Tabla 29.</b> Comparación de las definiciones de Agile SPI-Process, la experiencia en el LSD y Agile SPI-Process Académico.....	105



# Presentación

---

La calidad del software es un factor al que se le dedican muchos esfuerzos y ha sido un tema que han abordado un sin número de autores basándose en diferentes modelos; haciendo referencia a la calidad como el conjunto de atributos deseables que posee un producto software, los cuales son medibles (cuantitativa o cualitativamente), permitiendo hacer comparaciones para conocer, entre otras características, si se cumple con las expectativas del usuario o cliente, las cuales permiten indagar sobre la calidad del producto.

En busca de esta calidad para las empresas, la mejora de procesos es una de las maneras de lograrla, en respuesta a esto y para pequeñas organizaciones fue creado Agile SPI<sup>1</sup> – Process<sup>2</sup>. Esta metodología de mejora de procesos, se caracteriza por integrar características ágiles, definir el qué hacer, el cómo hacerlo, ser sencillo de leer, interpretar y aplicar; propone además una infraestructura de trabajo detallada, define técnicas de gestión para facilitar la mejora, guía la mejora de los procesos de desarrollo de software, mantiene el nivel de agilidad que la empresa desee y se adecua a una industria dinámica, creativa, innovadora e incierta como lo es la industria del software.

Con el fin de establecer una línea de trabajo ágil Agil SPI – Process define una serie de principios que deberían de ser considerados como pilares en el desarrollo de un proyecto SPI ágil y efectivo; entre los cuales se encuentran:

- Satisfacer la necesidad del cliente, como prioridad más alta por medio de entregas tempranas y continuas.
- El diagnóstico es una fase clave, permite identificar fortalezas y debilidades frente a un modelo de referencia.
- Entrega temprana de mejoras, obliga a que se generen ciclos cortos.
- Reuniones cara a cara, como manera efectiva y eficiente de comunicar información bilateral.
- Individuos motivados, brindando apoyo, respaldo, confianza y herramientas para realizar las tareas.
- Medición continúa del proceso.
- Mejora y aprendizaje continuos, a través de los ciclos de mejora.
- Infraestructura de trabajo para gestionar proyectos de SPI.
- Minimizar la documentación, solo proveer la de gran valor.
- Independencia del modelo a seguir.

---

<sup>1</sup> SPI: Software Process Improvement

<sup>2</sup> Ver Pardo, Hurtado, Collazos. “Mejora de procesos de software ágil con agile SPI process”. <http://revistas.unal.edu.co/index.php/dyna/article/viewFile/25595/26067>

Agile SPI fue desarrollado por el grupo de investigación IDIS<sup>3</sup> en los últimos años y solo ha sido aplicado en algunas pequeñas empresas de software. Por tanto, es una propuesta que requiere de evidencias y de mayores estudios experimentales para consolidarse como una opción madura a favor de la aplicación de la mejora de procesos en las MIPyMEs<sup>4</sup>.

Teniendo en cuenta lo anterior, se desarrolló una iniciativa que pretende contribuir al área de aseguramiento de la calidad de software, específicamente en el modelado de procesos presentes en el ámbito académico; por ello en esta monografía de trabajo de grado se ejecutó un plan de mejora que permitió evidenciar que Agile SPI es adecuado para establecer e implementar mejoras en las prácticas en cursos de software, al mismo tiempo que se demuestra su versatilidad y nuevas evidencias de su aplicación.

Este documento está compuesto por seis capítulos; dichos capítulos se describen a continuación:

**Capítulo 1 – Introducción:** Este capítulo presenta el planteamiento del problema que dio origen al proyecto y los objetivos propuestos para la solución del problema planteado.

**Capítulo 2 – Marco Teórico:** Este capítulo enmarca las bases teóricas utilizadas para el desarrollo del proyecto, las cuales comprenden: estándares en los cuales se basa dicho proyecto, trabajos y proyectos relacionados.

**Capítulo 3 – Propuesta:** Este capítulo presenta la propuesta del proyecto, los resultados obtenidos durante el desarrollo del proyecto y como principal resultado un procedimiento formal de la aplicación de Agile SPI – Process en un entorno académico.

**Capítulo 4 – Evaluación: Estudio de Caso:** Este capítulo presenta los objetivos propuestos para la solución del problema planteado, el contexto sobre el cual se está aplicando la solución del problema y el desarrollo del caso en cada una de sus fases estudiadas:

Diagnóstico: Aquí se describen las actividades de valoración que permitieron determinar cuál era el estado general de los procesos que se llevaban a cabo dentro del curso de desarrollo intensivo de software, además de describir el análisis de los resultados obtenidos, permitiendo así establecer la prioridad de los casos de mejora y el respectivo plan a seguir.

Formulación: Aquí se describe de manera formal las mejoras y actividades que se aplicaron en el curso de desarrollo intensivo de software.

---

<sup>3</sup> IDIS: Grupo de Investigación y Desarrollo en Ingeniería del Software

<sup>4</sup> MIPyMEs: Micro, pequeña y mediana empresa.

Mejora: Se presentan en detalle los resultados de la ejecución del plan de mejora, teniendo en cuenta las oportunidades de mejora identificadas y priorizadas en la fase de Diagnóstico y las actividades definidas en dicho plan.

Análisis de resultados: Se presenta el análisis de los resultados obtenidos después de ejecutar el plan de mejora, determinando la variación de los procesos y las mejoras resultantes con las comparaciones respectivas antes y después de dicha implementación. Además se describe el modelo de procesos refinado del Laboratorio de Sistemas Distribuidos (LSD<sup>5</sup>).

**Capítulo 5** – Conclusiones, recomendaciones y trabajo futuro: Aquí se describen las conclusiones generadas después de terminar el proyecto, además se presentan posibles mejoras o elementos adicionales que se puedan incluir en futuros trabajos de investigación.

**Capítulo 6** – Glosario y Bibliografía: Este capítulo contiene el catálogo de palabras importantes en la investigación, con su respectiva definición y la bibliografía empleada para soportar el proyecto.

---

<sup>5</sup> De aquí en adelante se usará la sigla LSD para referirse al curso de Laboratorio de Sistemas Distribuidos.

# Capítulo 1 – Introducción

---

## 1.1 PLANTEAMIENTO DEL PROBLEMA

Actualmente, la competitividad es una cualidad importante que debe estar presente en cualquier empresa para su progreso, subsistencia y reconocimiento en el mercado; específicamente las empresas de desarrollo software deben ser constantemente competitivas para producir un software de calidad y en el tiempo definido. Es preciso indicar entonces, que la competitividad, está dada en función de la productividad y la calidad [1].

Por un lado, la calidad del software es un factor al que se le dedican muchos esfuerzos y ha sido un tema que han abordado un sin número de autores basándose en diferentes modelos; entre ellos Roger Pressman, en su libro “Ingeniería Del Software, Un Enfoque Práctico” que define la calidad como la concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente que desea el usuario [2]; y lo planteado en el Estándar 610-1990 de la IEEE<sup>6</sup>, refiriéndose a ésta como el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario [3]. Mientras que la productividad es el grado de rendimiento con que se emplean los recursos disponibles para alcanzar los objetivos predeterminados, factor esencial para las organizaciones.

Dado lo anterior, las organizaciones de TI<sup>7</sup> que desarrollan software deben enfocarse en la gestión de la productividad y la calidad como un mecanismo adecuado para ser competitivas; teniendo en cuenta que no se debe renunciar a una con el pretexto de estar haciendo énfasis en la otra, ya que juntas definirán a la organización como una entidad competitiva [1].

Con el fin de incrementar la competitividad, la mejora de procesos se ha posicionado como una de las estrategias más efectivas por parte de Micro, Pequeñas y Medianas Empresas Desarrolladoras de Software (MiPyMEs\_DS), que están en pro de una continua búsqueda del mejoramiento. En respuesta a esto y enfocado a las pequeñas organizaciones fue creado Agile SPI<sup>8</sup> – Process<sup>9</sup>; el cual se caracteriza por ser liviano, orientado al valor organizacional y por lo tanto aplicable a pequeñas entidades.

---

<sup>6</sup> IEEE: *Institute of Electrical and Electronics Engineers*

<sup>7</sup> TI: Tecnologías de la información

<sup>8</sup> SPI: Software Process Improvement

<sup>9</sup> Ver Pardo, Hurtado, Collazos. “Mejora de procesos de software ágil con agile SPI process”.  
<http://revistas.unal.edu.co/index.php/dyna/article/viewFile/25595/26067>

Agile SPI – Process es un proceso que cuenta con una guía para la mejora de los procesos de desarrollo software a través de iteraciones que comprenden mini-ciclos de mejora, integrando características ágiles, definiendo el qué hacer, el cómo hacerlo, ser sencillo de leer, interpretar y aplicar; además de proponer una infraestructura de trabajo detallada y definir técnicas de gestión para facilitar la mejora [4].

Agile SPI fue desarrollado por el grupo de investigación IDIS en los últimos años y sólo ha sido aplicado en algunas pequeñas empresas de software [4]. Por tanto, es una propuesta que requiere más evidencias y mayores estudios experimentales para consolidarse como una opción madura a favor de la aplicación de la mejora de procesos en las MiPyMEs<sup>10</sup>. De la misma forma, Agile SPI no ha sido aplicado a un entorno académico, siendo de interés demostrar la adaptabilidad de su aplicación en este contexto.

Basándose en este modelo, el proyecto de mejora “*Modelado y Propuesta de Mejora de Procesos de Desarrollo para un entorno académico*” [6] se realizó para los procesos de desarrollo de software en un curso de desarrollo intensivo de software donde los estudiantes realizan la implementación de diferentes prácticas guiadas por el docente. Dicho proyecto abarcó desde el diagnóstico, el modelado de los procesos, el descubrimiento de debilidades hasta la elaboración de un plan de mejora con el soporte de Agile SPI – Process enfocándose en sus fases de Instalación, Diagnóstico y Formulación. Este proyecto permitió evaluar parcialmente que Agile SPI es adecuado para guiar las mejoras en las prácticas en un curso de desarrollo intensivo de software, demostrando la versatilidad y nuevas evidencias de su aplicación.

Como resultado final de este proyecto se obtuvo un plan de mejora y un conjunto de mejoras iniciales que corresponden a la primera iteración, con los cuales se experimentó que Agile SPI-Process no provee una guía fácil de seguir para la implementación de dichas mejoras en los procesos llevados a cabo en un entorno académico.

No obstante, se hizo necesario determinar, evidenciar y continuar con el estudio de Agile SPI – Process en las fases que no se han llevado a cabo en profundidad (fases Mejora y Revisión) y darle continuidad al proyecto de mejora en el curso académico de software. Además de ello, Agile SPI Process no ofrecía mucho detalle sobre cómo implementar las mejoras en forma concreta, sólo definía una estrategia de implementación iterativa e incremental; en busca de ello, fue necesario establecer un procedimiento en el cual se define el cómo ejecutar un plan de mejora adoptando el uso de herramientas (aplicaciones software, plantillas, documentos, guías y entre otras) que implementan las mejores prácticas y permiten la aplicación completa de dicho plan.

---

<sup>10</sup> MiPyMEs: Micro, pequeña y mediana empresa.

Al plan de mejora inicial se le sumaron las mejores prácticas analizadas teniendo como marco de referencia el estándar 29110, el cual establece diferentes requisitos para que las pequeñas entidades<sup>11</sup> produzcan software de calidad reduciendo costos de implementación y mantenimiento, al mismo tiempo que genera guías fáciles de comprender, acceder y utilizar con el fin de que desarrollen o mejoren sus procesos [7]. Para la implementación del plan de mejora fue preciso seleccionar y desarrollar herramientas (aplicaciones software, plantillas, documentos, guías y entre otras), aplicadas al proceso de un curso de desarrollo intensivo de software, principalmente recomendadas en las fases de Mejora y Revisión, que promuevan la aplicación y ejecución de este conjunto de mejores prácticas seleccionadas.

Es entonces cuando surgió la pregunta de investigación ¿Cómo implementar efectivamente las oportunidades de mejora establecidas en un plan de mejora para un curso de desarrollo intensivo de software mediante la ejecución de las fases de Mejora y Revisión de Agile SPI – Process?

## **1.2 OBJETIVOS**

Como objetivos de la propuesta planteada tenemos lo siguiente:

### **1.2.1 Objetivo General**

Ejecutar un plan de mejoramiento en un curso de desarrollo intensivo de software que permita validar y refinar la aplicación de las fases de Mejora y Revisión de Agile SPI-Process en el contexto académico.

### **1.2.2 Objetivos Específicos**

- Priorizar y evaluar oportunidades de mejora en el contexto del proceso académico<sup>12</sup> de un curso de desarrollo intensivo de software.
- Refinar e instrumentar el proceso académico que se lleva a cabo en un curso de desarrollo intensivo de software.
- Evaluar el nuevo proceso del curso académico, comparando los resultados obtenidos antes y después de la ejecución del plan de mejoramiento.
- Establecer un procedimiento que apoye las actividades de las fases de Mejora y Revisión de Agiles SPI-Process para llevar a cabo el proceso de mejora en un curso de desarrollo intensivo de software.

## **1.3 RESULTADOS OBTENIDOS**

A partir de la ejecución del proyecto y dando solución al problema planteado, se generaron los siguientes resultados que constituyen el aporte de este trabajo.

- Planteamiento del estado inicial en que se encuentran los procesos de desarrollo de software antes de la ejecución del plan de mejora.

---

<sup>11</sup> En el contexto de esta tesis, pequeña entidad hace referencia al término VSE de la norma ISO 29110.

<sup>12</sup> Un proceso académico constituye las actividades dentro de un curso académico, que van desde la planeación de una clase por parte del docente, su ejecución en donde los estudiantes realizan las tareas descritas por el docente hasta la evaluación de dichas tareas que generan la calificación para cada estudiante.

- Documento en el que se define y se adecua un plan de mejora a partir del plan resultante del proyecto “*Modelado y Propuesta de Mejora de Procesos de Desarrollo para un entorno académico*”.
- Procedimiento formal (un procedimiento preciso y determinado) que contiene todas las herramientas necesarias: guías, plantillas, documentos, aplicaciones software, entre otras, para llevar a cabo el proceso de mejora.
- Herramientas de soporte para la valoración del curso:
  - Aplicación Web para el LSD, que permitió recoger los tiempos de desarrollo de cada uno de los archivos creados por los estudiantes en las prácticas, código fuente e instaladores.
- Herramientas necesarias: guías, plantillas, documentos y aplicaciones software que se llevaron a cabo para aplicar el plan de mejora. Entre las cuales están:
  - Aplicación Web para el LSD que permite al docente una generación automática de las guías que se entregarán a los estudiantes según la tecnología a usar, de manera que se creen de manera más sencilla y rápida.
  - Tutoriales de refuerzo en las tecnologías necesarias para el curso, con las actividades necesarias para el soporte de cada uno de los temas.
  - Demos de cada una de las prácticas realizadas en las tecnologías impartidas en el curso, además de ejecutables correctos que permitieron probar lo que cada alumno había realizado.
  - Servidores y Clientes de prueba para cada una de las prácticas realizadas que permitieron que los estudiantes probaran si lo realizado por ellos arrojaba los valores correctos según lo solicitado.
  - Videos que permitieron ver el funcionamiento, la compilación y ejecución de un programa según la tecnología usada.
  - Guías de instalación de las tecnologías usadas para la verificación de su existencia en los computadores de la Universidad.
  - Scripts que apoyan el proceso de evaluación que realiza el docente permitiendo la rápida calificación de las prácticas entregadas por los estudiantes.
- Resultados y su respectivo análisis, obtenidos después de la ejecución del plan de mejora que se definió.
- Modelo de procesos del curso del LSD refinado.
- Monografía del trabajo de grado. Corresponde al presente documento, donde se describe el proceso seguido en el desarrollo del proyecto, los aportes más sobresalientes, las conclusiones y las recomendaciones para el desarrollo de futuras investigaciones en el área.

## Capítulo 2 – Contexto Teórico

---

### 2.1 DEFINICIONES / CONCEPTOS

#### 2.1.1 PYMES<sup>13</sup>

Las PYMES [5] son pequeñas y medianas empresas, las cuales cuentan con características distintivas, y dimensiones con ciertos límites ocupacionales y financieros. Son empresas que cuentan con un número reducido de empleados, los cuales pertenecen simultáneamente a diferentes grupos o equipos dentro de la empresa, sin responsabilidades bien definidas; son agentes con lógicas, culturas e intereses específicos.

Se ha demostrado, que las empresas de cualquier tamaño mejoran su rendimiento introduciendo programas de mejora de procesos. Sin embargo, el mayor error de implantar el programa SPI en PYMES se da por la falta de seguimiento de los planes de acción y de implantación, debido a que son actividades costosas tanto en tiempo como en recursos. Los grandes modelos de SPI que se aplican a las PYMES traen costos asociados a la aplicación y una espera a largo plazo para la observación de resultados, para lo cual no están preparadas.

Las características especiales de las pequeñas empresas hacen que los programas de mejora deban aplicarse de un modo particular a como se hace en grandes organizaciones, lo cual no es tan sencillo como considerarlas pequeñas versiones a escala de grandes compañías, de esta manera se hace necesario determinar planes de mejora en los cuales se pueda aprovechar las características de dichas pequeñas organizaciones y así brindar la mejora necesaria.

Paulatinamente, las actividades de mejora de los procesos de software han ido penetrando en las empresas de tamaño mediano que, a efectos estadísticos, se incluyen bajo el nombre de PYMES por el número de empleados que tiene, pero que difieren mucho de la estructura y funcionamiento de las empresas pequeñas o microempresas. A las microempresas, no le ha llegado aún dicha inquietud por la mejora de sus procesos ya que, en muchos casos, incluso desconocen su existencia.

#### 2.1.2 VSE (Very Small Entity)

Una VSE [8] es una Entidad (empresa, organización, departamento o proyecto) que tiene menos de 25 personas. Es considerada como una entidad dedicada a

---

<sup>13</sup> PYMES: Pequeñas Y Medianas Empresas

las actividades de implementación de software, la cual puede ser una organización, un grupo o un proyecto dentro de una organización.

Las pequeñas entidades (VSEs) están sujetas a una serie de características, necesidades y competencias deseables que afectan a los contenidos, la naturaleza y el alcance de sus actividades.

La industria reconoce el valor de las VSEs en la contribución de servicios y productos de valor, además desarrollan y/o mantienen software que es usado en sistemas más grandes, siendo reconocidas como proveedoras de software de calidad.

La mayoría de los estándares internacionales no son dirigidos a las necesidades de las VSEs. Cumplir con estos estándares resulta difícil, ya que las VSE tienen muy pocas posibilidades de ser reconocidas como entidades que producen software de calidad en dicho dominio. La mayoría de las VSEs no pueden darse el lujo de tener los recursos, en términos de número de empleados, presupuesto y tiempo, ni ver un beneficio neto en el establecimiento de procesos de ciclo de vida del software. Para solventar algunas de estas dificultades, un conjunto de estándares y reportes técnicos han sido desarrollados de acuerdo a un conjunto de características que estas entidades poseen.

## **2.2 ESTANDARES**

### **2.2.1 ISO 12207**

Esta norma establece un marco común para los procesos del ciclo de vida del software: procesos, actividades y tareas que se aplican desde la definición de requisitos, la adquisición de un producto de software o servicio y durante el suministro, desarrollo, operación, mantenimiento hasta la finalización del uso de los productos de software [9].

El estándar describe la arquitectura de los procesos del ciclo de vida de software, pero no detalla cómo implementar las actividades incluidas en tales procesos; por lo tanto, las organizaciones pueden utilizar otros estándares donde estas se especifiquen. Estas actividades se organizan en cinco procesos principales, ocho procesos de soporte, y cuatro procesos organizativos, así como el proceso de adaptación.

La ISO 12207 para la implementación de un plan de mejora se apoya en los procesos de soporte principalmente en los de aseguramiento de la calidad, verificación, validación, revisiones conjuntas y auditorías, y en el proceso organizacional de mejora. En el aseguramiento de la calidad, se definen las actividades para asegurar que el producto software cumpla con los requerimientos especificados; en la verificación y la validación se verifican y se validan respectivamente los productos software; en las revisiones conjuntas se evalúa el

estado de estos productos y en las auditorías se definen las actividades para determinar el cumplimiento de los requerimientos, planes y contratos. Finalmente, en el proceso de mejora se definen las actividades necesarias para establecer, medir, controlar y mejorar los procesos del ciclo de vida de la organización [10]. (Para ver más información sobre ISO 12207 referirse al Anexo A Digital Contexto Teórico, Sección 1).

### 2.2.2 ISO 15504

La norma ISO 15504 SPICE [11] es una norma abierta e internacional para evaluar y mejorar la capacidad y madurez de los procesos. Junto con la ISO 12207 [9], la norma aplica a la evaluación y mejora de la calidad del proceso de desarrollo y mantenimiento de software.

La norma ISO 15504 permite realizar evaluaciones usando niveles de madurez (Ver Figura 1). En esta norma, se han establecido 6 niveles que indican la madurez de la organización: el nivel inferior (nivel 0) corresponde con una organización inmadura, los siguientes niveles van haciendo crecer a la organización en su madurez, hasta el máximo nivel, el nivel 5.



**Figura 1.** Niveles de Madurez de ISO 15504

*Imagen tomada de <http://www.iso15504.es/> Fecha de actualización Enero 2013*

La consecución de los niveles de madurez es de forma escalonada, para alcanzar un determinado nivel de madurez deben haberse alcanzado también los niveles inferiores. Cada nivel de madurez estará formado por un conjunto de procesos, estos procesos se definen dependiendo de los esquemas de certificación [11].

Este marco de evaluación de la capacidad del proceso puede ser utilizado por organizaciones que participan en la planificación, gestión, seguimiento, control y mejora de la adquisición, suministro, desarrollo, operación, evolución y soporte de

productos y servicios. Además de las instituciones que trabajan en la mejora y modelos de mejora de procesos [12].

### **2.2.3 ISO 29110**

La norma ISO/IEC 29110 [7] Ingeniería de Software – Modelo de ciclo de vida para pequeñas entidades es un conjunto de normas y guías que han sido desarrolladas de acuerdo a un conjunto de características y necesidades de las pequeñas entidades. Las guías se basan en los subconjuntos de normas apropiadas, conocidas como perfiles VSE<sup>14</sup>.

El propósito de un Perfil de VSE es definir un subconjunto de las normas internacionales pertinentes en el contexto del entorno VSE. Donde el Perfil se define a partir del modelo de negocio, el nivel de riesgo, el medio de desarrollo, los factores situacionales y el tamaño de la empresa a la cual va orientada la mejora; esta norma define perfiles por cada combinación de dichos valores agrupándolos para ser aplicados a diversas categorías y relacionándolos con procesos como actividades y tareas.

### **2.2.4 CMMI – DEV**

CMMI para desarrollo (CMMI-DEV) [13] propone una solución integrada y completa para las actividades de desarrollo y de mantenimiento aplicadas a los productos y a los servicios. Su objetivo es servir de guía para mejorar el proceso de desarrollo de software, puede aplicarse de dos formas: “representación continua”, para mejorar algunas actividades hasta alcanzar un nivel esperado y “representación por etapas”, mejorando un grupo establecido de actividades organizadas en áreas de proceso. CMMI Propone una solución integrada y completa para las actividades de desarrollo y mantenimiento aplicadas a los productos y a los servicios. El propósito es ayudar a las organizaciones a mejorar sus procesos de desarrollo y mantenimiento, tanto para los productos como para los servicios.

Es un conjunto de mejores prácticas que cubre el ciclo de vida del producto, desde su concepción hasta su entrega y mantenimiento. Organizando estas prácticas en 22 áreas de proceso y establece unos niveles de madurez y capacidad que suponen dos representaciones o caminos posibles a la hora de abordar la mejora de estas áreas de proceso.

---

<sup>14</sup> VSE: Very Small Entities

## 2.3 MODELOS RELACIONADOS

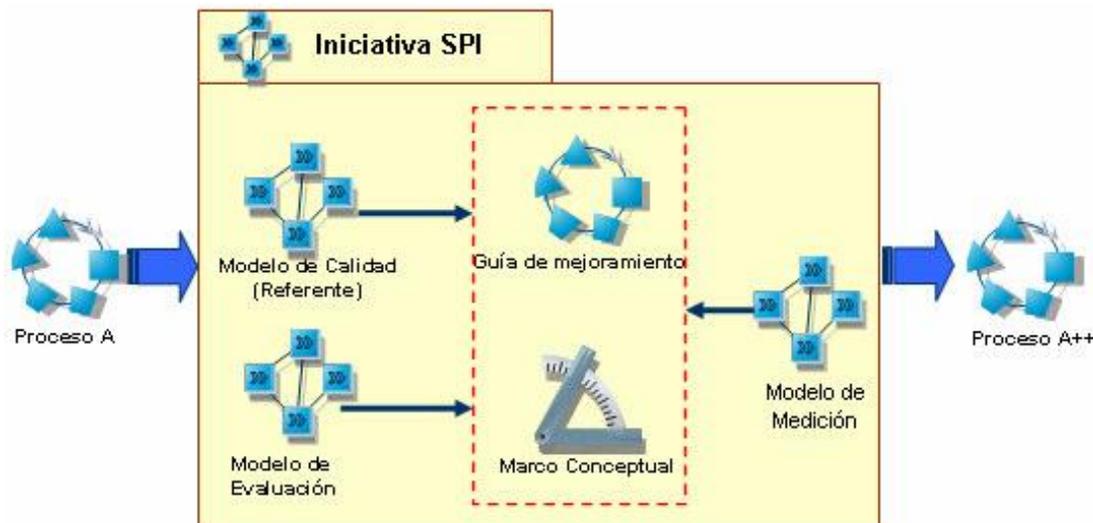
### 2.3.1 AGILE SPI – PROCESS

Agile SPI es un framework caracterizado por:

- Guiar la mejora de los procesos de desarrollo software, manteniendo la agilidad que la empresa desee. Normalmente un proceso liviano y/o ágil adecuado a las PYMES.
- Ser un framework basado en modelos livianos, soporta un programa de mejoramiento continuo, a través de un proceso de mejora ágil Agile SPI – Process.
- Estar adecuado a la industria dinámica, creativa, innovadora e incierta del software, en la cual el conocimiento y el talento humano son elementos fundamentales para garantizar su éxito.

Agile SPI tiene estructura a partir de los componentes primarios de un programa de mejora (Ver Figura 2): una guía de mejora y unos modelos de soporte. Estos modelos son: el de calidad: Agile SPI – Light Quality Model; el de evaluación: Agile SPI – Light Evaluation Model; y el de métricas: Agile SPI – Light Metrics Model. Hay dos elementos integradores de toda la estructura: el modelo conceptual de soporte: Framework PDS y el proceso que integra de manera dinámica los componentes que es: Agile SPI – Process.

A continuación se describe la arquitectura conceptual de Agile SPI:



**Figura 2.** Arquitectura conceptual de Agile SPI

*Imagen tomada de “Proyecto SIMEP-SW Agile SPI - Process”, Julio Ariel Hurtado, Juan Carlos Vidal, Cesar Pardo, Luis Fernandez.*

Dentro de Agile SPI, Agile SPI – Process es el proceso marco de referencia para la gestión de los proyectos de mejora, integra el método, modelos, infraestructura, técnicas y herramientas de soporte.

- **Principios de Agile SPI – Process:**

- Satisfacer la necesidad del cliente con entrega temprana y continua de mejoras significativas al proceso de desarrollo.
- No hay requisitos de mejora estables por parte de la empresa. El diagnóstico es una fase clave. Los requisitos de mejora deberán ser priorizados y acogidos si son factible realizarlos.
- Entregar con frecuencia mejoras del proceso de software.
- La mejora con Agile SPI – Process debe basarse en la colaboración efectiva entre las dependencias relacionadas con el proyecto SPI.
- Proyectos entorno a individuos motivados hacia la mejora de procesos individuales, grupales y organizacionales.
- Comunicar información de ida y vuelta de manera eficiente y efectiva en un equipo de mejora con la conversación cara a cara.
- La madurez del proceso, como el desempeño (productividad y calidad) promedio de los proyectos, debe ser la medida primaria de la mejora del progreso.
- El trabajo deberá ser continuo e indefinido.
- Agile SPI – Process promueve una infraestructura técnica y de gestión, adecuada para soportar la mejora del proceso.
- Agile SPI – Process promueve la conformación de una infraestructura organizacional dinámica, basada en objetivos, no en estrategias de control.
- El objetivo es que permita conocer el trabajo, reflexionar acerca de este y ajustarlo a través de iteraciones cortas y concisas.
- Agile SPI – Process promueve la conformación efectiva de los grupos propuestos por su infraestructura, se preocupa por la calidad del trabajo humano a realizar [14].

- **Ciclo de vida de Agile SPI:**

El ciclo de vida de Agile SPI – Process expone una estructura de proceso para guiar el alcance de los objetivos. Este es ágil, liviano y delimita un camino mínimo pero completo para que una MiPyMES ejecute un ciclo de mejora. Las fases que lo componen son: Instalación, Diagnóstico, Formulación, Mejora y Revisión del Proyecto (Ver figura 3).

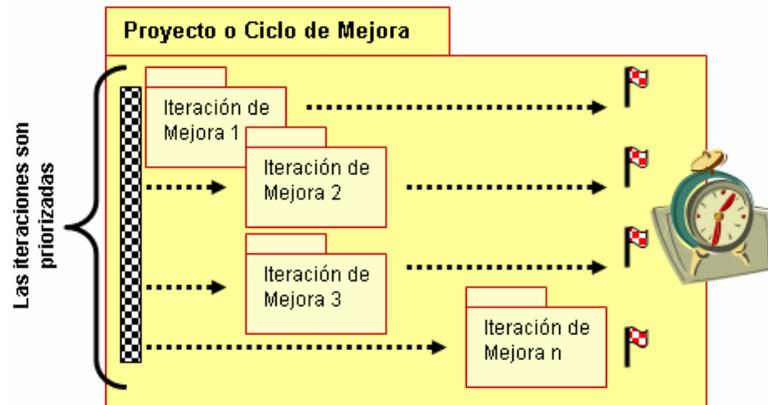


**Figura 3.** Modelado bajo SPEM de las fases de Agile SPI-Process

*Imagen tomada de "Proyecto SIMEP-SW Agile SPI - Process", Julio Ariel Hurtado, Juan Carlos Vidal, Cesar Pardo, Luis Fernandez.*

Dentro de este ciclo Agile SPI – Process se comporta como un proceso iterativo e incremental (Ver Figura 4), basándose en lo que se denomina casos de mejora

(unidades atómicas a mejorar en las áreas de procesos seleccionadas según prioridades). El orden de las iteraciones y sus respectivos casos de mejora dependen de los criterios de priorización de la empresa.



**Figura 4.** Agile SPI-Process: Proceso de Mejora iterativo e incremental  
 Imagen tomada de “Proyecto SIMEP-SW Agile SPI - Process”, Julio Ariel Hurtado, Juan Carlos Vidal, Cesar Pardo, Luis Fernandez.

El arrojar resultados rápidos de mejora permitirá que estas sean visibles desde las fases tempranas del proyecto SPI, esto brinda agilidad y rapidez en la medida que las iteraciones de mejora brindan retroalimentación. La mejora rápida y permanente de los procesos permite motivar al personal frente al proyecto y facilita mitigar riesgos el proyecto en las fases tempranas, en especial en lo referente a aspectos económicos, de motivación, incertidumbre y poca experiencia [4].

Las fases del ciclo de vida se definen a continuación:

**Fase 1 - Instalación:** En esta fase se crea una propuesta de mejora basada en las necesidades del negocio. También se definen los objetivos de mejora generales.

**Fase 2 – Diagnóstico:** Se realizan actividades de valoración para saber cuál es el estado general de los procesos de la empresa; además de un análisis de resultados para establecer la prioridad de los casos de mejora con el fin de crear la guía o plan general de mejora.

**Fase 3 – Formulación:** En esta fase se toman los casos de mejora más prioritarios y se planifican las iteraciones de mejora, con el fin de realizar una medida de esfuerzo, costo y tiempo.

**Fase 4 – Mejora:** En esta fase se gestiona todo el esfuerzo de los casos de mejora en base al plan de ejecución de mejora de la fase anterior y se desarrollan las planificaciones correspondientes a las diferentes iteraciones. Se documentan la ejecución de los pilotos de prueba, la evaluación del antes y después de la

mejora. Si los planes piloto son satisfactorios, se crea un plan de aceptación e institucionalización de los nuevos procesos en la empresa.

**Fase 5 – Revisión del Programa:** Se realiza una retroalimentación antes de volver a comenzar la fase de inicio. Se concluyen todas las lecciones aprendidas y las métricas desarrolladas para medir el cumplimiento de los objetivos [14]. (Para ver más información sobre Agile SPI referirse al Anexo A Digital Contexto Teórico, Sección 2).

### **2.3.2 SIMEP – SW (Sistema Integral para la Mejora de los Procesos Software en Colombia)**

Teniendo como premisa que las empresas de desarrollo de software de Latinoamérica deben implementar proyectos de mejoramiento de procesos de desarrollo y de calidad, existen en Latinoamérica diferentes esfuerzos que propenden por el fortalecimiento de la industria de software de cada país. La mayoría de estos esfuerzos, están enfocados a trasladar los requisitos que imponen los modelos como el CMMI e ISO a empresas típicas en Latinoamérica: las micro, pequeña y mediana empresa de software.

En el marco del proyecto SIMEP-SW [15] se ha definido una estrategia de mejora, la cual intenta cubrir esfuerzos tales como: alivianar requisitos, guiar el proceso de mejora, así como el de generar un conjunto de recomendaciones prácticas para la implementación de los requisitos del proceso software.

La base de la arquitectura está compuesta por modelos de calidad, basándose en la evaluación de modelos ya existentes y de prácticas que siguen un conjunto de empresas de desarrollo de software del sur occidente Colombiano.

Además de contar con un modelo de mejoramiento a seguir por parte de las empresas que definirá la forma de llevar a cabo un proyecto de mejoramiento dentro de una PYME y permitirá implementar los elementos de gestión (un proceso, métodos, prácticas y técnicas), necesarios para asegurar el éxito.

SIMEP para la implementación de procesos de software ágiles crea Agile SPsL<sup>15</sup>, haciendo uso de activos tales como: modelos o instancias de ciclo de vida, actividades, roles, productos de trabajo, guías, flujos de trabajo, disciplinas, etc.; de manera que sirvan para la organización en el proceso de definición, evaluación y de mejora del procesos de software.

### **2.3.3 MOPROSOFT (Modelo de Procesos para la Industria del Software)**

MOPROSOFT [16] es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Desarrollado por

---

<sup>15</sup> Agile SPsL: Agile Software Process Line

la Asociación Mexicana para la Calidad en Ingeniería de Software a través de la Facultad de Ciencias de la Universidad Nacional Autónoma de México (UNAM) y a solicitud de la Secretaría de Economía para obtener una norma mexicana que resulte apropiada a las características de tamaño de la gran mayoría de empresas mexicanas de desarrollo y mantenimiento de software.

Desarrollado principalmente para pequeñas y medianas empresas dedicadas al desarrollo y/o mantenimiento de software, proporcionando un modelo basado en las mejores prácticas internacionales, siendo fácil de entender, de aplicar y proporcionando una adopción no costosa.

El modelo pretende apoyar a las organizaciones en la estandarización de sus prácticas, en la evaluación de su efectividad y en la integración de la mejora continua. Categoriza las mejores prácticas en un conjunto pequeño de procesos que abarcan las responsabilidades asociadas a la estructura de una organización que son:

- Categoría Alta Dirección (DIR): Debe actuar como promotor del buen funcionamiento de la organización a través de su implicación en la revisión y mejora continua del modelo.
- Categoría Gerencia (GER): El modelo considera a la gestión como proveedora de recursos, procesos y proyectos; así como responsable de la vigilancia del cumplimiento de los objetivos estratégicos de la organización.
- Categoría Operación (OPE): El modelo considera a la operación como ejecutora de los proyectos de desarrollo y mantenimiento de software.

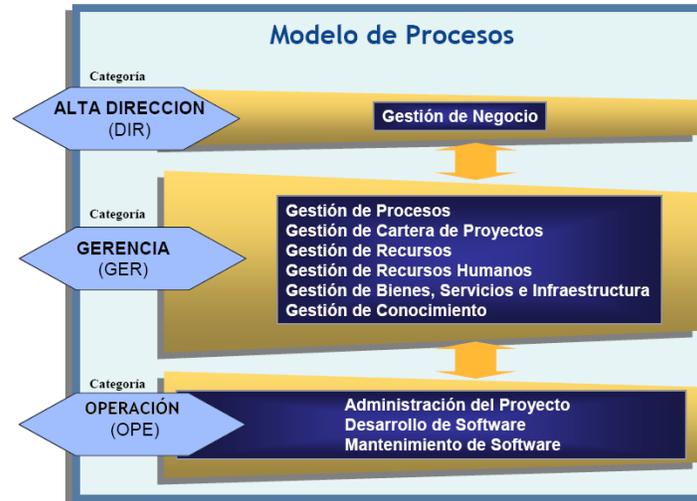
En cada proceso están definidos los roles responsables por la ejecución de las prácticas. Los roles se asignan al personal de la organización de acuerdo a sus habilidades y capacitación para desempeñarlos.

#### **2.3.4 COMPETISOFT**

El objetivo del proyecto COMPETISOFT [17] es incrementar el nivel de competitividad de las PYMES Iberoamericanas productoras de software mediante la creación y difusión de un marco metodológico común (compuesto por Modelo de Procesos, un Método de Evaluación y un Modelo de Mejora) que, ajustado a sus necesidades específicas, pueda llegar a ser la base sobre la cual se establezca un mecanismo de evaluación y certificación de la industria.

- **Modelo de Procesos:**

El modelo de procesos está basado en MoProSoft (Ver Figura 5), se compone de tres categorías de procesos: Alta Dirección, Gerencia y Operación que reflejan la estructura de una organización.



**Figura 5.** Estructura del Modelo de Procesos de COMPETISOFT

*Imagen tomada de "COMPETISOFT, Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica".*

En cada proceso se definen los roles responsables por la ejecución de las prácticas; estos roles se clasifican en Grupo Directivo, Responsable de Proceso y otros roles involucrados. Además se considera al Cliente y al Usuario como roles externos a la organización.

Por otro lado, el modelo de evaluación de procesos, no está definido en particular, cada país interesado define su propio modelo que esté de acuerdo a las necesidades de su industria de software y conforme a las normas ISO/IEC 15504-2 e ISO/IEC 15504-4.

- **Modelo de Mejora:**

El Modelo de Mejora de Procesos está basado en algunos componentes de Agile SPI; este ha sido desarrollado con el fin de:

- Establecer los elementos necesarios para guiar y gestionar la mejora de procesos en una pequeña organización.
- Facilitar su aplicación en las pequeñas organizaciones de forma económica, con pocos recursos y en poco tiempo, buscando siempre obtener resultados de mejora visibles a corto plazo.

Competisoft pretende brindar herramientas necesarias para fomentar en las empresas de software de Colombia la mejora en sus procesos de desarrollo. El modelo liviano recopila las mejores prácticas de desarrollo de software de los modelos de calidad más conocidos, motivando así, a las empresas a establecer un modelo en pro de la calidad de sus productos.

(Para ver más información sobre COMPETISOFT referirse al Anexo A Digital Contexto Teórico, Sección 3).

## **2.4 TRABAJOS RELACIONADOS**

### **2.4.1 Modelado y Propuesta de Mejora de Procesos de Desarrollo para un entorno académico**

López y Magé [6] se basan en Agile SPI-Process, adecuándolo como guía de mejoramiento de las prácticas seguidas en el curso de desarrollo intensivo de software. Como resultado, el proyecto genera un Plan de Mejora que se aplica al curso abarcando desde el diagnóstico, el modelado de los procesos, el descubrimiento de debilidades hasta la elaboración de este plan de mejora con el soporte de Agile SPI – Process enfocándose en sus fases de Instalación, Diagnóstico y Formulación.

Este trabajo define un modelo de procesos para guiar, especificar, controlar, medir y mejorar el proceso que se lleva a cabo en un curso de desarrollo intensivo de software en la Universidad del Cauca, donde los estudiantes realizan diferentes prácticas guiadas por el docente, contribuyendo así a la calidad del software que se desarrolla en el entorno académico.

El proyecto permitió evaluar parcialmente que Agile SPI es adecuado para guiar las mejoras en las prácticas en un curso de desarrollo intensivo de software, al mismo tiempo que establece una evidencia empírica obtenida de la aplicación de este en el caso de estudio, sin embargo se evidencia que la definición de las actividades en las fases Mejora y Revisión de Agile SPI –Process no son especificadas para la implementación de un plan de mejora en un curso de desarrollo intensivo de software.

### **2.4.2 Aplicación de modelado de software y mejora de procesos en el ámbito académico.**

Jaccheri y Lago [18] trabajan sobre el proceso de software en la Universidad “Politecnico di Torino” en Italia; definen un manual de calidad que contiene un modelo de proceso en general, escrito en un lenguaje de modelado de software formal que describe las actividades, herramientas, productos, responsabilidades y medidas. Sirviendo este como base para la obtención de los planes específicos del proyecto, y para comunicar el proceso a los estudiantes, colegas y clientes.

Un modelo de proceso de software es la especificación de un proceso de software. Esta especificación puede tener múltiples objetivos entre los cuales están: la comunicación del proceso de software para los participantes y gestores; la mejora de procesos, la reutilización de conocimiento del proceso, la integración con las medidas de proceso y la ejecución de procesos automáticos.

En este trabajo se presenta la aplicación de las técnicas de modelado de procesos de software en un curso de pregrado de ingeniería de software; explotando las técnicas de modelado de procesos para describir el proceso de software de dicho curso. El modelo de proceso resultante se utiliza para comunicar, razonar, y

mejorar el proceso, y no como una base para la ejecución del proceso automatizado.

El proceso que se sigue es informar sobre el manual de calidad y el plan del proyecto que contiene las actividades necesarias para el desarrollo de una aplicación software específica: los roles que se deben tener, las herramientas a usar y los datos a recoger; teniendo en cuenta un análisis del modelo de procesos del curso de software y con la obtención de la respectiva retroalimentación se obtienen las mejoras sugeridas para dicho modelo. Las mejoras son aplicadas a un nuevo curso de desarrollo software, generando así la definición de un nuevo modelo de procesos, sin determinar el impacto que genera esta mejora realizada sobre los procesos identificados y cambiados.

### **2.4.3 Implementación del Proceso Personal Software en un Ámbito académico**

Hairul, Aalyaa, Sanim y Hashim [19] se basan en el Personal Software Process (PSP), un marco de desarrollo de software estructurado que incluye operaciones definidas, medidas y técnicas de análisis diseñadas para ayudar a los ingenieros de software a entender y desarrollar sus habilidades personales y mejorar su rendimiento. El PSP ofrece un proceso personal definido, guías a los ingenieros de software en la recolección y registro de los datos, y define las formas de analizar datos y hacer mejoras en los procesos. En este trabajo se desarrollaron varias herramientas para superar estos obstáculos, las cuales constituyen una mano de ayuda tanto para los estudiantes e ingenieros en la adopción de la PSP.

Uno de los principios por los cuales se creó las PSP es que todos los ingenieros son diferentes, por lo tanto, los ingenieros deben planear su trabajo sobre la base de sus propios datos personales. El segundo principio implícito en la PSP es que los ingenieros deben utilizar procesos bien definidos y medidos para mejorar constantemente su propio rendimiento. En total, los ingenieros deben entender su desempeño personal, el tiempo que dedican a cada parte de su trabajo y a la eliminación de los defectos.

### **2.4.4 Una experiencia de mejora de procesos software aplicada a la educación**

De Amescua, García, Sánchez, Martínez y Llorens [20] describen el uso de la técnica de Planificación Personal del Trabajo (PWP) como una herramienta de gestión del tiempo para las prácticas de los estudiantes en el curso de Ingeniería de Software II en la Universidad Carlos III. Se analizan los resultados obtenidos y se presenta la metodología utilizada para llevar a cabo actividades relacionadas con la técnica de trabajo de Planificación Personal en una institución académica. Además, se llevó a cabo un estudio empírico para determinar el nivel de satisfacción de los alumnos después de usar esta técnica. Este caso de estudio permitió que muchos estudiantes se dieran cuenta de la utilidad de la PWP para

sus tareas. Esta técnica es de inestimable ayuda a los profesores que desean mejorar el plan de estudios del curso y el de la práctica.

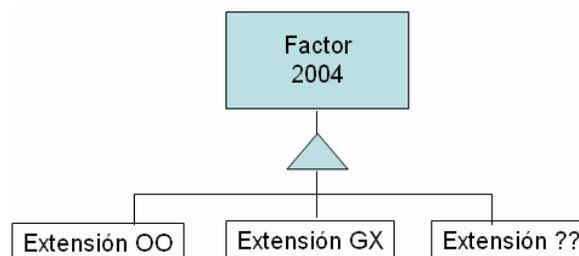
Este trabajo toma como base la PSP, donde se introduce los principios del proceso de software para enseñar a los estudiantes una práctica personal disciplinada para producir trabajos de alta calidad. Los autores de este proyecto creen firmemente que la PSP debería enseñarse en cada curso, pero se deben considerar diferentes aspectos en cada uno, por ejemplo. Por esta razón, las técnicas propuestas en la PSP para planificar el trabajo individual se han extraído y modificado surgiendo la técnica de Planificación Personal del Trabajo (PWP).

#### 2.4.5 MUM - Proceso de Desarrollo de Software Modularizado, Unificado y Medible

Pérez, Pedrana y Bellini [21] definieron un programa de construcción y prueba de modelos de procesos para el Grupo de Ingeniería de Software (Gris) de la Facultad de Ingeniería de la Universidad de la República del Uruguay. El programa cuenta con tres hitos: obtener un proceso definido, validado - ajustado y mejorado. La definición de los procesos se realiza por estudiantes del quinto año que cursan la asignatura “Proyecto de Grado” durante el primer semestre, obtienen como resultado el proceso definido, su puesta en práctica se hace durante el segundo semestre en la asignatura “Proyecto de Ingeniería de Software”. Los docentes del curso son los encargados de dirigir a los grupos cumpliendo el rol de “director”, los grupos tienen una agenda definida con las principales actividades a realizar y los productos que se espera se entreguen al cliente y al director. Cada grupo construye un producto para un cliente real.

El producto se instala en el ambiente del cliente en las últimas dos semanas; luego se identifican mejoras al proceso definido, obteniendo un proceso ajustado y mejorado, el cual puede ser puesto en práctica para el siguiente año.

Desde el año 2000, se han creado distintas versiones de procesos que son puestos en la práctica cada año. En el año 2004 se mejoran los procesos anteriores creando una nueva versión Factor (Ver Figura 6) que tiene un esqueleto común de actividades, entregables y roles, e instancias particulares para desarrollo con Genexus (extensión GX) o para desarrollo orientado a objetos (extensión OO).



**Figura 6.** Estructura de Factor

*Imagen tomada de “MUM – Proceso de Desarrollo de software Modularizado, Unificado y Medible”, Beatriz Perez, Lucía Pedrana, Marcelo Bellini*

En el año 2005 se plantea el objetivo de mejorar las mediciones del proceso y mejorar la satisfacción del cliente. Como resultado se obtiene una nueva versión del proceso llamada “MUM - Modularizado, Unificado y Medible”.

La mejora del proceso tiene como objetivo enriquecer la predicción y la calidad de lo entregado, mientras se mantiene (o se mejora) la productividad.

#### **2.4.6 Aplicación del modelo de procesos en un curso de Ingeniería del Software**

En el centro de Investigación Científica y Superior de Ensenada (CICESE), se imparte la asignatura obligatoria de Ingeniería y metodología de la programación, dentro del programa de Maestrías en Ciencias de la Computación. Las características del curso permitieron tomarlo como referencia por García y Rodríguez [22] para realizar un proyecto de mejoramiento de procesos dentro del mismo. El objetivo de esta investigación fue sistematizar el proceso de desarrollo utilizado hasta este momento en el curso, identificar las prácticas útiles y determinar las debilidades del proceso con el propósito de mejorarlo. En el mejoramiento del proceso que se siguió en el curso, se consideraron las metas de la ingeniería de software para liberar un producto con la funcionalidad acordada con el cliente, la calidad requerida y dentro del tiempo estipulado para el proyecto. Para este estudio, se tomó como referencia las etapas de un proyecto de mejoramiento de procesos: definición, captura, representación, evaluación, rediseño, ejecución y valoración.

La metodología utilizada para mejorar el proceso que se sigue en el curso contempla las etapas de definición, captura, evaluación, rediseño y ejecución. La fase de definición establece los objetivos de un proceso. La fase de representación y captura modela el proceso de manera detallada, partiendo de la información obtenida para generar una representación gráfica del proceso. La etapa de evaluación analiza y evalúa el proceso con el propósito general de buscar debilidades y problemas referentes a la ineffectividad o ineficiencia para lograr las metas del proceso establecidas. Los resultados obtenidos de la evaluación facilitan el rediseño del proceso, en el que se utiliza un lenguaje de modelado comprensible para los usuarios del proceso. El proceso rediseñado se pone en marcha en la organización para comprobar que realmente satisface las metas establecidas.

Los modelos descritos contribuyen a que los estudiantes tengan una mejor comprensión de sus responsabilidades en el proceso y les permiten enfocar sus esfuerzos hacia las actividades que son significativas en el desarrollo del proyecto. Los resultados preliminares, utilizando procesos rediseñados, sugieren que los modelos son una guía poderosa para entrenar a los participantes del proyecto.

### **2.4.7 Programa de mejora UAMISoft**

Cervantes y Martínez [23] definen UAMISoft , un programa de mejora realizado en la Universidad Autónoma Metropolitana Iztapalapa (UAM-I), México, para que facilite la realización de proyecto de software de calidad y permita a los alumnos de licenciatura y posgrado participar en un proyecto de desarrollo siguiendo prácticas de ingeniería de software. UAMISoft tiene dos objetivos principales: preparar profesionales capacitados en el desarrollo de software y desarrollar aplicaciones de calidad; para lograr estos dos objetivos, el proyecto busca implantar un programa de mejora, el cual está basado en el Modelo de Procesos para la Industria Mexicana de Software Moprosoft.

UAMISoft está adaptado al entorno de universidad, donde se tuvo en cuenta la limitación de tiempo y recursos de los alumnos. La etapa inicial de la implantación del proceso de mejora involucra la puesta en pie de un proceso de desarrollo que se adecue al contexto de la UAM-I, donde se tomen en cuenta las limitaciones de tiempo, la dificultad de implantar un proceso en un ambiente universitario, la dificultad de instalación de las herramientas de soporte y la falta de capacitación. (Para ver más información sobre programa de mejora UAMISOFT referirse al Anexo A Digital Contexto Teórico, Sección 4).

## **2.5 ANÁLISIS DE TRABAJOS RELACIONADOS**

A partir de la investigación que se realizó de los diferentes trabajos que están relacionados con el ámbito académico y con la mejora de procesos, se obtuvo que si existen proyectos que se enfocan en la mejora de procesos dentro de los cursos de desarrollo software, la diferencia existente entre los revisados y el proyecto aquí planteado es que la mayoría no tienen un modelo de referencia a seguir para ejecutar los diferentes planes de mejora; estos proyectos basan su mejora en fases básicas de un plan sin seguir rigurosamente las actividades de un plan de mejora, tal y como este proyecto establece. Además de que son proyectos que desarrollan trabajos de desarrollo más extensos, abarcando desde la toma de requisitos, el diseño, la documentación, desarrollo y pruebas hasta la puesta en marcha de la aplicación obtenida, mientras que en el curso sobre el cual se realizó el estudio de caso se desarrollan practicas pequeñas y continuas por cada sesión guiada por el docente.

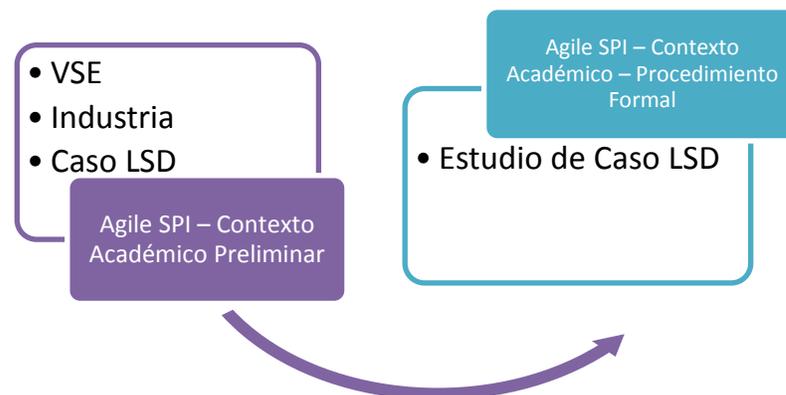
La diferencia principal es que nuestro proyecto define un procedimiento formal donde se especifican herramientas, actividades y diferentes mecanismos necesarios para aplicar en los procesos que se siguen dentro de un curso de desarrollo intensivo de software, esto en busca del mejoramiento de dichos procesos; de lo cual adolecen los trabajos relacionados que se han investigado.

Para ver la comparación de los trabajos relacionados referirse al Anexo A Digital Contexto Teórico, Sección 5.

## Capítulo 3 – Propuesta

Esta tesis propone extender los alcances de Agile SPI-Process hacia un contexto académico, después de contemplar los resultados satisfactorios de su aplicación en industrias. La propuesta pretende conseguir esta adaptación de Agile SPI-Process mediante un procedimiento formal que apoye principalmente las actividades de las fases de Mejora y Revisión de Agiles SPI-Process para llevar a cabo el proceso de mejora en un curso de desarrollo intensivo de software.

A partir de la literatura sobre Agile SPI-Process [14] [4], el trabajo “*Modelado y Propuesta de Mejora de Procesos de Desarrollo para un entorno académico*” [6] y la experiencia de los ingenieros de procesos (tesistas) en cursos de desarrollo intensivo de software, se propuso un procedimiento formal preliminar (ver Anexo B Digital Procedimiento Formal Preliminar). Este procedimiento fue aplicado en un estudio de caso, el cual se describe en el Capítulo 4.



**Figura 7.** Agile SPI Contexto Académico

A partir de los resultados obtenidos y la experiencia en el estudio de caso, se realizaron cambios en dicho procedimiento formal, generando un Agile SPI-Process adaptado a un contexto académico.

### 3.1 AGILE SPI-PROCESS PARA EL CONTEXTO ACADÉMICO: PROCEDIMIENTO FORMAL

Agile SPI - Process como se menciona en uno de sus principios es un proceso ágil y liviano de mejora de procesos de software, el cual puede ser utilizado como guía para la ejecución de un programa de mejora de procesos de software en pequeñas y medianas empresas (PyMES) [14]. Agile SPI-Process en un contexto

académico<sup>16</sup> es mucho más ágil y liviano que para el contexto industrial. Liviano porque los cursos de desarrollo intensivo de software poseen muy bajos recursos, un recurso humano mucho más reducido que las PyMES y limitaciones de tiempo, debido a que los estudiantes deben entregar el desarrollo de las prácticas en las horas disponibles para el curso. Por lo anterior, en el contexto académico se necesita un modelo que soporte un programa de mejora teniendo en cuenta todas las características de los cursos de desarrollo intensivo de software.

### 3.1.1 Fases

Al igual que el Agile SPI – Process para el contexto industrial, el Agile SPI-Process para el contexto académico tiene 5 fases (Ver Figura 8): Instalación, Diagnóstico, Formulación, Mejora y Revisión.



Figura 8. Fases y actividades de Agile SPI-Process Académico

#### 3.1.1.1 Fase 1: Instalación

En esta fase se crea una propuesta de mejora basada en las necesidades del curso, donde se deben definir los objetivos de mejora generales que guiarán las fases siguientes. También se deben definir los objetivos de mejora generales.

##### 3.1.1.1.1 Actividades

**Empezar la instalación:** Para esta actividad es aconsejable organizar un equipo para presentar la propuesta de mejora. Los pasos específicos para esta actividad son:

1. Definir el grupo de mejora: ingenieros de procesos y docente (cliente), teniendo en cuenta el rol de los estudiantes.

<sup>16</sup> De aquí en adelante se denominará Agile SPI-Process Académico para hacer referencia a Agile SPI-Process adaptado a un contexto académico.

**Identificar las necesidades del curso y los requisitos que conducen la mejora:** Se identifican las necesidades del curso, y a partir de éstos se definen los objetivos generales de mejora. Los pasos específicos para esta actividad son:

1. Identificar las necesidades generales del curso de desarrollo intensivo de software (académico).
2. Definir los objetivos de mejora generales.

**Construir una propuesta de mejora del proceso de software:** Se construye una propuesta donde se desarrolle una estrategia para el desarrollo de los objetivos de mejora planteados, el tiempo estimado del proyecto y de sus resultados, razones por las cuales debe de llevarse a cabo, riesgos, etc. Los pasos específicos para esta actividad son:

1. Crear propuesta de mejora inicial respecto a estas necesidades.
2. Definir el estándar de calidad que brinde un conjunto de mejores prácticas para VSE<sup>17</sup>, de forma que es más adaptable al contexto académico. En este trabajo se sugiere trabajar con la ISO 29110.

**Obtener la aprobación de la propuesta SPI:** Esta actividad es donde se presenta la propuesta al docente (cliente) y se requiere de su aprobación para iniciar el proceso de mejora; los pasos específicos para esta actividad son:

1. Obtener la aprobación con el docente del curso sobre la propuesta de mejora inicial.

**Adecuar la propuesta de mejora:** Se realizan las modificaciones necesarias sobre la propuesta de mejora inicial a partir de los comentarios del docente; los pasos específicos para esta actividad son:

1. Adecuar la propuesta si es necesario respecto a las observaciones del docente.

**Lanzar el proyecto de mejora:** Iniciar las actividades determinadas para dar soporte a la infraestructura. Por ejemplo: facilitar, animar y compartir información, comunicar y hacer recomendaciones a los equipos.

### 3.1.1.2 Fase 2: Diagnóstico

En esta fase se evalúa y diagnóstica el estado actual de los procesos en el curso académico.

---

<sup>17</sup> VSE: Very Small Entities

### 3.1.1.2.1 Actividades

**Planificar:** Se definen claramente los objetivos, el alcance de la evaluación de los procesos, la estrategia y las fuentes de evaluación a utilizar. Con base en esta información se planifica las actividades necesarias presentes en esta fase. Los pasos específicos para esta actividad son:

1. Determinar los objetivos del diagnóstico.
2. Definir las métricas e indicadores de mejora necesarios para guiar y obtener la información más relevante para el proyecto.
3. Definir las fuentes de información, tales como: observación directa, aplicación software, encuestas, entrevistas y demás mecanismos necesarios para valorar el estado actual en la actividad siguiente.

**Valorar:** Se realiza una valoración del estado actual de los procesos para identificar el nivel en que estos se encuentran implementados dentro del curso y obtener una visión más clara acerca de los casos de mejora que deben ser iniciados según las necesidades encontradas. Por otro lado la actividad de *valorar* permite evaluar los conocimientos, dificultades, fortalezas, debilidades y errores que presentan los estudiantes, de tal manera que se pueda seleccionar y construir las herramientas de soporte para la implementación de la mejora. Los pasos específicos para esta actividad son:

1. Ejecutar las actividades definidas en las fuentes de información y tabular los datos de manera que generen información cuantitativa y cualitativa.
2. Identificar los procesos, cada uno de los pasos que se siguen dentro de estos, los actores implicados y las diferentes actividades que influyen directa o indirectamente en lo que se realiza dentro del curso. Adicionalmente, se identifican las falencias encontradas a partir de la ejecución de las fuentes de información.

Como plantilla para diagnosticar los procesos se puede usar:

Proceso	Pasos del proceso	Roles implicados	Falencia encontrada	Observaciones

**Tabla 1.** Plantilla para identificar procesos.

**Priorizar:** La información obtenida en la valoración del curso de desarrollo intensivo de software es el fundamento para priorizar los casos de mejora con base a las necesidades del curso. Los pasos específicos para esta actividad son:

1. Identificar las posibles mejoras que se pueden aplicar en cada uno de los procesos que se analizaron dentro del curso.

2. Luego de tener identificadas dichas oportunidades de mejora, se priorizan de acuerdo a la importancia que tengan sobre cada uno de los procesos. Al mismo tiempo se definen las herramientas de soporte candidatas que se podrían usar para aplicarlas.

Como plantilla para la priorización e identificación de las posibles mejoras se puede usar:

Deficiencia	Justificación	Descripción	Herramienta de soporte candidata	Prioridad

Tabla 2. Plantilla para priorizar posibles mejoras

**Construir el plan general de mejora:** Luego de tener las mejoras priorizadas se procede a realizar un plan de implementación de la mejora; los pasos para realizar esta actividad son:

1. Crear un plan general de mejora el cual permite pilotear los procesos a través de todo el proyecto de mejora; además de incluir información referente a cada uno de los casos de mejora que se están iniciando, ejecutando o terminando; como ayuda para mantener una organización de todo el proyecto de mejora.
2. El plan de implementación debe contener principalmente la carta Gantt del proyecto la cual sirve para prever la dedicación, esfuerzo, tiempo y recursos necesarios en las diferentes tareas o actividades que se realizaron en el transcurso del proyecto de mejora y como herramienta básica en la gestión del mismo; un plan de administración de mejoras, una agenda del plan de control y un plan de medición; todo esto con el fin de saber cómo se va a monitorear y gestionar el proyecto y cada una de sus actividades de mejora.

**Comunicar el plan de mejora:** Comunicar el plan de mejora a todos los roles incluidos dentro del proceso: docente y estudiantes. Esto mediante la plataforma web para las actividades del curso, también por medio de una reunión u otros mecanismos de comunicación que se definan.

### 3.1.1.3 Fase 3: Formulación

En ésta fase se toman las necesidades de implementación más prioritarios para introducir mejoras vía la implementación del proceso según los resultados arrojados por la valoración. En esta fase se definen las mejoras a introducir por cada iteración; con base en los resultados de las primeras iteraciones de implementación se realiza la planificación de las demás iteraciones.

### 3.1.1.3.1 Actividades

**Planificar la iteración de mejora:** Como esta fase es de forma iterativa, en esta actividad se describen los objetivos de la iteración y se revisan las oportunidades de mejora descritas. Los pasos específicos para esta actividad son:

1. Definir los objetivos de la iteración.
2. Revisar oportunidades de mejora.

**Evaluar y analizar procesos:** Se evalúan en profundidad las partes del proceso a ser mejoradas en la iteración. Para ello se incluye a los líderes usuarios del proceso para conocer los requisitos propios. Los pasos específicos para esta actividad son:

1. Definir oportunidades de mejora para la iteración y las partes del proceso involucradas.

**Diseñar procesos:** En esta actividad se llevan a cabo tareas de diseño (modelado) para la ejecución de las necesidades de implementación. El resultado es un conjunto actividades, roles, productos de trabajo, plantillas recomendaciones, y demás elementos que provean descripción del área de proceso. Los pasos específicos para esta actividad son:

1. Definir los mecanismos para aplicar cada una de las oportunidades de mejora que se llevarán a cabo dentro de los procesos del curso en la iteración. En esta definición se puede utilizar el conjunto de mejores prácticas proporcionadas por un estándar para VSE, en este caso se sugiere la ISO 29110, de manera que sea adaptable al contexto académico.
2. Describir estos mecanismos. Como plantilla para especificar cada una de las mejoras se puede usar:

<b>Nombre Actividad:</b>	
<b>Descripción:</b>	
<b>Artefacto:</b>	
<b>Estrategia de aplicación:</b>	
<b>Encargado de realizar la actividad:</b>	

Tabla 3. Plantilla para especificar las oportunidades de mejora

**Adoptar procesos:** Se aplican las mejoras definidas a los procesos del curso.

### 3.1.1.4 Fase 4: Mejora

Con las oportunidades de mejora establecidas en la fase anterior, se ejecuta la mejora (de manera iterativa).

#### 3.1.1.4.1 Actividades

**Ejecutar mejora:** Teniendo las posibles mejoras, el plan de cómo se administrará el proyecto y la especificación de las mejoras; se procede a implementar cada una de ellas siguiendo los parámetros establecidos, teniendo presente los objetivos planteados desde el inicio del proyecto. Los pasos específicos para esta actividad son:

1. Aplicar los mecanismos de las oportunidades de mejora.
2. Actualizar plan de implementación para la siguiente iteración.
3. Realizar y actualizar informe de implementación.

#### 3.1.1.5 Fase 5: Revisión

Después de la ejecución de la mejora se realiza una retroalimentación del proceso, evaluando el estado del curso académico antes y después de la mejora, esto a partir de los indicadores que se propongan para determinar el rendimiento de los estudiantes.

##### 3.1.1.5.1 Actividades

**Realizar Retroalimentación:** Durante el transcurso del ciclo de mejora se generan experiencias donde se identifican algunas cosas que se pueden hacer mejor o que se deberían cambiar porque se presentaron problemas o por el contrario porque generaron beneficios consecuentemente a las diversas soluciones planteadas. Por ello, es importante recolectar toda esta información para posteriores ciclos de mejora en el curso académico de desarrollo intensivo de software o futuros proyectos de mejora. Los pasos específicos para esta actividad son:

1. Seleccionar las experiencias más importantes e influyentes que tuvieron lugar en la ejecución del proyecto de mejora.
2. Describir estas experiencias como las lecciones aprendidas en un documento para futuros proyectos.

**Analizar el Impacto de la Mejora:** Después de aplicar las distintas mejoras se hace un análisis del impacto que se obtuvo sobre los procesos y si se cumplió con los objetivos propuestos dentro del curso. Dentro de esta actividad hay que tener en cuenta el historial o información de los casos de mejora para poder desarrollar estudios comparativos en los cuales se pueda determinar si la mejora realizada ha disminuido o aumentado la eficiencia de dichos procesos.

Los pasos específicos para esta actividad son:

1. **Realizar la evaluación de la mejora desde la perspectiva de los Indicadores de Mejora definidos:** Los indicadores son analizados respecto a las decisiones y acciones aplicadas al proceso y los resultados obtenidos después de la aplicación de la mejora; dichos indicadores señalan si se ha cumplido o no los objetivos del proyecto de mejora.

Se evalúa la mejora con los indicadores establecidos, comparando los resultados del diagnóstico y los resultados actuales.

Métrica	Indicador de Mejora	Resultados del Diagnóstico	Mejora con Agile SPI-Process	Análisis

Tabla 4. Plantilla de comparación de indicadores con resultados obtenidos

- 2. Realizar la evaluación de la mejora desde la perspectiva de las deficiencias encontradas:** Las oportunidades de Mejora identificadas a partir de la información recolectada en la fase de Diagnóstico surgieron a partir de las deficiencias encontradas. Se analiza si dichas deficiencias fueron superadas con la mejora realizada, de acuerdo a los resultados actuales y la experiencia obtenida.

Deficiencia	Herramienta de mejora utilizada	Experiencia	Actualidad

Tabla 5. Plantilla de comparación de indicadores con resultados obtenidos

**Refinar Modelo de Procesos:** A partir todo lo realizado en el proyecto de mejora se refina el modelo de procesos del curso, adicionando los procesos adaptados a las mejoras e incluyendo los mecanismos, herramientas y demás actividades agregadas al curso.

**Preparación de un siguiente ciclo de mejora:** Luego de tener qué tan productivo fue la mejora implementada se puede determinar cuáles procesos pueden seguir siendo mejorados y continuar con un proceso de mejora constante, al igual que determinar cuáles son los mejores mecanismos para que el curso sea eficiente y este siempre en pro de la calidad.

### 3.1.2 Disciplinas

En esa sección se describen las disciplinas para Agile SPI-Process que se consideran parte de un proyecto de mejora en un contexto académico.

#### 3.1.2.1 Entrenamiento

En esta disciplina se capacita a los participantes o a los impulsores del proyecto de mejora. Esta capacitación incluye conocer los procesos del curso de desarrollo intensivo de software, sus actividades y toda la información referente que aporte al proyecto de mejora. Por otro lado, la capacitación también incluye los

conocimientos sobre Agile SPI-Process en entornos académicos, sus fases y el procedimiento para aplicarlo.

**Participantes:**

- Ingenieros de procesos
- Docente

**Herramientas:**

- Documentación
- Asesoría de un experto en Agile SPI-Process o en los procesos del curso, según sea el caso.

**3.1.2.2 Gestión del proyecto de Mejora**

Esta disciplina consiste en realizar el plan de trabajo para desarrollar el proyecto de mejora. En esta disciplina se analiza el estado del proyecto de mejora para definir las actividades a ejecutar.

**Participantes:**

- Ingenieros de procesos
- Docente

**Productos de trabajo:**

- Plan de trabajo que contenga las actividades a seguir en la ejecución del proyecto. En este plan se incluye el calendario de trabajo, las fechas y los días destinados para las actividades definidas.

**3.1.2.3 Análisis de Resultados**

En esta disciplina se analizan los resultados obtenidos de la evaluación de los procesos, principalmente en tres fases del proyecto de mejora (Diagnóstico, Mejora y Revisión).

**Participantes:**

- Ingenieros de procesos (Analistas de procesos): interpretan los resultados mediante gráficas o tablas, y su respectivo análisis.

**Productos de trabajo:**

- Análisis de la valoración: el análisis, tabulación y realización de gráficas de la información recogida en la fase de Diagnóstico.
- Análisis de los casos de mejora: el análisis, tabulación y realización de gráficas de la información obtenida después de la ejecución de los casos de mejora.

**3.1.2.4 Evaluación**

Consiste en examinar el estado de lo que nos interesa conocer, como puede ser el trabajo que se ha realizado, los procesos, el desempeño del programa de mejora, etc.

**Participantes:**

- Ingenieros de procesos (Evaluadores).

**Productos de trabajo:**

- Resultados de la valoración: resultados de la ejecución de las fuentes de información seleccionadas para recoger información en la fase de Diagnóstico.
- Resultados de los casos de mejora: resultados obtenidos después de la ejecución de la mejora mediante los mecanismos o herramientas establecidas.

**Herramientas:**

- Observación Directa.
- Entrevistas informales.
- Encuestas.
- Otras técnicas de recolección de información.

**3.1.2.5 Diseño**

En esta disciplina se lleva a cabo el diseño de las soluciones, lo cual abarca diseñar las nuevas mejoras que se definieron para implantar en los procesos del curso, así como también el diseño de los procesos adaptados (refinación del modelo).

**Participantes:**

- Ingenieros de procesos (Diseñadores).

**Productos de trabajo:**

- Mecanismos o herramientas para la ejecución de las mejoras.
- Modelo de procesos refinado: modelo de procesos con sus componentes actualizados después de la ejecución del proyecto de mejora.

**Herramientas:**

- Software para la Definición de Procesos: Herramientas de apoyo en la definición de procesos las cuales permiten modelarlos, una de ellas es Rational Rose. Agile SPI cuenta con el Framework PDS, el cual sirve para este fin.
- EPF Composer.

**3.1.2.6 Implantación**

Esta disciplina consiste en la implantación de cada uno de las mejoras en los procesos, realizando la adaptación y generando los nuevos procesos ya mejorados.

**Participantes:**

- Ingenieros de procesos (Implantadores de la mejora).
- Docente

**Productos de trabajo:**

- Resultados de la Implantación: Éste reporte refleja los resultados obtenidos al adaptar las mejoras a los procesos.
- Proceso: Éste producto de trabajo no es un documento, sino que ya es la forma en la que los estudiantes del curso realizan su trabajo después de haber adoptado la mejora.
- Reporte de la Mejora: Información correspondiente a los cambios realizados al proceso.

**3.1.2.7 Gestión de la configuración del proceso**

En esta disciplina se lleva el registro de los diferentes estados de los procesos. Esto con el fin de controlar los cambios y las adaptaciones a estos procesos a partir de las mejoras realizadas.

**Participantes:**

- Ingenieros de procesos (gestores de seguimiento y control del proceso).
- Docente

**Productos de trabajo:**

- Proceso: Esta disciplina produce cada vez versiones de un proceso a partir de los cambios realizados.

**3.1.2.8 Aprendizaje**

La disciplina Aprendizaje se encuentra en todas las fases, ya que se refiere a las experiencias y lecciones aprendidas que al final del proyecto serán documentadas para futuros proyectos o nuevos ciclos de mejora.

**Participantes:**

- Ingenieros de procesos
- Docente
- Estudiantes

**Productos de trabajo:**

- Lecciones aprendidas: Cada uno de los reportes de los participantes del programa de mejora, para su almacenamiento y posterior revisión en ciclos o iteraciones siguientes.

Agile SPI-Process Académico se diferencia de Agile SPI – Process porque es mucho más liviano debido a que es enfocado al contexto académico, al mismo tiempo que busca esclarecer el “cómo” aplicar un proceso de mejora, que no es detallado por el Agile SPI – Process original.

Cada fase de Agile SPI – Process Académico es más concreta que las de Agile SPI-Process, se definen los pasos necesarios, recomendaciones, actividades y

plantillas de apoyo al ingeniero de procesos. En adición al Agile SPI-Process, en la fase de Instalación de Agile SPI – Process Académico se debe definir un estándar de mejores prácticas que apoye el proceso de mejora (ISO 29110). En la fase de Diagnóstico, Agile SPI – Process Académico agrega la actividad *planificar*, donde se planifique y se guíe la selección de la información necesaria para esta fase. Por otro lado, Agile SPI-Process Académico reestructura la fase de Formulación estableciendo las actividades de *Planificar la iteración de mejora*, *Evaluar y analizar procesos*, *Diseñar procesos* y *adoptar procesos* buscando como principal objetivo el refinamiento del plan de mejora mediante primeras iteraciones y en la fase de Mejora se enfoca en la ejecución de los mecanismos diseñados para implementar las oportunidades de mejora. Finalmente Agile SPI-Process Académico en la fase de Revisión propone analizar el impacto de la mejora desde dos perspectivas: desde los indicadores de Mejora y desde las deficiencias encontradas, además que se indica refinar el modelo de procesos para que se modifiquen o adicionen los procesos adaptados a las mejoras incluyendo los mecanismos, herramientas y demás actividades resultantes del proyecto de mejora.

# Capítulo 4 – Evaluación: Estudio de Caso

---

## 4.1 METODOLOGÍA

Según Runeson et al. [24], aunque los estudios de casos se utilizan comúnmente en áreas como la psicología, la sociología, la ciencia política, trabajo social y en área de negocios, la aceptación de trabajos empíricos en ingeniería de software y sus contribuciones a aumentar el conocimiento está en continuo crecimiento. *Runeson* et al. describe que el estudio de caso es una metodología de investigación adecuada para la ingeniería del software debido a que estudia un fenómeno contemporáneo en su contexto real, buscando mantener la integridad y las características significativas de los eventos; además se lleva a cabo cuando el investigador tiene poco control sobre los eventos y cuando los objetos de estudio son más fáciles de observar en grupo que de manera aislada.

Para la conducción de este estudio de caso ha sido necesario seguir los siguientes pasos:

- **Diseño del estudio de caso:** Se estableció el estudio de caso a partir de la definición de su objetivo, el cual es principalmente evaluar la aplicación de Agile SPI-Process en un contexto académico mediante un procedimiento formal. Con el fin de lograr dicho objetivo, se seleccionó el estudio de caso determinando el tipo de caso de estudio, además de las fuentes de información y recolección de datos en el proceso.
- **Preparación:** Se definieron los instrumentos y protocolos para la recolección de datos en el desarrollo del estudio de caso: capacitación, documentación del proceso, observación directa y las herramientas de recolección (encuesta, entrevistas informales, aplicación software de soporte para la recolección de tiempos, entre otras).
- **Ejecución y recolección de información:** El estudio de caso fue desarrollado durante tres periodos académicos en la Universidad del Cauca, un periodo académico equivale a cuatro meses. La sesión de laboratorio era de dos horas por semana, y la cantidad de estudiantes variaba según el semestre. Dependiendo de los requisitos de cada fase se aplicaban las herramientas o mecanismos de mejora definidos para la práctica. Con ello, se obtuvieron los resultados para el posterior análisis de la información.

- **Análisis:** En el estudio de caso se realizó un análisis de datos de tipo cuantitativo y cualitativo teniendo en cuenta la información obtenida de las diferentes fuentes de información establecidas por los ingenieros de proceso en el desarrollo del estudio.
- **Informes:** Los resultados del estudio de caso fueron reportados y hacen parte de éste documento específicamente en este capítulo.

## 4.2 PREGUNTA

Esta propuesta ejecuta un plan de mejora mediante un procedimiento definido que permita evidenciar que Agile SPI-Process puede ser adaptado para establecer e implementar mejoras en las prácticas de cursos académicos de desarrollo intensivo de software, y al mismo tiempo demostrar su versatilidad y nuevas evidencias de su aplicación. Esto condujo a formular la siguiente pregunta: ¿La propuesta aquí planteada: proveer Agile SPI – Process adaptado para el ámbito académico es útil y aplicable, para este contexto; permitiendo así la mejora en sus procesos?. Respondiendo a esta pregunta se pretende aportar a responder la pregunta de investigación de este proyecto de tesis, esto es “¿Cómo implementar efectivamente las oportunidades de mejora establecidas en un plan de mejora para un curso de desarrollo intensivo de software mediante la ejecución de las fases de Mejora y Revisión de Agile SPI – Process?

## 4.3 OBJETIVO

El objetivo del estudio de caso es evaluar la aplicación de Agile SPI-Process a un contexto académico mediante un procedimiento formal, de manera que se determine si es útil y aplicable a dicho contexto a través de los resultados de la ejecución del proyecto de mejora en los procesos del curso.

## 4.4 SELECCIÓN DEL ESTUDIO DE CASO

Debido a que la práctica de la definición de procesos, de mejoras; observación y ejecución del estudio de caso es desarrollada por los ingenieros de procesos (tesistas), son ellos los indicados para evaluar la complejidad y practicidad de la propuesta. Así, la unidad de análisis es el proyecto de un procedimiento formal definido para un entorno académico partiendo de las fases y adaptación que se le hace a Agile SPI – Process; las fuentes de información primarias son los ingenieros de procesos responsables del procedimiento formal, su selección respondió a criterios de disponibilidad [25] de ingenieros con conocimientos en Agile SPI - Process, debido a que la propuesta aborda la adaptación del mismo. De acuerdo a Benbasat et al. [25], el caso de estudio es de tipo Holístico considerando una unidad de análisis, el modelo de procesos seleccionado fue

debido a su tipo<sup>18</sup> típico (caso real en la industria) y por ser revelatorio<sup>19</sup> (es un caso lo suficientemente completo para evaluar la aplicabilidad de la propuesta).

#### 4.5 CONTEXTO

Teniendo como premisa que la primera fase de Agile SPI – Process correspondiente a la fase de Instalación fue realizada en el proyecto “Modelado y Propuesta de Mejora de Procesos de Desarrollo para un entorno académico” [6], la cual permitió conocer a fondo el contexto en el que se va a llevar a cabo el proyecto, con cada una de las restricciones que este nos ofrece. Al mismo tiempo que se definieron de manera específica cada uno de los objetivos y el alcance del proyecto; se puede determinar cuál es el estado general de los procesos que se llevaban a cabo dentro del curso de desarrollo intensivo de software, siendo este uno de los objetivos principales de la siguiente fase de Agile SPI – Process: Diagnóstico.

Para la aplicación de Agile SPI – Process en entornos académicos se ha tomado como caso de estudio el curso de desarrollo intensivo de software LSD perteneciente a los cursos de pregrado de Ingeniería de Sistemas de la Universidad del Cauca. Este curso está asignado para el semestre VII dentro del programa, el cual complementa la clase de teoría en donde se imparte los conceptos que posteriormente se llevan a la práctica en dicho laboratorio; por lo tanto, los estudiantes tienen la formación necesaria para el desarrollo de software y la ingeniería que este conlleva. Es por ello que en este curso se han analizado los procesos que se llevan a cabo con el fin de obtener la información necesaria que permita tener un diagnóstico tal y como lo propone Agile-SPI.

Las clases son dirigidas semanalmente, el docente diseña las guías correspondientes a la temática a evaluar especificando el problema que se debe resolver de acuerdo a la tecnología definida. Las prácticas se llevan a cabo en parejas y los resultados son entregados al finalizar la clase.

El laboratorio se califica según las notas obtenidas en 3 cortes; el primer y segundo corte corresponden al 70% de la nota y el tercer corte al 30% restante (Ver Figura 9). Cada uno de los cortes evalúa una tecnología diferente: RPC, RMI y CORBA respectivamente. Al finalizar cada corte se realiza un requerimiento (una práctica con un grado de complejidad mayor y realizada en un mayor tiempo) el cual debe ser sustentado individualmente y se promedia con las demás prácticas realizadas en dicho corte.

---

<sup>18</sup> Un caso de estudio típico tiene como objetivo capturar las circunstancias y condiciones de situaciones comunes.

<sup>19</sup> Un caso de estudio revelatorio existe cuando un investigador tiene la oportunidad de observar y analizar un fenómeno previamente inaccesible a la investigación científica.



Figura 9. Descripción gráfica del LSD

#### 4.6 INDICADORES Y MÉTRICAS

Para evaluar de manera objetiva éste caso y dar respuesta a la pregunta formulada para el estudio de caso fue necesario definir un conjunto de métricas<sup>20</sup> e indicadores. A continuación se describen estos indicadores y métricas:

Pregunta de estudio de caso	Indicador	Métrica	Instrumento
¿Agile SPI – Process adaptado para el ámbito académico es útil y aplicable, para este contexto; permitiendo así la mejora en sus procesos?	Utilidad	<p>El desempeño de los estudiantes medido en la nota final obtenida en el curso.</p> <p>La percepción de la utilidad de las herramientas de mejora en el desarrollo de las actividades por parte de los estudiantes en el curso.</p> <p>La percepción de la utilidad del proceso de mejora en el curso por parte del docente.</p> <p>Mejoras realizadas en cada iteración.</p>	<ul style="list-style-type: none"> <li>• Encuestas.</li> <li>• Reporte de Notas finales.</li> <li>• Resultados de la mejora.</li> </ul>
	Aplicabilidad	La percepción de aplicabilidad del modelo de procesos por parte de	<ul style="list-style-type: none"> <li>• Encuesta.</li> <li>• Reporte de Horas.</li> </ul>

<sup>20</sup> Métrica es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado (IEEE).

	<p>los ingenieros de procesos (tesistas).</p> <p>La percepción de aplicabilidad del procedimiento formal de Agile SPI-Process académico, por parte del docente.</p> <p>Esfuerzo necesario para aplicar Agile SPI-Process académico.</p>	
--	---	--

**Tabla 6.** Indicadores y métricas del estudio de caso

La descripción en detalle de los indicadores y la forma en que estos son calculados a través de las métricas identificadas es la siguiente:

**Utilidad:** La utilidad se define como la propiedad por la cual Agile SPI-Process académico adquiere la condición de valor útil para satisfacer los objetivos de mejora propuestos para el curso. Las pautas que se han establecido para calcular la utilidad son:

- El rango de estudiantes que aprueben el curso esté entre 80% y 100%.
- El rango de estudiantes que consideren que las herramientas de mejora son un apoyo positivo para el curso esté entre el 80% y 100%.
- El porcentaje del número de preguntas que tengan un impacto positivo sobre el proceso de mejora obtenido a partir de la percepción del docente entre el nivel cuatro y cinco (siendo cinco el grado de utilidad más alto) sea superior o igual al 80%.

**Aplicabilidad:** La aplicabilidad se define como la propiedad por la cual Agile SPI-Process académico se puede emplear fácilmente para obtener resultados de mejora favorables para el curso. Las pautas que se han establecido para determinar la aplicabilidad son:

- El promedio del grado de aplicabilidad del procedimiento formal obtenido a partir de la percepción del docente que esté entre cuatro y cinco (siendo cinco el grado de aplicabilidad más alto) sea superior o igual al 80%.
- El esfuerzo para aplicar Agile SPI-Process académico sea en promedio de cuatro meses (duración de un periodo académico).

#### 4.7 EJECUCION DEL ESTUDIO DE CASO Y RESULTADOS

El estudio de caso se realizó en el LSD descrito anteriormente, en tres periodos académicos (II-2011, I-2012, II-2012) del curso, la propuesta abarcará varios iteraciones, donde cada uno de ellos realizará una de las oportunidades de mejora encontradas y contendrá las fases de Agile SPI – Process de la siguiente forma:

## Fase: Instalación

---

Esta fase se realizó en el proyecto “Modelado y Propuesta de Mejora de Procesos de Desarrollo para un entorno académico” [6] nombrado anteriormente, donde se creó una propuesta de mejora basada en las necesidades del curso, la cual ayudará a guiar cada una de las fases siguientes. Durante esta fase también se definieron los objetivos de mejora generales, los cuales son establecidos desde las necesidades del curso.

El proyecto aquí planteado orienta su mejoramiento de procesos enfocándose en cada uno de los procesos que se siguen en el curso de desarrollo intensivo de software, desde el momento de su preparación por parte del docente, su ejecución por los estudiantes y su respectiva calificación por parte del docente. Son procesos que están relacionados con la manera como se lleva a cabo una codificación de productos pequeños de software y no sobre la temática impartida en las diferentes tecnologías de parte del docente. Es por esta razón que la calidad y la mejora de los procesos no se especializan sobre el tipo de información entregada a los estudiantes, sino la manera como ellos aprovechan cada una de las herramientas que se provee para el mejoramiento de sus prácticas de desarrollo dentro del curso.

## Fase: Diagnóstico

---

En la fase de Diagnóstico de Agile SPI-Process se inicia un programa hacia la mejora de los procesos y el trabajo que aquí se realizó fue fundamental para la realización de las fases siguientes. La fase incluye la realización de actividades de valoración para saber cuál es el estado general de los procesos del curso de desarrollo intensivo de software, además del análisis de los resultados que permitieron establecer la prioridad de los casos de mejora. Posteriormente con ello, se creó uno de los productos de trabajo principales de esta fase conocido como guía o plan general de mejora.

### 4.8 ACTIVIDADES PARA EL DIAGNÓSTICO

Para la realización de la fase de diagnóstico dentro de Agile SPI – Process, es preciso seguir una serie de actividades que permitan conocer el estado actual de los procesos; dichas actividades realizadas para cumplir el objetivo son:

#### 4.8.1 Planificar

Antes de iniciar actividades, se definieron claramente los objetivos, el alcance de la evaluación de los procesos, la estrategia y las fuentes de evaluación a utilizar. Con base en esta información se planifica las actividades necesarias presentes en esta fase.

Con el objetivo de planificar el diagnóstico a realizar se establecieron los objetivos, métricas e indicadores necesarios para guiar y obtener la información más relevante para el proyecto.

##### 4.8.1.1 Objetivos del diagnóstico

A continuación se establecen los objetivos que permitieron obtener los estados actuales de los procesos dentro del curso:

- Determinar si el uso de las herramientas proporcionadas para la realización de las prácticas reducen el tiempo de desarrollo que emplean los estudiantes y facilitan la producción de código en términos de agilidad en su codificación.
- Establecer si la aplicación de las mejoras sobre los estudiantes permite que se completen satisfactoriamente las prácticas planteadas por el docente.
- Determinar si la aplicación de las mejoras sobre el curso LSD incrementó el porcentaje de estudiantes que aprobaron la materia.

- Establecer si las ayudas brindadas a los estudiantes son de utilidad para las dudas, asesorías y errores que se presentan en el desarrollo de cada una de las prácticas del curso.
- Determinar qué características, actividades y tareas de Agile SPI son adoptadas con satisfacción para la aplicación de las mejoras en el curso académico de desarrollo software.

#### **4.8.1.2 Métricas**

A continuación se definen las métricas utilizadas para obtener los indicadores de mejora mediante la adopción de Agile SPI:

- Tiempo promedio de desarrollo: El tiempo promedio que emplean los estudiantes en la elaboración del código del servidor, el cliente y la interfaz en las prácticas de laboratorio.
- Porcentaje de grupos con registro de tiempos incompletos y tiempos completos: Porcentaje de grupos que registraron en totalidad los tiempos (tiempos iniciales y finales de todos los módulos) y el porcentaje de grupos que no registraron todos los tiempos (no finalizaron la codificación de uno o varios módulos de la práctica).
- Promedio de notas: Indica la nota final la cual es promediada de las notas parciales con su respectivo porcentaje equivalente en cada corte.
- Porcentaje de reporte de asesorías: El porcentaje de cada tipo de asesoría que realizaron los estudiantes durante el semestre.
- Porcentaje de reporte de errores: El porcentaje de cada tipo de errores que presentaron los estudiantes en la elaboración de sus prácticas.

#### **4.8.1.3 Indicadores de mejora**

Respecto a las métricas definidas, se establecieron los siguientes indicadores de mejora:

- Tiempos promedio de desarrollo después de la mejora: se comparan los tiempos promedio de desarrollo antes y después de la mejora. Si los tiempos de desarrollo después de la mejora son más reducidos, indica que las herramientas proporcionadas a los estudiantes en la mejora han influido positivamente en la producción del código necesario en las prácticas.
- Porcentaje de grupos que finalizaron la práctica: Comparar el antes y después de la mejora, el porcentaje de los grupos que registraron tiempos finales, lo cual indica que finalizaron por completo la práctica dentro del tiempo establecido por el docente.

- Porcentaje de estudiantes que aprobaron la materia: El estudiante que obtenga una nota final superior o igual a 3.0 aprueba la asignatura. Por tanto, si el porcentaje de estudiantes que aprueba la materia es mayor después de la mejora, la deserción de estudiantes habrá reducido con la aplicación de Agile SPI.
- Porcentaje de reporte de asesorías después de la mejora: El porcentaje de cada tipo de asesoría que realizaron los estudiantes durante el semestre después de la mejora. Estos resultados se comparan con el reporte de asesorías antes de la mejora, para establecer si se redujo la necesidad de asesoría y que tipos de asesorías siguen siendo las más recurrentes.
- Porcentaje de reporte de errores después de la mejora: Comparar el porcentaje de cada tipo de errores que presentaron los estudiantes en la elaboración de sus prácticas antes y después de la mejora, y definir cuáles se redujeron y qué errores aun se presentan.

#### **4.8.1.4 Fuentes de evaluación**

Las herramientas usadas para la obtención de la información necesaria fueron:

- Aplicación Web para el Registro de Tiempos: de acuerdo a los grupos formados por los estudiantes del curso, se les asignó a cada uno un usuario y una contraseña; los estudiantes ingresaban a la plataforma cada que iniciaban una práctica según la tecnología asignada para la misma y registraban los tiempos de inicio y fin de cada uno de los archivos que eran necesarios desarrollar para cumplir con los requerimientos solicitados. El docente, quien contaba con un usuario y una contraseña asignados tenía como opción la manera de ver por cada grupo los tiempos que se iban registrando, lo que permitía controlar los estudiantes que cumplían a tiempo con la práctica para la sesión planteada.
- Observación Directa: la asistencia a cada una de las sesiones programadas por el docente para la realización de las prácticas, permitió que se diagnosticaran falencias y actividades que podrían ser mejoradas en pro de obtener mejores resultados de parte tanto de los estudiantes como del docente.

Por otro lado, se consideraron como ventajas para la obtención de la información y soporte a las fuentes de evaluación:

- La experiencia, debido a que las ejecutoras del proyecto como estudiantes del programa de Ingeniería de Sistemas ya habían cursado la asignatura sobre la cual se estaba tomando el caso de estudio. Por tanto, se facilitó el conocimiento de los procesos que se llevan a cabo dentro del curso y con esto determinar posibles mejoras al mismo.

- El apoyo del docente de la materia, dado que la permanente compañía del docente durante la ejecución del proyecto, permitió que las dudas fueran solucionadas a tiempo.

#### **4.8.2 Valorar**

Antes de comenzar a evaluar los procesos del curso de desarrollo intensivo de software fue conveniente realizar una valoración del estado actual de éstos. Lo anterior se realizó con el fin de identificar el nivel en que los procesos se encuentran implementados dentro del curso en base a un referente, esto permitió obtener una visión más clara acerca de los casos de mejora que deben ser iniciados según las necesidades encontradas.

En el curso de desarrollo intensivo de software LSD se recogió la información necesaria para realizar el Diagnóstico que establece Agile SPI – Process; esto con el fin de evaluar los conocimientos, dificultades, fortalezas, debilidades y errores que presentan los estudiantes, de tal manera que se pueda seleccionar y construir las herramientas de soporte para la implementación de la mejora.

A continuación se describe la información recolectada:

##### **4.8.2.1 Formación de los grupos de trabajo**

Para cada una de las prácticas y requerimientos solicitados se definieron parejas seleccionadas según el criterio propio de los estudiantes para trabajar durante todo el semestre. Los estudiantes del segundo periodo del 2011 que cursaron la materia se organizaron en 12 grupos (Ver Formación de los grupos de trabajo en el Anexo C Digital Medidas realizadas en el Diagnóstico, sección 1).

##### **4.8.2.2 Tiempos registrados de las prácticas realizadas**

Por cada una de las prácticas que los estudiantes realizaron se midió el tiempo que estos empleaban en la codificación de los archivos solicitados en las guías entregadas por el docente, según el tema correspondiente. Dichos tiempos se recolectaron en minutos.

Las prácticas en las cuales se registraron estos tiempos fueron:

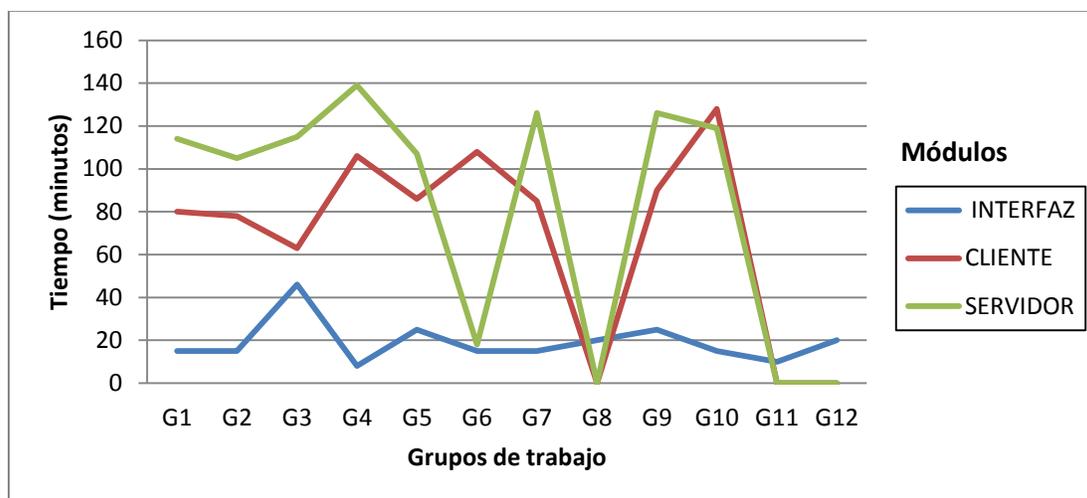
- Práctica 1: Proceso Básico de desarrollo con RPC
- Práctica 2: XDR y RPC
- Práctica 3: Proceso Básico de desarrollo con RMI
- Práctica 4: Paso de Parámetros con RMI
- Práctica 5: Carga Dinámica con RMI
- Práctica 6: Aplicaciones de tres capas con RMI
- Práctica 7: Proceso Básico de Desarrollo con CORBA
- Práctica 8: CORBA e IDL
- Práctica 10: Invocación Dinámica con CORBA
- Práctica 11: Servidores por herencia y parámetros OUT – CORBA

## Análisis de tiempos por práctica

### Práctica 1: Proceso Básico de desarrollo con RPC

GRUPO	INTERFAZ	CLIENTE	SERVIDOR
G1	15	80	114
G2	15	78	105
G3	46	63	115
G4	8	106	139
G5	25	86	107
G6	15	108	18
G7	15	85	126
G8	20	0	0
G9	25	90	126
G10	15	128	119
G11	10	0	0
G12	20	0	0

**Tabla 7.** Registro de tiempos de la práctica Proceso Básico con RPC - LSD II periodo 2011



**Figura 10.** Tiempo (min) vs grupos de la práctica Proceso Básico de desarrollo con RPC - LSD II periodo 2011

Basándose en la anterior tabla y su respectiva gráfica, se puede observar que en la primera práctica referente al proceso básico que se sigue en RPC, los estudiantes en general emplearon menos tiempo en la realización de la interfaz, seguido del tiempo empleado en el cliente y por último en el servidor.

Estas medidas de tiempo son muy similares entre todos los grupos de trabajo, siguiendo un patrón de rangos muy parecido entre dichos grupos.

Las posteriores prácticas con su tabla, gráfica y análisis respectivo, se encontrarán en el Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 2.

#### 4.8.2.3 Tiempos registrados de los requerimientos realizados

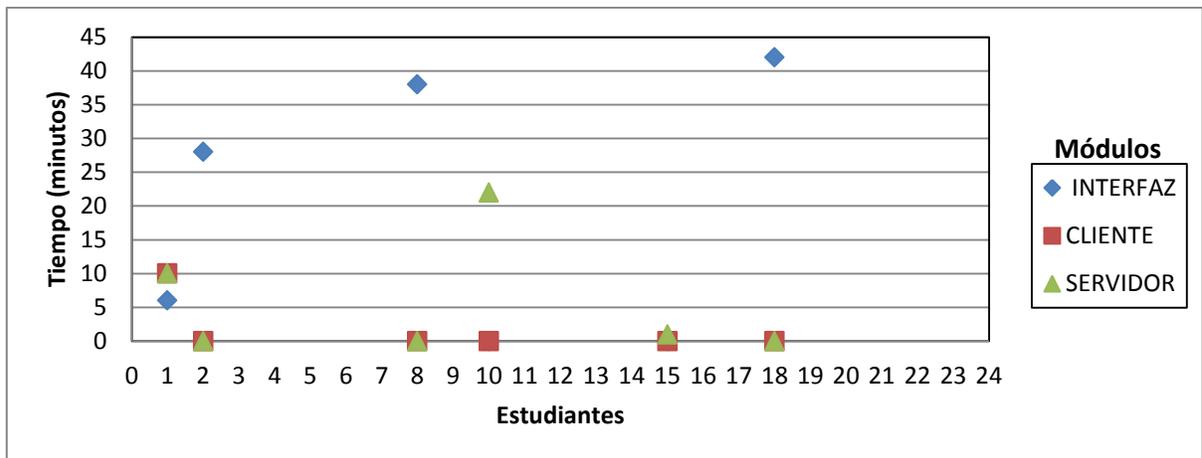
Los estudiantes presentan por cada una de las tecnologías un requerimiento el cual es sustentado en una fecha establecida por el docente, durante un rango de tiempo de media hora por cada grupo los estudiantes realizan la sustentación de lo que realizaron mediante el desarrollo de un nuevo requerimiento que verifica que cada uno de los integrantes tenga pleno conocimiento del código de la aplicación entregada. Para esta sustentación se midió el tiempo que cada estudiante empleaba en la realización del mismo.

A partir de una lista de chequeo que se aplicó a cada estudiante en su sustentación se obtuvieron los siguientes tiempos y observaciones (ver la plantilla de la lista de chequeo de sustentación en el Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 3):

#### Requerimiento 1: RPC

	INTERFAZ	CLIENTE	SERVIDOR
Estudiante 1	6	10	10
Estudiante 2	28	0	0
Estudiante 3			
Estudiante 4			
Estudiante 5			
Estudiante 6			
Estudiante 7			
Estudiante 8	38	0	0
Estudiante 9			
Estudiante 10	0	0	22
Estudiante 11			
Estudiante 12			
Estudiante 13			
Estudiante 14			
Estudiante 15	0	0	1
Estudiante 16			
Estudiante 17			
Estudiante 18	42	0	0
Estudiante 19			
Estudiante 20			
Estudiante 21			
Estudiante 22			
Estudiante 23			
Estudiante 24			

**Tabla 8.** Registro de tiempos del Requerimiento RPC - LSD II periodo 2011



**Figura 11.** Tiempo (min) vs grupos Requerimiento RPC - LSD II periodo 2011

Basándose en la anterior tabla y su respectiva gráfica, se puede observar que en la primera sustentación del requerimiento referente a RPC, los estudiantes en general no terminaron la implementación del requerimiento solicitado para la sustentación y de los pocos archivos terminados se puede ver que en lo que más emplean tiempo es en la realización del cliente, seguido de la interfaz y por último del servidor.

Los posteriores requerimientos con su tabla, gráfica y análisis respectivo, se encontrarán en el Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 2.

#### 4.8.2.4 Forma de Trabajo

Durante la medición de los tiempos en cada práctica se realizaron observaciones sobre la forma de trabajo de los estudiantes, la cual afecta también la productividad en la ejecución de las prácticas. Dichas observaciones se resumen en los siguientes porcentajes:

- El 80.83% de los grupos realiza la repartición del código, uno de los estudiantes se encarga del servidor y el otro de cliente (trabajan por separado).
- El 17.34% de los grupos tienen apoyo de un estudiante asistente.
- El 21.38% de los grupos trabajan juntos en un solo computador.
- El 16.75% de los grupos utilizan computador de apoyo.
- El 16.7% de los grupos visualizan el cliente y el servidor en una misma pantalla de manera paralela.
- EL 25% de los grupos envían la práctica sin compilar y ejecutar ni el cliente ni el servidor.

Adicionalmente, se observó que:

- Muchos grupos se apoyan en prácticas ya realizadas, o por el demo hecho en la clase.

- Muchos grupos buscan información referente al tema básico de la práctica haciendo uso del Internet.

Los porcentajes son resultado de las observaciones registradas para cada grupo en cada una de las prácticas.

#### **4.8.2.5 Notas parciales y totales de los estudiantes**

El curso de LSD se compone de tres cortes, para cada uno de los cuales se realizan prácticas referentes a los temas de RPC<sup>21</sup>, RMI<sup>22</sup> y CORBA<sup>23</sup> respectivamente. El primer y segundo corte sumados equivalen al 70% de la nota total y el tercer corte al 30% restante.

El criterio de evaluación que se tiene en cuenta para cada una de las prácticas es el cumplimiento de las funcionalidades solicitadas. Mientras que los criterios que se tienen en cuenta para la calificación del requerimiento son: manuales, aplicación y el factor de sustentación.

Los estudiantes del segundo periodo del 2011 que cursaron la materia obtuvieron notas para cada una de las prácticas realizadas, a partir de las cuales se obtuvieron los siguientes resultados en las notas finales:

(Ver la discriminación de las notas de cada corte en el Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 4).

- **Notas Finales**

La nota final proviene de las notas finales de cada uno de los cortes, teniendo en cuenta que el primer y el segundo corte equivalen a un 35% cada uno y el tercer corte a un 30% de la nota final.

Las notas finales iguales o superiores a 3,0 indican que la materia es aprobada y consecuentemente, las notas inferiores a dicho valor indican que la materia es reprobada.

Teniendo en cuenta las notas finales de los estudiantes (Ver la discriminación de las notas en el C Medidas realizadas en el Diagnóstico, Sección 3) se obtuvieron las siguientes observaciones:

#### **Observaciones:**

- El 70,833% de los estudiantes tuvieron notas finales superiores o iguales a 3,0 lo que corresponde a los estudiantes que aprobaron la materia.
- El 29,166% de los estudiantes tuvieron notas finales inferiores a 3,0 lo cual refiere al porcentaje de estudiantes que reprobaron la materia.

---

<sup>21</sup> RPC: Remote Procedure Call,

<sup>22</sup> RMI: Java Remote Method Invocation

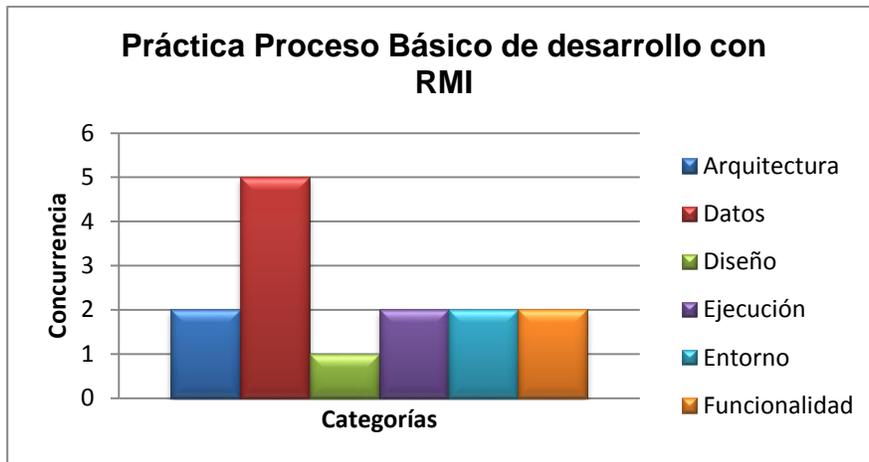
<sup>23</sup> CORBA: Common Object Request Broker Architecture

En general la mayoría de los estudiantes del curso obtuvieron notas finales superiores o iguales a 3,0, lo cual significa que un gran porcentaje de los estudiantes aprobaron la materia para el segundo periodo del 2011.

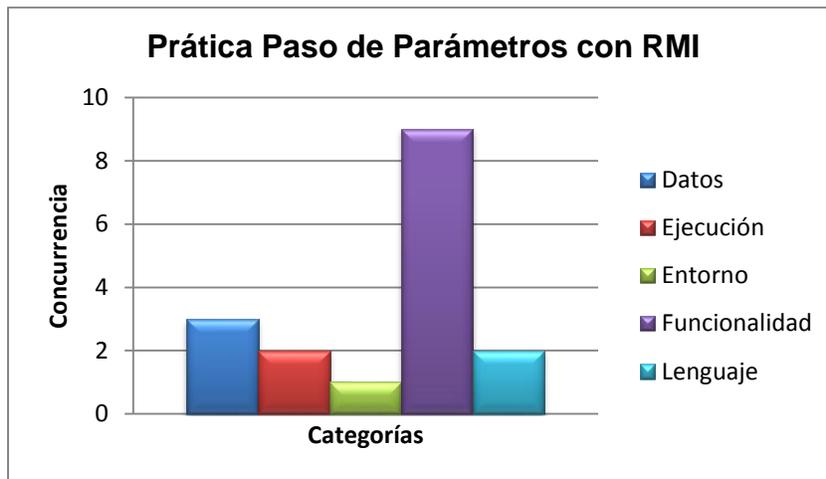
#### **4.8.2.6 Reporte de Asesorías**

A continuación se muestran las asesorías solicitadas por los estudiantes en las diferentes prácticas sobre el tema RMI (tres prácticas) asignadas en el laboratorio.

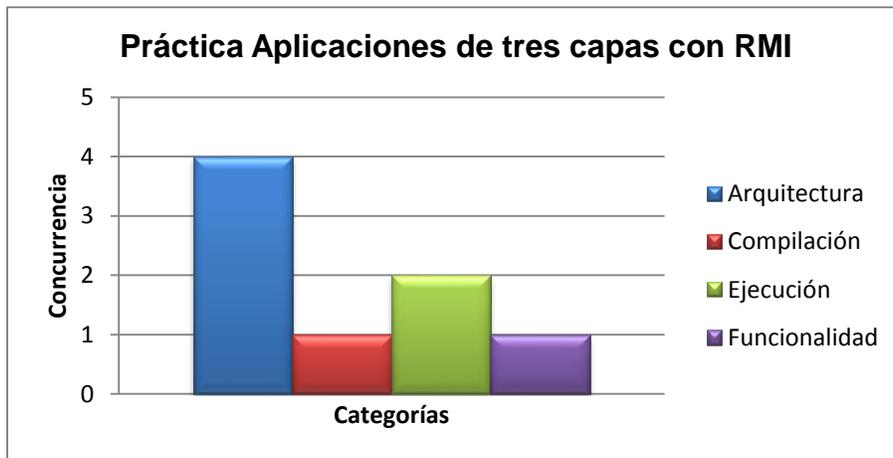
- La Categoría establece el tipo de inquietud que presenta el estudiante. Estas categorías son:
  - Entorno: inquietudes referentes a la configuración del ambiente software de trabajo.
  - Datos: inquietudes sobre los tipos de datos que se utilizan en la implementación de las prácticas.
  - Diseño: inquietudes sobre la definición de métodos o procesos de desarrollo de las prácticas.
  - Arquitectura: inquietudes sobre los componentes, diseño y la implementación de estructuras de software para cada una de las prácticas.
  - Funcionalidad: inquietudes referentes a los requerimientos que se solicitan en cada una de las prácticas solicitadas.
  - Compilación: inquietudes sobre los procesos de compilación de cada uno de los archivos que conforman las prácticas a realizar.
  - Ejecución: inquietudes sobre los procesos de ejecución de las prácticas a realizar.
  - Lenguaje: inquietudes que tienen que ver con el lenguaje de programación usado para la implementación de dichas prácticas.
- La Concurrencia se refiere al número de veces que los estudiantes solicitan una asesoría por cada categoría.



**Figura 12.** Concurrencia vs Categoría de asesorías Práctica Proceso Básico de desarrollo con RMI - LSD II periodo 2011



**Figura 13.** Concurrencia vs Categoría de asesorías Práctica Paso de Parámetros con RMI - LSD II periodo 2011



**Figura 14.** Concurrencia vs Categoría de asesorías Práctica Aplicaciones de tres capas con RMI - LSD II periodo 2011

La información tabulada correspondiente a las gráficas anteriores puede encontrarse en el Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 5.

**Conclusiones del reporte de asesorías:** En promedio, las asesorías que más se presentaron fueron las de categoría de datos y funcionalidad. Esto nos remite a suponer, que la mayoría de las inquietudes de los estudiantes provienen de problemas en el manejo del lenguaje de programación y en las funcionalidades solicitadas en los requerimientos de cada práctica.

#### 4.8.2.7 Reporte de Errores

Se reportan en esta sección los porcentajes de errores presentados por cada grupo en siete prácticas que se realizaron en el laboratorio.

- Los errores de tipo “Revisión Diseñador” son los que se presentan en el código e implementación.
- Los errores de tipo “Revisión Cliente” definen los errores presentados en la ejecución de la práctica, la perspectiva del cliente.

#### Práctica 1: Proceso Básico de desarrollo con RPC

Teniendo en cuenta el reporte de errores de los estudiantes en esta práctica (Ver reporte en el Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 6) se obtuvieron los siguientes porcentajes:

- El 53,33% de los errores en esta práctica son de tipo “Revisión Diseñador”.
- El 46,66% de los errores corresponden al tipo “Revisión Cliente”.

Los porcentajes indican que se presentaron más errores de código e implementación que de ejecución en esta práctica.

(Ver reporte de errores de las siguientes prácticas en el Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 6).

**Conclusiones del reporte de errores:** En promedio, se presentaron durante la ejecución de las prácticas más errores en el código y en la implementación que errores de ejecución. Para cada tipo de error, el registro porcentual en general fue:

- El 53,33% de los errores en esta práctica son de tipo “Revisión Diseñador”.
- El 46,66% de los errores corresponden al tipo “Revisión Cliente”.

#### 4.8.2.8 Tiempo empleado para la creación de las prácticas

En el diagnóstico se consideró el tiempo que el docente emplea para la creación de las prácticas en PDF, con el propósito de valorar el tiempo de producción del docente en la elaboración de las prácticas de laboratorio.

El tiempo promedio empleado por el docente para la creación de prácticas en PDF manualmente fue de 90 minutos.

#### **4.8.2.9 Tiempo empleado para la evaluación de las prácticas**

En el diagnóstico se consideró el tiempo que el docente emplea para la evaluación de las prácticas, con el propósito de valorar el tiempo de producción del docente en esta tarea.

El tiempo promedio empleado por el docente para la evolución de las prácticas manualmente fue de 2 horas y media, es decir 150 minutos.

#### **4.8.3 Priorizar**

La información obtenida en la valoración del curso de desarrollo intensivo de software fue el fundamento para priorizar los casos de mejora con base a las necesidades del curso.

#### 4.8.3.1 Oportunidades de Mejora

Con la anterior información recolectada como parte de la fase de Diagnóstico de Agile SPI-Process y con sesiones de trabajo con el docente se identificaron oportunidades de mejora, de cada una de ellas se representa la prioridad (Utilizando una escala de 0 a 5: 0 para ninguna influencia , 1 en el caso de ser muy influyente en la mejora del curso, 3 para una opción que influye en la mejora del curso y 5 para una influencia débil), que indica el nivel de importancia en el proceso de mejora del curso académico, la cual es usada para distribuir incrementalmente la mejora en las diferentes iteraciones.

Para definir el impacto de la oportunidad de mejora sobre el curso se tuvo en cuenta las tres áreas sobre las cuales se pretendía hacer la mejora de sus procesos: el área de preparación de las practicas por parte del docente, el área de ejecución de dichas prácticas por los estudiantes y el área de calificación de las mismas por el docente. El impacto de las mejoras está determinado por el nivel crítico de las áreas, de tal manera que a partir del diagnóstico se determinó que el área más crítica es la ejecución de las practicas ya que los estudiantes tienen más procesos involucrados y son los directamente beneficiados en la adquisición de conocimiento. Seguida de esta área esta la preparación de las guías y por último la calificación de las mismas.

A continuación se describen estas oportunidades de mejora identificadas durante el diagnóstico:

Deficiencia	Justificación	Descripción	Herramienta de soporte candidata	Prioridad
<ul style="list-style-type: none"> <li>• Falta de conocimiento y poca experiencia en comandos Linux.</li> <li>• Retraso en el envío de las prácticas.</li> </ul>	<p>Los reportes obtenidos a partir de una evaluación de conocimientos en Linux, la observación de la ejecución de prácticas previas y el reporte obtenido de asesorías y errores demuestran que los estudiantes presentan problemas de ejecución, compilación y configuración del entorno en el cual desarrollan; en ello influye que los estudiantes no tengan un correcto manejo de los comandos Linux.</p> <p>Proporcionales esta herramienta permitirá que elaboren los scripts necesarios para cada práctica con el fin de</p>	<p>Proporcionar a los estudiantes un tutorial que explique los principales comandos Linux necesarios para el desarrollo de prácticas.</p> <p>Enseñar al estudiante cómo desarrollar scripts que contengan los pasos de ejecución y compilación.</p>	Tutorial	1

	automatizar y agilizar la compilación y ejecución.			
Falta de conocimiento y poca experiencia en la compilación y ejecución de un programa en C para Linux.	Luego de la evaluación de conocimientos sobre la compilación y ejecución de un programa en C, la observación del desarrollo de ejercicios sobre el tema y el reporte obtenido de asesorías y errores se detectó que los estudiantes presentan errores respecto a estos conocimientos, lo que conlleva a tener un documento con los pasos claramente visualizados que les proporcionará un acceso fácil a estos parámetros y una mejor oportunidad de aprendizaje.	Ofrecer un tutorial de compilación y ejecución de un programa en C para Linux, de tal manera que el estudiante recuerde y practique los conocimientos necesarios.	Tutorial	1
<ul style="list-style-type: none"> <li>• Uso inadecuado de la gestión de memoria dinámica en C.</li> <li>• Errores de sintaxis en la implementación de las prácticas referentes al lenguaje de programación Java.</li> </ul>	Las observaciones realizadas y los reportes obtenidos a partir de la evaluación de conocimientos en el lenguaje C y lenguaje Java y el reporte obtenido de asesorías y errores demuestran que se comenten errores en la sintaxis del lenguaje de programación.	Proporcionar un tutorial de punteros en C y un tutorial de Java que les permita recordar conocimientos adquiridos.	Tutorial	1
Muchos estudiantes no llegan puntualmente al inicio de la clase y pierden la explicación proporcionada por el profesor.	A partir de las observaciones de cada una de las prácticas que se realizan en el curso se pudo diagnosticar que los estudiantes al llegar tarde al inicio de la clase, perdían la explicación de lo solicitado para dicha práctica por parte del docente, por ello se hace necesario un video como una ayuda visual que puede apoyar la comprensión de los requerimientos, además de ser reproducido cuantas veces sea necesario para recordar lo que se debe realizar.	Proveer un video explicativo de cada una de las tecnologías usadas en las prácticas, donde se explique la manera de compilar y ejecutar un programa.	Video	2
• Trabajo no colaborativo en el	Observaciones y reportes de las notas de sustentaciones individuales de los parciales prácticos,	Entregar a un estudiante el ejecutable del servidor y al otro	Servidor y Clientes de	1

<p>grupo.</p> <ul style="list-style-type: none"> <li>• En la pareja, un estudiante toma toda la responsabilidad de la práctica.</li> <li>• Dificultad para apropiarse los requerimientos.</li> </ul>	<p>demuestran que generalmente un estudiante del grupo es quien realiza la mayor parte o toda la implementación de la práctica.</p> <p>Además de ayudar en la comprensión de los requerimientos, ya que los reportes demuestran errores de lógica que definen la incompreensión frente al requerimiento asignado.</p>	<p>estudiante el ejecutable del cliente, ambos ya realizados. Esto con el fin de dividir el trabajo (Asignación de roles).</p>	<p>Prueba</p>	
<p>Presencia de los mismos errores en diferentes prácticas.</p>	<p>Luego del análisis de cada una de las prácticas presentadas por cada grupo se detectó que los estudiantes presentaban repetidamente los mismos errores en las diferentes prácticas, lo cual conlleva a determinar que la retroalimentación del trabajo realizado por el grupo es de gran importancia para la corrección de errores y la percepción de las debilidades que los estudiantes presentan en la implementación de las prácticas para una mejora en una próxima sesión.</p>	<p>Documento que además de brindar la calificación de la práctica, presente las principales observaciones de las inconsistencias, errores y dificultades.</p>	<p>Documento de Retroalimentación</p>	<p>3</p>
<p>Dificultad para apropiarse los requerimientos de cada práctica</p>	<p>A partir de la observación y de las entrevistas con los estudiantes se pueden determinar que existen dificultades para la apropiación de los requerimientos solicitados en cada práctica, para ello se hace necesario que los estudiantes realicen un modelo de secuencia que muestre el paso de mensajes entre los usuarios y las funcionalidades que se deben crear en cada práctica.</p>	<p>Solicitar a los estudiantes realizar un modelo de secuencia de cada una de las prácticas, lo que permita aplicar una fase de diseño antes de llegar a la implementación.</p>	<p>Modelo de secuencia</p>	<p>3</p>
<p>Retraso en la implementación de las prácticas debido a la falta de configuración de las plataformas y entornos necesarios.</p>	<p>La observación del retraso de los estudiantes en la elaboración de las prácticas debido a la falta de configuración de plataformas y entornos, mostró la necesidad de que se realizara una configuración previa de los equipos que permite que los estudiantes no retrasen su trabajo a causa de problemas de este tipo.</p>	<p>Proveer al monitor de la sala una guía de configuración tanto de las plataformas y los entornos necesarios para la elaboración de las prácticas por parte de los estudiantes.</p>	<p>Guía de configuración de equipos</p>	<p>4</p>

Repetición de partes de las guías y demora en su creación.	A partir del análisis de las guías realizadas por el docente y de observar que existe cierta demora en la creación a causa de la repetición de ciertas partes de las mismas y su extensión, se hace necesario crear una ayuda para que el docente tome menos tiempo en la creación de las guías necesarias para cada una de las prácticas.	Una herramienta que le permita al docente una generación automática de las guías necesarias del laboratorio.	Generación de Guías en PDF	5
Falta de experiencia y acercamiento con cada una de las tecnologías realizadas en las prácticas.	Con la observación del desarrollo de cada una de las prácticas realizadas por los estudiantes, de los errores que se cometían y de la falta de conocimiento en las tecnologías planteadas, es necesario que se tenga una base de prácticas realizadas por anteriores estudiantes, que permita visualizar con anterioridad, cómo se abordan los requerimientos según la tecnología a usar; lo cual les permita desarrollar con mayor facilidad y en un menor tiempo.	En la página disponible para el curso, poner al acceso de los estudiantes las mejores prácticas de cada una de las tecnologías desarrolladas por los estudiantes de semestres anteriores.	Repositorio de las mejores prácticas	3
Retraso en la entrega oportuna de las notas por parte del docente.	A partir del análisis del tiempo que tarda el docente para la entrega de notas y observaciones correctivas a los estudiantes después de la realización de una práctica o la ejecución de un requerimiento, se hace necesario crear un soporte de evaluación. Este soporte de evaluación permitirá que el docente tome menos tiempo en la evaluación de los módulos especificados, en la compilación y ejecución de los mismos para determinar si cumplen con los objetivos de la práctica o el requerimiento respectivamente.	Ejecutable de evaluación para los módulos servidor, cliente, interfaz.	Script (ejecutable) para evaluación	5

Tabla 9. Oportunidades de mejora

#### **4.8.4 Construir un plan general de mejora**

Se creó un plan general de mejora que permitió pilotear el curso de desarrollo intensivo de software a través de todo el proyecto de mejora; además, en este plan se recomendó incluir información referente a cada uno de los casos de mejora que se están iniciando, ejecutando o terminando; como ayuda para mantener una organización de todo el proyecto de mejora.

##### **4.8.4.1 Plan de Implementación**

El diagnóstico realizado permitió evaluar el proceso de desarrollo de software, lo que significó poder medir el estado actual de los procesos de desarrollos del LSD, de manera que se conocieron sus fortalezas, riesgos y debilidades. Como resultado de este diagnóstico se generó un Plan de Mejoramiento (Plan de Implementación) adecuado, que permita solventar las necesidades encontradas.

El propósito del Plan de Implementación es permitir organizar, controlar y administrar la documentación generada, además de guiar las oportunidades de mejora de los procesos y la definición del estimado del tiempo y del esfuerzo requerido.

Dentro del proceso de administración, se especifica:

##### **4.8.4.1.1 Carta Gantt del proyecto**

Para prever la dedicación, esfuerzo, tiempo y recursos necesarios en las diferentes tareas o actividades que se realizaron en el transcurso del proyecto de mejora y como herramienta básica en la gestión del mismo, se definió una Carta Gantt que identifica cada una de dichas actividades, las relaciones entre ellas, el tiempo de su ejecución y los responsables de cada actividad.

Las fases descritas en la Carta Gantt fueron una guía para cumplir con los objetivos propuesto de acuerdo a los tiempos establecidos por el equipo de trabajo, de manera que durante el transcurso del proyecto se evaluó constantemente el cumplimiento de ellos en función de lo programado para asegurarse de que se iba desarrollando de acuerdo al plan previsto.

##### **4.8.4.1.2 Plan de Administración de Mejoras**

Para administrar las oportunidades de mejora que fueron implementadas se partió de las oportunidades de mejora detectadas y priorizadas obtenidas del diagnóstico, definiendo así iteraciones en las cuales se establecieron actividades que se llevaron a cabo según el proceso especificado.

##### **4.8.4.1.3 Agenda del Plan de Control**

Para tener un control del proyecto, semanalmente se realizó una revisión del avance de cada una de las actividades realizadas comparándolas contra los planes definidos en cada una de las fases. Se efectuaron acciones correctivas en caso de existir desvíos en los cuales se consideraban re-planificar las actividades anteriormente programadas.

#### **4.8.4.1.4 Plan de Medición**

El control de las horas destinadas al proyecto fue importante para llevar a cabo con efectividad el plan de mejora en el LSD. Para ello, los involucrados en el proyecto debieron especificar o realizar una autoevaluación de las horas para este fin después de cada reunión y de cada actividad efectuada. Por otro lado, fue necesario medir después de cada implementación de área de proceso, la cantidad y nivel de elementos reusados, así como el esfuerzo de la adaptación de Agile SPI al entorno académico.

(Para ver el Plan de implementación completo y la Carta Gantt referirse al Anexo D Digital Plan de Implementación).

#### **4.8.5 Comunicar el plan general de mejora**

La comunicación del plan general de mejora contribuyó en gran parte a que los estudiantes del curso de desarrollo intensivo de software estuvieran informados acerca de todo el proyecto de mejora y de la manera como éste se planeó.

La estrategia que se utilizó para comunicar el plan de mejora, en un principio, fue la exposición del docente frente a los estudiantes sobre el proyecto que se estaba realizando y los principales aportes de la mejora sobre el curso de desarrollo intensivo de software LSD. Por otro lado, se especifica en un enlace de la aplicación web denominado “Plan de Mejora”, los objetivos y las razones por las que se estaba llevando a cabo la implementación de este plan de mejora, y la planeación de su puesta en marcha.

## Fase: Formulación

---

En ésta fase se tomaron las necesidades de implementación más prioritarios para introducir mejoras vía la implementación del proceso según los resultados arrojados por la valoración. En esta fase se realizaron las primeras iteraciones de implementación; con base en los resultados de implementación se realizaron la planificación de las demás iteraciones de implementación. Esta primera iteración tiene el objetivo de medir el esfuerzo que se usó con el fin de estimar de una manera más real el esfuerzo, costo y tiempo que las demás iteraciones de implementación requieren.

### 4.9 ACTIVIDADES PARA LA FORMULACIÓN

Para la realización de la fase de formulación dentro de Agile SPI – Process, es preciso seguir una serie de actividades que permitan implementar las mejoras; dichas actividades realizadas para cumplir el objetivo son:

#### 4.9.1 Planificar la iteración de mejora

En esta actividad se describieron los objetivos de la iteración y se revisaron las oportunidades de mejora descritas.

##### Objetivos:

- Realizar las primeras iteraciones de implementación a partir de los datos obtenido en el diagnóstico.
- Planificar las iteraciones que se llevaran a cabo dentro del desarrollo del proyecto de mejora.
- Medir el esfuerzo en términos del tiempo que las iteraciones gastaran en su ejecución, con el fin de tener una estimación para su desarrollo.

#### 4.9.2 Evaluar y analizar procesos

Se evaluaron en profundidad las partes del proceso a ser mejoradas en la iteración, de manera que se entendieran en el contexto de los procesos actuales. Para ello se incluyeron a líderes usuarios del proceso para que se conocieran los requisitos propios.

Para evaluar los procesos que se llevan a cabo dentro del curso y descubrir las oportunidades de mejora, se definieron los siguientes roles: observadoras del proceso o ejecutoras del proyecto y el docente experto del curso. Teniendo los roles designados se usó como técnicas para evaluar los procesos:

- Observación Directa: la asistencia a cada una de las prácticas realizadas dentro del curso permitió definir cuáles eran los procesos que se llevaban a cabo en cada sesión guiada por el docente.
- Experiencia: las ejecutoras del proyecto al conocer y haber cursado la materia del caso de estudio permitió que los procesos fueran establecidos según el conocimiento que se tenía de ellos.
- Entrevistas informales: se realizaron entrevistas informales al docente y a los estudiantes para conocer las actividades, roles, y demás procesos realizados.
- Revisión del modelo de procesos obtenido en el proyecto “Modelado y Propuesta de Mejora de Procesos de Desarrollo para un entorno académico” [6], de manera que sirvió de guía para ver las posibles mejoras al mismo.

#### 4.9.3 Diseñar procesos

En esta actividad se llevaron a cabo tareas de diseño (modelado) para la implementación de las necesidades de implementación. El resultado es un conjunto actividades, roles, productos de trabajo, plantillas recomendaciones, y demás elementos que provean descripción del área de proceso.

En este proyecto se pretende realizar tres iteraciones dentro de las cuales se llevan a cabo diferentes implementaciones de mejora, dichas iteraciones están definidas de la siguiente manera:

##### 4.9.3.1 Iteración 1

Esta iteración se realiza en el proceso de preparación que hace parte del curso de desarrollo intensivo de software, donde se lleva a cabo la planeación por parte del docente de las prácticas que en dicho curso se ejecutan.

Nombre Actividad:	<b>Aplicación para generación de guías en PDF</b>
Descripción:	La aplicación web realizada para el proyecto, en la sesión del docente, dispone una funcionalidad para la generación de las guías de laboratorio en PDF.
Artefacto:	Generación de guías en PDF sobre una aplicación web.
Estrategia de aplicación:	El docente accede a su sesión en la aplicación web; una vez iniciada la sesión, en el enlace “Guías” tendrá la posibilidad de elegir la generación de la guía en PDF clasificada por tema. Después de seleccionar el tipo de guía, se despliega el formulario que solicita el nombre de la práctica, imágenes y demás

	información pertinente. Finalmente el docente solicita generar el PDF y la aplicación le provee la guía descrita.
Encargado de realizar la actividad:	Docente
Anexo donde se encuentra:	Anexo E: Herramientas de Mejora, Sección 2: Aplicación Web.

**Tabla 10.** Actividad aplicación para generación de guías en PDF

Para ver la definición de las demás mejoras de la iteración 1 y de cada una de las siguientes iteraciones referirse al referirse al Anexo E: Herramientas de Mejora, Sección 1: Definición de mejoras.

(Para ver las mejoras que fueron adoptadas de la norma ISO- 29910 referirse al Anexo E: Herramientas de Mejora, Sección 9: Mejoras obtenidas norma técnica 29110)

#### **4.9.4 Adoptar procesos**

Después de realizar el diseño de cada una de las mejoras a implementarse, el siguiente paso es llevarlas a cabo. Por ello, se aplicaron las mejoras definidas a los procesos del curso en forma preliminar, para luego generar los nuevos procesos resultantes.

## Fase: Mejora

---

En esta fase de Agile SPI-Process se implementan los procesos a partir de los repositorios disponibles y se gestiona todo el esfuerzo de las necesidades de implementación a través de iteraciones. Además se realizan iteraciones con cada una de las áreas de proceso a mejorar o a crear, para cada una de las cuales se efectúa un plan del caso de mejora basándose en la información recolectada del o de los piloto de mejora ejecutados, esto con el objetivo de realizar estimativos de recursos como personal, tiempo y dinero.

Las diferentes iteraciones se realizaron de acuerdo a lo planeado en la fase de formulación dependiendo de la priorización establecida.

Luego de la aplicación de las mejoras planteadas en el capítulo anterior, en cada una de las iteraciones se obtuvieron los siguientes resultados, los cuales permitieron determinar los procesos que en la implementación de las mejoras presentaron efectos positivos.

### **4.10 RESULTADOS IMPLEMENTACIÓN DE MEJORA**

#### **4.10.1 Iteración 1**

Esta iteración se realizó en el proceso de preparación que hace parte del curso de desarrollo intensivo de software, donde se lleva a cabo la planeación por parte del docente de las prácticas que en dicho curso se ejecutan.

##### **4.10.1.1 Aplicación para generación de guías en PDF**

La aplicación web realizada para el proyecto, permitió al docente la generación de las guías de laboratorio en PDF. Para generar las guías, la aplicación dispone de diferentes opciones por cada tema del LSD. Posteriormente después que el docente elija el tipo de práctica para generar el PDF, se le solicita las gráficas y demás información pertinente.

Se evaluó el tiempo promedio empleado por el docente para la creación de prácticas en PDF con la aplicación web, dicho tiempo fue de 20 minutos.

##### **4.10.1.2 Aplicación de una guía y demo de configuración de equipos**

Antes de cada una de las prácticas se realizó una guía de configuración de los equipos de la sala que los estudiantes usarían y de un demo que probara la correcta funcionalidad de las mismas, las cuales eran ejecutadas por el monitor de la sala el cual era el encargado de verificar que los solicitado en cada computador se cumpliera a cabalidad.

Después de aplicar lo correspondiente y recibir la lista de chequeo, se obtuvo que el 100% de los computadores contaba con las herramientas necesarias.

(Ver la lista de chequeo de los computadores de la sala en el Anexo F Digital Resultados Implementación de Mejoras, Sección 1).

#### **4.10.1.3 Tutorías para cada una de las tecnologías**

Para comprobar la efectividad de la aplicación de los tutoriales se realizaron evaluaciones de conocimiento antes y después de la ejecución de cada uno de los tutoriales y sus respectivos ejercicios.

Los refuerzos realizados a los estudiantes antes de realizar las prácticas correspondientes a cada tecnología fueron los siguientes:

##### **4.10.1.3.1 Comandos en Linux**

Al inicio del curso se realizó la evaluación de conocimiento sobre comandos usados en Linux, lo que permitió determinar el nivel de conocimiento que los estudiantes tenían antes de iniciar las prácticas que necesitaban de estos temas.

Los resultados obtenidos a partir de la evaluación previa en el I periodo del 2012 fueron los siguientes:

- El promedio de preguntas correctas que tuvieron los estudiantes es de 58,12%.
- El promedio de preguntas incorrectas que tuvieron los estudiantes es de 16,88%.
- El promedio de preguntas no contestadas que tuvieron los estudiantes es de 25%.

Los resultados obtenidos a partir de la evaluación después de la aplicación del instrumento en el I periodo del 2012 fueron los siguientes:

- El promedio de preguntas correctas que tuvieron los estudiantes es de 62,15%.
- El promedio de preguntas incorrectas que tuvieron los estudiantes es de 11,4 %.
- El promedio de preguntas no contestadas que tuvieron los estudiantes es de 26,48%.

Los resultados obtenidos a partir de la evaluación previa en el II periodo del 2012 fueron los siguientes:

- El promedio de preguntas correctas que tuvieron los estudiantes es de 44,16%.
- El promedio de preguntas incorrectas que tuvieron los estudiantes es de 17,98%.
- El promedio de preguntas no contestadas que tuvieron los estudiantes es de 37,89%.

Los resultados obtenidos a partir de la evaluación después de la aplicación del instrumento en el II periodo del 2012 fueron los siguientes:

- El promedio de preguntas correctas que tuvieron los estudiantes es de 55,49%.
- El promedio de preguntas incorrectas que tuvieron los estudiantes es de 4,71%.
- El promedio de preguntas no contestadas que tuvieron los estudiantes es de 39,85%.

(Para ver la evaluación antes y después de instrumento sobre Comandos en Linux referirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 2 y Sección 3 respectivamente. Para ver los resultados de los estudiantes en cada una de las preguntas realizadas referirse al mismo Anexo en la Sección 4 para el I periodo 2012 y Sección 6 para el II periodo 2012).

Al entregarles a los estudiantes el tutorial con los conocimientos básicos y necesarios sobre los comandos en Linux, cada uno de ellos leía y aplicaba lo que ahí encontraba en una práctica solicitada al final de dicho tutorial. Los resultados obtenidos son los siguientes:

Para el I periodo del 2012, los resultados generales son:

#### Ejercicio 1:

Los estudiantes en el primer ejercicio en general tuvieron los siguientes resultados:

- Entrega Completa: 66,7 %
- Entrega Incompleta: 33,3 %
- No entrego nada: 0 %

Los resultados indican que la mayoría de los estudiantes entregaron completamente el ejercicio solicitado.

#### Ejercicio 2:

Los estudiantes en el segundo ejercicio en general tuvieron los siguientes resultados:

- Entrega Completa: 22,2 %
- Entrega Incompleta: 33,3 %
- No entrego nada: 44,5 %

Los porcentajes indican que la mayoría de los estudiantes no entregaron el ejercicio solicitado.

(Para ver los resultados de la práctica del tutorial aplicado sobre Comandos en Linux referirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 5).

Los porcentajes calculados anteriormente se realizaron de igual forma para los siguientes tutoriales descritos en esta sección.

#### **4.10.1.3.2 Compilación y ejecución de un programa en C**

Para ver la evaluación antes y después de instrumento sobre compilación y ejecución de un programa en C en Linux referirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 7 y Sección 8 respectivamente. Para ver los resultados de los estudiantes en cada una de las preguntas realizadas referirse al mismo Anexo en la Sección 9 para el I periodo 2012 y Sección 11 para el II periodo 2012.

Para ver los resultados de la práctica del tutorial aplicado sobre compilación y ejecución de un programa en C en Linux referirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 10.

#### **4.10.1.3.3 Conocimientos en lenguaje C**

Para ver la evaluación antes y después de instrumento sobre conocimientos en lenguaje C referirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 12 y Sección 13 respectivamente. Para ver los resultados de los estudiantes en cada una de las preguntas realizadas referirse al mismo Anexo en la Sección 14 para el I periodo 2012 y Sección 16 para el II periodo 2012.

Para ver los resultados de la práctica del tutorial aplicado sobre conocimientos en lenguaje C referirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 15.

#### **4.10.1.3.4 Conocimientos en lenguaje Java**

Para ver la evaluación antes y después de instrumento sobre conocimientos en lenguaje Java referirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 17 y Sección 18 respectivamente. Para ver los resultados de los estudiantes en cada una de las preguntas realizadas referirse al mismo Anexo en la Sección 19 para el I periodo 2012 y Sección 21 para el II periodo 2012.

Para ver los resultados de la práctica del tutorial aplicado sobre conocimientos en lenguaje Java referirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 20.

#### **4.10.1.4 Entrega de videos del desarrollo de prácticas**

Se entregó un video por cada una de las prácticas básicas de las tecnologías usadas en el laboratorio, los cuales fueron puestos a disposición de los estudiantes en la página destinada para la información que el docente entrega a los estudiantes.

#### **4.10.1.5 Repositorio de prácticas**

En la página destinada para los estudiantes el docente dispuso un repositorio de las mejores prácticas de cursos anteriores, prácticas que fueron usadas como guías para el desarrollo que se realizaba por parte de los estudiantes.

#### 4.10.1.6 Entrega de Servidor y Cliente de prueba

El docente por cada una de las prácticas entregó un cliente y servidor de prueba que permitiera que los estudiantes pudieran probar sus archivos creados, de tal manera que sus casos de prueba pudieran ser validados.

#### 4.10.1.7 Entrega de documento de retroalimentación

Por cada una de las prácticas el docente entrego un documento de retroalimentación donde se consignaban los errores y falencias por cada uno de los archivos creados por los estudiantes, lo cual permitía tener en cuenta los errores cometidos para próximas prácticas.

### 4.10.2 Iteración 2

Esta iteración se realizó en el proceso de ejecución de las prácticas, donde los estudiantes preparan y llevan a cabo el desarrollo de las mismas.

#### 4.10.2.1 Aplicación web que permita al docente ver tiempos de desarrollo de las prácticas y requerimientos

Para cada una de las prácticas desarrolladas por los estudiantes se tomó el tiempo que estos empleaban en la realización de los archivos correspondientes para las mismas por medio de una aplicación web, la cual era accedida por docente para monitorear el proceso de desarrollo de los estudiantes.

#### Primer periodo del 2012:

Para el primer periodo del 2012, las prácticas en las cuales se registraron estos tiempos fueron:

- Práctica 1: Proceso Básico de desarrollo con RPC
- Práctica 2: XDR y RPC
- Práctica 3: Proceso Básico de desarrollo con RMI
- Práctica 4: Paso de Parámetros con RMI
- Práctica 5: Callback con RMI
- Práctica 6: Proceso Básico de Desarrollo con CORBA
- Práctica 7: CORBA e IDL
- Práctica 8: Invocación Dinámica con CORBA
- Práctica 9: Callback con CORBA

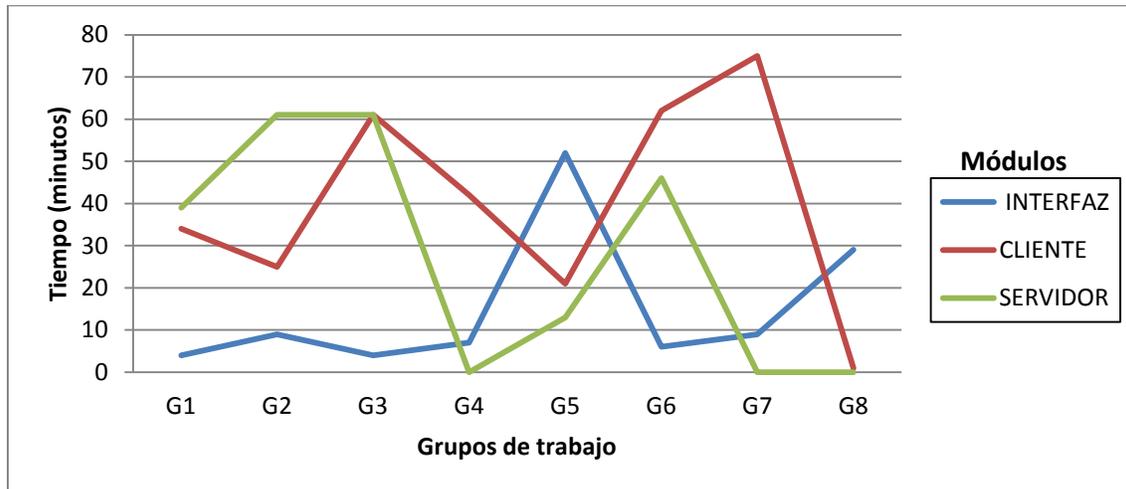
### Análisis de tiempos por práctica

#### Práctica 1: Proceso Básico de desarrollo con RPC

GRUPO	INTERFAZ	CLIENTE	SERVIDOR
<b>G1</b>	4	34	39
<b>G2</b>	9	25	61
<b>G3</b>	4	61	61

<b>G4</b>	7	42	0
<b>G5</b>	52	21	13
<b>G6</b>	6	62	46
<b>G7</b>	9	75	0
<b>G8</b>	29	1	0

**Tabla 11.** Registro de tiempos de la práctica Proceso Básico con RPC - LSD I periodo 2012



**Figura 15.** Tiempo (min) vs grupos de la práctica Proceso Básico de desarrollo con RPC - LSD I periodo 2012

Basándose en la anterior tabla y su respectiva gráfica, se puede observar que en la primera práctica referente al proceso básico que se sigue en RPC, los estudiantes en general emplearon menos tiempo en la realización de la interfaz, seguido del tiempo empleado en el servidor y por último en el cliente.

En esta y todas las prácticas siguientes se hizo uso de la aplicación web creada para la recolección de los tiempos, la cual no fue usada por todos los equipos de trabajo por lo que no permitió tener medidas precisas que se pudieran analizar en algunos casos.

Las posteriores prácticas con su respectiva tabla, gráfica y análisis se encontrarán en el Anexo F Digital Resultados Implementación de mejoras, Sección 23.

Para ver las tablas, gráficas y análisis de los tiempos empleados por los estudiantes en las prácticas realizadas en el segundo periodo del 2012 referirse al Anexo F Digital Resultados Implementación de mejoras, Sección 24

### **Análisis de tiempos por requerimientos**

Los estudiantes presentan por cada una de las tecnologías un requerimiento el cual es sustentado en una fecha establecida por el docente, durante un rango de tiempo de media hora por cada grupo los estudiantes realizan la sustentación de lo que realizaron mediante el desarrollo de un nuevo requerimiento que verifica que

cada uno de los integrantes tenga pleno conocimiento del código de la aplicación entregada. Para esta sustentación se midió el tiempo que cada estudiante empleaba en la realización del mismo.

A partir de una lista de chequeo que se aplicó a cada estudiante en su sustentación se obtuvieron los siguientes tiempos y observaciones (ver la plantilla de la lista de chequeo de sustentación en el Anexo F Digital Lista de Chequeo Sustentación de requerimientos, Sección 26):

### Requerimiento 1: RPC

	INTERFAZ	CLIENTE	SERVIDOR
Estudiante 1	11	50	30
Estudiante 2	8	0	0
Estudiante 3	8	16	0
Estudiante 4	5	11	29
Estudiante 5	18	75	85
Estudiante 6	10	40	50
Estudiante 7			
Estudiante 8			
Estudiante 9	10	63	0
Estudiante 10	13	12	40
Estudiante 11	0	65	0
Estudiante 12	3	15	27
Estudiante 13	4	30	5
Estudiante 14	15	45	53
Estudiante 15	22	0	0
Estudiante 16	20	5	35

Tabla 12. Registro de tiempos del Requerimiento RPC - LSD I periodo 2012

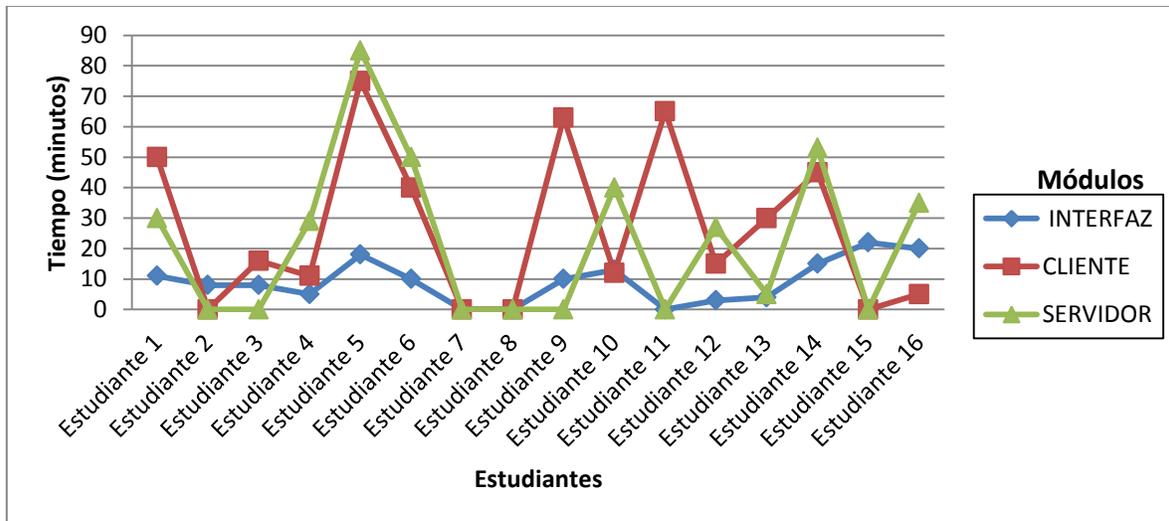


Figura 16. Tiempo (min) vs grupos Requerimiento RPC- LSD I periodo 2012

Basándose en la anterior tabla y su respectiva gráfica, se puede observar que en la primera sustentación del requerimiento referente a RPC, los estudiantes en

general gastan más tiempo en la elaboración del servidor, siguiendo el cliente y por último de la interfaz.

Los posteriores requerimientos con su tabla, gráfica y análisis respectivo, se encontrarán en el Anexo F Digital Resultados Implementación de Mejoras, Sección 23.

#### **4.10.2.2 Aplicación del diseño de un modelo de secuencia**

Para las prácticas de RPC y RMI desarrollados por los estudiantes se solicitó entregar un modelo de secuencia como parte del diseño necesario antes de un desarrollo realizado para cada una de dichas prácticas.

#### **4.10.2.3 Definición de roles y responsabilidades durante el desarrollo**

Para las prácticas realizadas se tuvo una estrategia de repartición de roles y responsabilidades por cada estudiante perteneciente a un grupo de trabajo, lo cual permitió que el desarrollo no se realizara por solo uno de los estudiantes.

En los requerimientos también se solicitó la definición de roles (ver la definición de roles para los requerimientos en el Anexo F Digital Resultados Implementación de Mejoras, Sección 27).

#### **4.10.2.4 Estrategia para la administración de versiones**

La estrategia de administración de versiones usada por los estudiantes para la realización de sus requerimientos de cada tecnología fue escogida por cada uno de los grupos, la cual sirvió de mecanismo para que se llevara un recuento del trabajo realizado y de los errores que cometían en el desarrollo de lo necesario para sus respectivos requerimientos.

(Ver la definición de las estrategias para la administración de versiones en los requerimientos en el Anexo F Digital Resultados Implementación de Mejoras, Sección 27).

#### **4.10.2.5 Identificación de tareas a realizar durante el requerimiento**

Para cada uno de los requerimientos se entregó una lista especificando las actividades que se debían desarrollar dentro de dichos requerimientos lo cual permitió a los estudiantes tener claro los objetivos a cumplir al finalizar y realizar la respectiva entrega de lo solicitado.

#### **4.10.2.6 Construcción y ejecución preliminar de casos de prueba**

Para cada uno de los requerimientos se solicitó a los estudiantes realizar una lista preliminar de casos de prueba, de tal manera que se pudiera comprobar cuáles eran los casos que el requerimiento cumplía y cuáles no, facilitando así la revisión del docente de lo entregado por cada grupo (ver plantilla de requisitos de los entregables en el Anexo F Digital Resultados Implementación de Mejoras, Sección 27).

### **4.10.3 Iteración 3**

Esta iteración se realizó en el proceso de evaluación de las prácticas entregadas por los estudiantes, donde el docente hace la revisión de las mismas en base a los objetivos previamente establecidos y de acuerdo a ello define una calificación.

#### **4.10.3.1 Scripts para la agilidad en la calificación de las prácticas**

Para la evaluación de las prácticas de laboratorio se realizaron scripts de compilación y ejecución dependiendo de la tecnología utilizada y a ser evaluada, con el fin de apoyar la labor del docente.

Se evaluó el tiempo promedio empleado por el docente para la evaluación de las prácticas utilizando los scripts, dicho tiempo fue de una hora y media, es decir, 90 minutos.

#### **4.10.3.2 Formalización de la entrega con una lista de soporte de aceptación**

Para cada uno de los requerimientos entregados se realizó una lista de aceptación que permitió validar la entrega completa, incompleta o incorrecta de lo solicitado en las tecnologías desarrolladas en cada uno de los cortes (ver la plantilla de la lista de chequeo de sustentación en el Anexo F Digital Lista de Chequeo Sustentación de requerimientos, Sección 26).

## **4.11 ENCUESTA REALIZADA A LOS ESTUDIANTES DESPUÉS DE LA APLICACIÓN DE LAS MEJORAS**

Al finalizar el curso se realizó una encuesta a los estudiantes del segundo periodo del 2012 que cursaron la materia, la cual fue necesaria para evaluar el efecto que tuvieron las mejoras realizadas sobre los estudiantes y los resultados que obtuvieron en su aprendizaje y calidad de prácticas entregadas.

Para la validación de las mejoras aplicadas a cada uno de los procesos y la correcta aceptación por parte de los estudiantes, se realizó una encuesta entregada por 10 estudiantes y los resultados obtenidos por cada pregunta realizada son los siguientes:

- El 80% de los estudiantes requiere leer más de una vez la guía de la práctica que el docente entrega para su realización.
- El 60% de los estudiantes requiere información adicional a la encontrada en página del laboratorio, para la solución de cada práctica.
- El 80% de los estudiantes requiere más tiempo del asignado para la realización de cada una de las prácticas; el promedio de tiempo extra que según los estudiantes se necesita es de 45 minutos.
- Para el 100% de los estudiantes es esencial el acompañamiento del docente para la resolución de dudas.
- Para el 70% de los estudiantes la especificación del problema que se desarrolla en la guía es claro.

- El 70% de los estudiantes cree que los problemas que se desarrollan en la guía son complejos.
- Para el 100% de los estudiantes la primera práctica de cada tecnología es el primer acercamiento que tienen con la misma.
- El 100% de los estudiantes aseguran que los equipos en los que trabajan tienen configurado de manera adecuada cada una de las herramientas necesarias para el desarrollo de las prácticas.
- El 100% de los estudiantes realiza una preparación previa para el desarrollo de la práctica del laboratorio; las herramientas más usadas son: 10% de los estudiantes hace uso de los videos del desarrollo de las prácticas, el 60% realiza un preparación teórica, el 40% realiza el demo antes de las prácticas, el 70% realiza una revisión de las prácticas de anteriores semestres y el 20% de los estudiantes pide una asesoría de un compañero y realiza búsqueda de los temas en la web.
- El 20% de los estudiantes creen que las dudas en el desarrollo de la práctica son esencialmente generadas por el problema al que se enfrentan, el 40% debido a los conocimientos del lenguaje de programación, el 70% a la tecnología usada, y el 10% de los estudiantes piensas que es por desconocimiento de librerías y comandos.
- Según los estudiantes entre las estrategias de asignación de roles en el grupo de trabajo que adoptan para desarrollar la práctica son:
  - Estudiar el enunciado de la práctica y de acuerdo eso elegir el rol de cada uno.
  - Turnarse los roles en cada práctica.
  - Los roles se asignan a la suerte de cada estudiante.
  - Mutuo acuerdo en la asignación de roles.
  - Apoyo mutuo: el estudiante que terminó su tarea de acuerdo al rol, ayuda al otro estudiante.
  - El estudiante que haya estudiado de antemano el cliente se encarga de la lógica del mismo, de igual manera para el servidor.
  - Se escoge a quien le va mejor con determinado tema y se reparte el trabajo.
- Entre las razones dadas por los estudiantes de la no entrega de las prácticas están:
  - Falta de tiempo para realizar la práctica.
  - Problema con el demo entregado.
  - Problemas en la comprensión del requerimiento solicitado.
  - Errores de compilación.

(Para ver la encuesta y sus resultados referirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 28).

## 4.12 CRITERIOS PARA EVALUAR LA MEJORA

Para determinar el efecto sobre el curso LSD de la mejora, se utilizaron como criterios adicionales, los siguientes:

### 4.12.1 Notas parciales y totales de los estudiantes

Uno de los criterios usados para determinar que las mejoras surgieron efecto en los estudiantes fue las notas obtenidas en cada uno de los cortes, lo cual pudo demostrar que las mejoras implementadas con sus herramientas mejoraron la calidad de conocimiento adquirido por los estudiantes, reflejado en notas más altas en cada corte.

#### Primer periodo del 2012:

El curso de LSD se compone de tres cortes, para cada uno de los cuales se realizan prácticas referentes a los temas de RPC<sup>24</sup>, RMI<sup>25</sup> y CORBA<sup>26</sup> respectivamente. El primer y segundo corte sumados equivalen al 70% de la nota total y el tercer corte al 30% restante.

El criterio de evaluación que se tiene en cuenta para cada una de las prácticas es el cumplimiento de las funcionalidades solicitadas. Mientras que los criterios que se tienen en cuenta para la calificación del requerimiento son: manuales, aplicación y el factor de sustentación (Dicho factor en este semestre se promedia con la nota obtenida entre los manuales y la entrega de la aplicación. Anteriormente el factor de sustentación era una nota que variaba entre 0 – 1, la cual era ponderada con la nota obtenida entre los manuales y la entrega de la aplicación).

Los estudiantes del primer periodo del 2012 que cursaron la materia obtuvieron notas para cada una de las prácticas realizadas, a partir de las cuales se obtuvieron las siguientes notas finales:

(Ver la discriminación de las notas de cada uno de los cortes en el Anexo F Digital Resultados Implementación de Mejoras, Sección 29).

- **Notas Finales**

La nota final proviene de las notas finales de cada uno de los cortes, teniendo en cuenta que el primer y el segundo corte equivalen a un 35% cada uno y el tercer corte a un 30% de la nota final.

---

<sup>24</sup> RPC: Remote Procedure Call,

<sup>25</sup> RMI: Java Remote Method Invocation

<sup>26</sup> CORBA: Common Object Request Broker Architecture

Las notas finales iguales o superiores a 3,0 indican que la materia es aprobada y consecuentemente, las notas inferiores a dicho valor indican que la materia es reprobada.

Teniendo en cuenta las notas finales de los estudiantes (Ver la discriminación de las notas en el Anexo F Digital Resultados Implementación de Mejoras, Sección 29) se obtuvieron las siguientes observaciones:

**Observaciones:**

- El 88,2% de los estudiantes tuvieron notas finales superiores o iguales a 3,0 lo que corresponde a los estudiantes que aprobaron la materia.
- El 5,9% de los estudiantes tuvieron notas finales inferiores a 3,0 lo cual refiere al porcentaje de estudiantes que reprobaron la materia.
- El 5,9% de los estudiantes no presentaron ninguna de las prácticas ni el requerimiento asignado para el corte.

En general la mayoría de los estudiantes del curso obtuvieron notas finales superiores o iguales a 3,0, lo cual significa que un gran porcentaje de los estudiantes aprobaron la materia para el primer periodo del 2012.

**Segundo periodo del 2012:**

**Notas parciales y totales de los estudiantes**

El curso de LSD se compone de tres cortes, para cada uno de los cuales se realizan prácticas referentes a los temas de RPC<sup>27</sup>, RMI<sup>28</sup> y CORBA<sup>29</sup> respectivamente. El primer y segundo corte sumados equivalen al 70% de la nota total y el tercer corte al 30% restante.

El criterio de evaluación que se tiene en cuenta para cada una de las prácticas es el cumplimiento de las funcionalidades solicitadas. Mientras que los criterios que se tienen en cuenta para la calificación del requerimiento son: manuales, aplicación y el factor de sustentación (Dicho factor en este semestre se promedia con la nota obtenida entre los manuales y la entrega de la aplicación. Anteriormente el factor de sustentación era una nota que variaba entre 0 – 1, la cual era ponderada con la nota obtenida entre los manuales y la entrega de la aplicación).

Los estudiantes del segundo periodo del 2012 que cursaron la materia obtuvieron notas para cada una de las prácticas realizadas, a partir de las cuales se obtuvieron las notas finales; para ver la discriminación de las notas de cada uno de los cortes dirigirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 30.

---

<sup>27</sup> RPC: Remote Procedure Call,

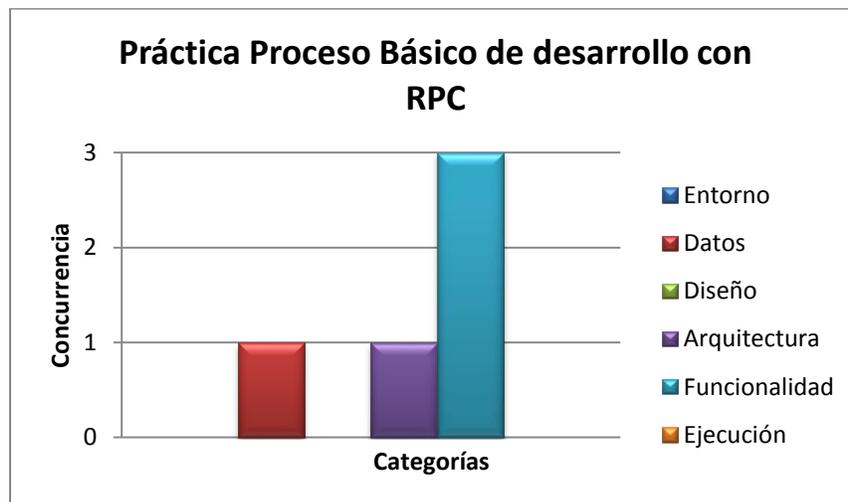
<sup>28</sup> RMI: Java Remote Method Invocation

<sup>29</sup> CORBA: Common Object Request Broker Architecture

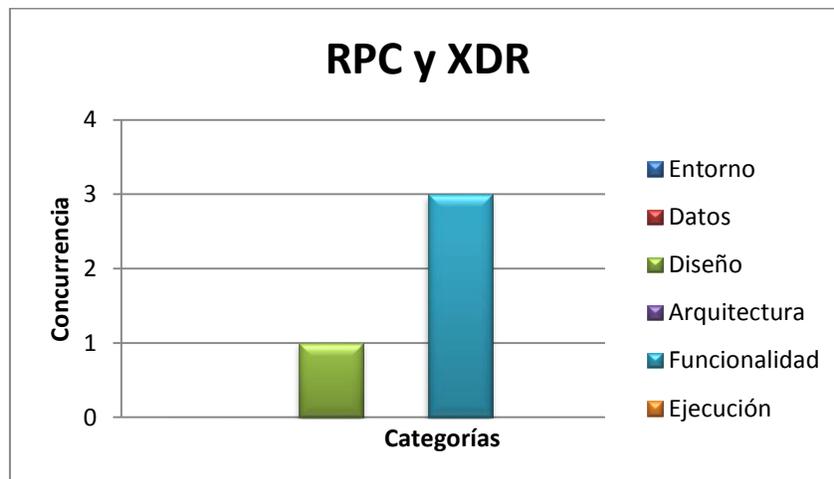
#### 4.12.2 Reporte de Asesorías

Otro de los criterios usados para determinar que las mejoras surgieron efecto en los estudiantes fue la cantidad de errores y asesorías que los estudiantes tuvieron en el desarrollo del curso, lo cual pudo demostrar que las mejoras implementadas con sus herramientas mejoraron la calidad de conocimiento adquirido por los estudiantes, reflejado en menos cantidad de asesorías y de errores presentados en sus implementaciones.

A continuación se muestran las asesorías solicitadas por los estudiantes en las diferentes prácticas sobre el tema RPC (dos prácticas) y RMI (dos prácticas) asignadas en el laboratorio en el II periodo del 2012.



**Figura 17.** Concurrencia vs Categoría de asesorías Práctica Proceso Básico de desarrollo con RPC - LSD II periodo 2012



**Figura 18.** Concurrencia vs Categoría de asesorías Práctica Paso de Parámetros con RMI - LSD II periodo 2012

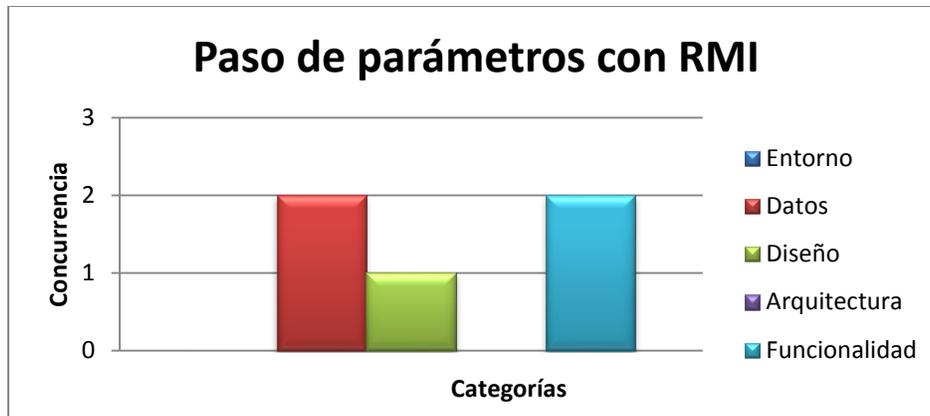


Figura 19. Concurrencia vs Categoría de asesorías Práctica Paso de parámetros con RMI - LSD II periodo 2012

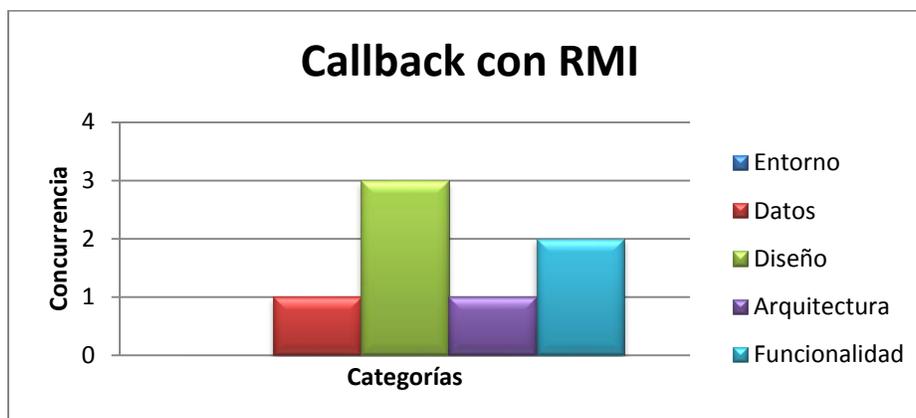


Figura 20. Concurrencia vs Categoría de asesorías Práctica Callback con RMI - LSD II periodo 2012

La información tabulada correspondiente a las gráficas anteriores puede encontrarse en el Anexo F Digital Resultados Implementación de Mejoras, Sección 31.

#### 4.12.3 Reporte de Errores

Se reportan en esta sección los porcentajes de errores presentados por cada grupo en siete prácticas que se realizaron en el laboratorio en el II periodo del 2012.

- Los errores de tipo “Revisión Diseñador” son los que se presentan en el código e implementación.
- Los errores de tipo “Revisión Cliente” definen los errores presentados en la ejecución de la práctica, la perspectiva del cliente.

#### Práctica: Proceso Básico de desarrollo con RPC

Teniendo en cuenta el reporte de errores de los estudiantes en esta práctica (Ver reporte en el Anexo F Digital Resultados Implementación de Mejoras, Sección 32) se obtuvieron los siguientes porcentajes:

- El 20% de los errores en esta práctica son de tipo “Revisión Diseñador”.

- El 80% de los errores corresponden al tipo “Revisión Cliente”.

Los porcentajes indican que se presentaron más errores de ejecución que de código e implementación en esta práctica.

Para ver los porcentajes y el reporte de errores obtenidos en las siguientes prácticas referirse al Anexo F Digital Resultados Implementación de Mejoras, Sección 32.

#### 4.13 INFORME DE IMPLEMENTACIÓN

Se construyó un informe de implementación que permitió determinar si las mejoras realizadas sobre el curso de desarrollo intensivo de software a través de todo el proyecto fueron de ayuda en cada una de las áreas sobre las cuales se aplicó cada una de las actividades y herramientas en pro de la mejora continua; además, en este informe se recomendó evaluar en detalle las oportunidades de mejora identificadas en el diagnóstico y analizar los procesos actuales.

El diagnóstico realizado permitió evaluar el proceso de desarrollo de software, lo que significó poder medir el estado actual de los procesos de desarrollos del LSD, de manera que se conocieron sus fortalezas, riesgos y debilidades. A partir de ello se generaron unas ideas de mejora las cuales fueron aplicadas con diferentes herramientas, instrumentos entre otros, de las cuales se analizaron los resultados de su aplicación en dicho informe.

El propósito del Informe de Implementación es permitir organizar, controlar y administrar la documentación generada, además de guiar las oportunidades de mejora de los procesos y la definición del estimado del tiempo y del esfuerzo requerido.

Dentro del informe de implementación, se especifica:

##### 4.13.1 Esfuerzo de implementación por fases

El esfuerzo se describe en semanas por persona. El esfuerzo planeado corresponde a las semanas por persona que se esperaban utilizar en la ejecución de cada fase; mientras que el esfuerzo efectivo se refiere a las semanas por persona que se utilizaron realmente en la ejecución de estas fases.

Fase	Esfuerzo planeado (Semanas / Persona)	Esfuerzo efectivo (Semanas / Persona)
Instalación	0	0
Diagnóstico	7	3
Formulación	5	5
Mejora	10	8
Revisión	4	2

**Tabla 13.** Esfuerzo de implementación por fases

#### 4.13.2 Información general de iteraciones

Fase	Iteraciones planeadas	Iteraciones Efectuadas
Instalación	0	0
Diagnóstico	0	0
Formulación	1	1
Mejora	2	2
Revisión	0	0

**Tabla 14.** Información general de iteraciones

Iteración	Fase	Objetivos	Esfuerzo Estimado (Semanas / Persona)	Esfuerzo Efectuado (Semanas / Persona)
Iteración 1	Formulación	<ul style="list-style-type: none"> <li>• Evaluar y priorizar oportunidades de mejora dentro del proceso de Preparación por parte del docente de las prácticas que en dicho curso se ejecutan.</li> <li>• Diseñar e instrumentar el proceso de desarrollo software.</li> <li>• Adoptar el nuevo proceso</li> </ul>	5	5
Iteración 2	Mejora	<ul style="list-style-type: none"> <li>• Evaluar y priorizar oportunidades de mejora dentro del proceso Ejecución de las prácticas, donde los estudiantes llevan a cabo el desarrollo de las mismas.</li> <li>• Diseñar e instrumentar el proceso de desarrollo software.</li> <li>• Adoptar el nuevo proceso</li> </ul>	10	10
Iteración 3	Mejora	<ul style="list-style-type: none"> <li>• Evaluar y priorizar oportunidades de mejora dentro del proceso de Evaluación de las prácticas entregadas por los estudiantes.</li> <li>• Diseñar e instrumentar el proceso de desarrollo software.</li> <li>• Adoptar el nuevo proceso</li> </ul>	10	10

**Tabla 15.** Información de objetivos por fase

##### 4.13.2.1 Iteración 1

###### Objetivos de iteración

Objetivo 1. Apoyar la preparación y la planeación de las prácticas por parte del docente en el curso de desarrollo intensivo de software.

###### Cumplimiento del objetivo

El objetivo 1, se cumplió en un 99%, se realizaron las actividades de mejora necesarias para apoyar la labor del docente en el curso de LSD.

#### **4.13.2.2 Iteración 2**

##### **Objetivos de iteración**

Objetivo 1. Apoyar el desarrollo y ejecución de las prácticas por parte de los estudiantes en el curso de desarrollo intensivo de software.

##### **Cumplimiento del objetivo**

El objetivo 1, se cumplió en un 99%, apoyando a los estudiantes en el desarrollo de sus prácticas dentro del LSD.

#### **4.13.2.3 Iteración 3**

##### **Objetivos de iteración**

Objetivo 1. Apoyar el proceso de evaluación de las prácticas entregadas por los estudiantes.

##### **Cumplimiento del objetivo**

El objetivo 1, se cumplió en un 90%, se realizaron las actividades de mejora necesarias para apoyar la labor del docente en el curso de LSD. Sin embargo, se sugiere un mayor soporte para la evaluación de las prácticas, con el fin de ser más efectivo en la entrega de notas y la retroalimentación a los estudiantes del trabajo que han realizado.

(Para ver el Informe de implementación completo referirse al Anexo G Digital Informe de Implementación).

## Análisis de Resultados

---

Finalmente en el proyecto de mejora se realiza una retrospectiva del trabajo realizado, evaluando si se han cumplido los objetivos de la mejora propuesta para el curso de LSD. El análisis de resultados, que aquí se consolida, hace parte de la fase de Revisión propuesta por Agile SPI – Process.

Este análisis permitirá obtener conclusiones sobre si se podrían haber realizado mejor las actividades o tomar mejores soluciones y decisiones, que se analizarán en el próximo capítulo.

### 4.14 MODELO DE PROCESOS REFINADO

La aplicación de Agile SPI-Process principalmente sobre las fases de Mejora y Revisión en el curso de desarrollo intensivo de software LSD, originó modificaciones en el modelo de procesos definido en el trabajo propuesto por López y Magé [6].

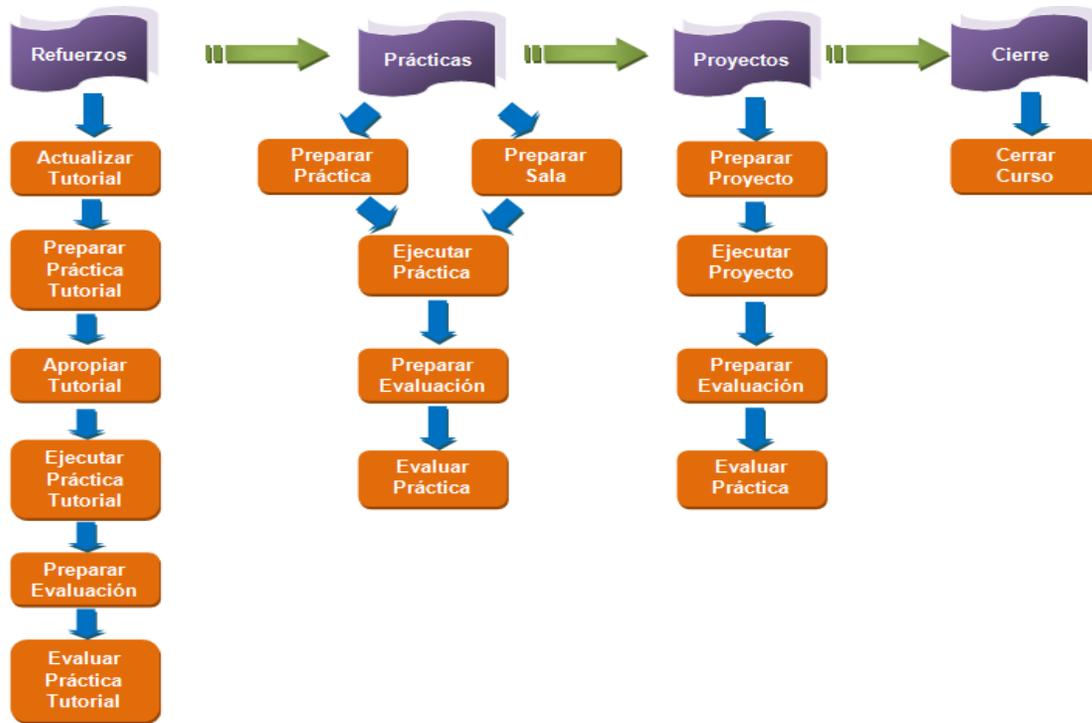
En esta sección se describen las modificaciones realizadas sobre el modelo, teniendo en cuenta las herramientas de mejora adaptadas a los procesos del LSD y los resultados de la ejecución del plan de mejora analizados en el siguiente capítulo. A continuación se describen las fases y actividades adicionadas o modificadas (para ver el modelo de procesos completo y sus especificaciones referirse al Anexo H Modelo de Procesos Refinado, por otro lado para ver el despliegue del modelo con EPF Composer<sup>30</sup> ir al enlace <http://artemisa.unicauca.edu.co/~pimage/modelodeprocesos/>).

#### 4.14.1 Fase Refuerzos

La fase de Refuerzos y sus actividades se adicionaron al modelo. Esta fase corresponde a la preparación previa que se les realizó a los estudiantes en temas que no son específicos del LSD pero que constituyen las bases para desarrollar las prácticas apropiadamente tales como: manejo del entorno Linux y manejo de punteros en C; dichos temas se denominaron *refuerzos* dentro del curso y corresponden a los tutoriales adoptados como herramientas de mejora.

---

<sup>30</sup> EPF Composer es una herramienta que permite generar marcos de procesos de software para una organización.



**Figura 21.** Modelo de Procesos LSD General

En la figura 7 se muestra el modelo de procesos del LSD general obtenido a partir de las modificaciones propuestas.

#### 4.14.1.1 Actividades de la Fase Refuerzos

A continuación se muestran las tareas propuestas para cada actividad de la Fase Refuerzos.

- Actividad Actualizar tutorial: En la fase de refuerzo, como primera medida se debe actualizar el tutorial específico para reforzar los conocimientos deseados en los estudiantes.



**Figura 22.** Diagrama de tareas de la actividad Actualizar tutorial

Tarea: Recolectar información

<b>Descripción</b>	Se recolecta información sobre el tema específico del tutorial en diversas fuentes como la web, libros y otros tutoriales relacionados.
<b>Entradas</b>	Tutorial*
<b>Salidas</b>	Tutorial actualizado
<b>Roles</b>	Docente

**Tabla 16.** Descripción formal de la tarea Recolectar información

\*Se establece como entrada el tutorial, debido a que los tutoriales ya fueron creados como parte de las herramientas de mejora realizadas, por tanto, para los siguientes semestres del curso de LSD sólo es necesario actualizar su información.

Tarea: Adaptar concepto LSD

<b>Descripción</b>	Se realiza los cambios necesarios a la información recolectada de manera que se adapte a las necesidades del laboratorio.
<b>Entradas</b>	Tutorial actualizado
<b>Salidas</b>	Tutorial adaptado
<b>Roles</b>	Docente

**Tabla 17.** Descripción formal de la tarea Adaptar concepto LSD

Tarea: Presentar tutorial

<b>Descripción</b>	Se realiza la exposición general del tutorial que se va a trabajar en la clase.
<b>Entradas</b>	Tutorial adaptado
<b>Salidas</b>	Ninguna
<b>Roles</b>	Docente

**Tabla 18.** Descripción formal de la tarea Presentar tutorial

Para cada una de las tareas de las siguientes actividades se realizó la descripción formal como se mostró en las tareas de la actividad Actualizar tutorial; para ver dichas descripciones referirse al Anexo H Modelo de Procesos Refinado.

- Actividad Preparar práctica tutorial: Para reforzar los conocimientos que aporta el tutorial, el docente prepara una práctica acorde al tema.



**Figura 23.** Diagrama de tareas de la actividad Preparar práctica tutorial

- Actividad Apropiar tutorial: Los estudiantes apropian la información proporcionada por el docente en el tutorial.



**Figura 24.** Diagrama de tareas de la actividad Apropiar tutorial

Para ver la descripción formal de cada una de las tareas referirse al Anexo H Modelo de Procesos Refinado.

- Actividad Ejecutar práctica tutorial: Los estudiantes ejecutan la práctica definida para reforzar los conocimientos del tutorial.



Figura 25. Diagrama de tareas de la actividad Ejecutar práctica tutorial

- Actividad Preparar evaluación



Figura 26. Diagrama de tareas de la actividad Preparar evaluación

- Actividad Evaluar práctica tutorial



Figura 27. Diagrama de tareas de la actividad Evaluar práctica tutorial

#### 4.14.2 Modificaciones en la Fase Prácticas

Las modificaciones realizadas en la Fase Prácticas se enfocan en la Actividad *Ejecutar Prácticas*, específicamente en las tareas de Diseñar y Desarrollar.



Figura 28. Diagrama de tareas de la actividad Ejecutar práctica

##### 4.14.2.1 Tarea Diseñar

El modelo propuesto por López y Magé [6] define en los pasos para la tarea Diseñar de la actividad Ejecutar prácticas, diferentes diagramas: diagramas de casos de uso, diagramas de clase, diagramas de colaboración y diagramas de secuencia.

Sin embargo, la experiencia indica que los estudiantes no realizan todos estos diagramas debido a las restricciones de tiempo para ejecutar la práctica, esto porque dan prioridad a la codificación. En algunos casos sólo se realizaron diagramas de secuencia, ya que las actividades eran de menor complejidad. A continuación se muestra un diagrama de los pasos modificados para esta tarea.



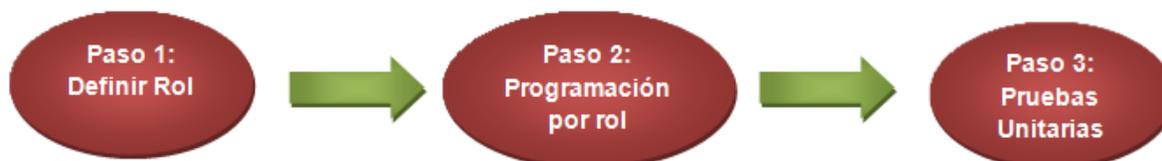
**Figura 29.** Diagrama de pasos para la tarea Diseñar - actividad Ejecutar práctica

Cabe resaltar que todos los diagramas considerados por López y Magé [6] si son desarrollados por los estudiantes pero en los proyectos (requerimientos) donde se concede mayor tiempo para el diseño.

Para ver la descripción formal de los pasos de esta tarea referirse al Anexo H Modelo de Procesos Refinado.

#### 4.14.2.2 Tarea Desarrollar

Dentro de las actividades de Mejora se aplicó la definición de roles y responsabilidades en el desarrollo, por tanto se hace necesario incluir este paso dentro de la tarea Desarrollar. Consecuentemente, también se modificó el tipo de programación: programación por rol y no programación por pares como lo definió López y Magé [6].



**Figura 30.** Diagrama de pasos para la tarea Desarrollar - actividad Ejecutar práctica

#### Paso 1: Definir Rol

<b>Descripción</b>	Al inicio de cada una de las prácticas en los grupos, cada uno de los estudiantes define su rol, dividiendo las responsabilidades y comprometiéndose a realizar un módulo de la práctica. El docente considera este rol para la calificación, por lo que los estudiantes son evaluados individualmente.
<b>Entradas</b>	Ninguna
<b>Salidas</b>	Definición de roles y responsabilidades
<b>Roles</b>	Docente (supervisor), Estudiante (ejecutor)

**Tabla 19.** Descripción formal del paso Definir rol – tarea Desarrollar

## Paso 2: Programación por rol

<b>Descripción</b>	Los estudiantes desarrollan las líneas de código necesarias para implementar cada una de las funcionalidades que fueron analizadas y diseñadas en las actividades anteriores; esta tarea será realizada por rol, es decir cada estudiante asume la responsabilidad del módulo asignado dependiendo del rol definido.
<b>Entradas</b>	Entregable de Diseño, Definición de roles y responsabilidades
<b>Salidas</b>	Fuentes de la práctica
<b>Roles</b>	Docente (supervisor), Estudiante (ejecutor)

**Tabla 20.** Descripción formal del paso Programación por rol – tarea Desarrollar

Para ver la descripción formal de los otros pasos de esta tarea referirse al Anexo H Modelo de Procesos Refinado.

#### 4.14.3 Modificaciones en la Fase Proyectos

La fase Proyectos es muy similar a la Fase Prácticas. Las modificaciones realizadas en la Fase Proyectos se enfocan en la Actividad *Ejecutar Proyecto*, específicamente en la tarea Desarrollar.



**Figura 31.** Diagrama de tareas de la actividad Ejecutar proyecto

##### 4.14.3.1 Tarea Desarrollar

Como se ha especificó anteriormente, dentro de las actividades de Mejora se aplicó la definición de roles y responsabilidades en el desarrollo, lo cual aplica también en el desarrollo que se realiza dentro de los requerimientos o proyectos. Por tanto se hace necesario incluir este paso dentro de la tarea Desarrollar. Consecuentemente, también se modificó el tipo de programación: programación por rol y no programación por pares como lo definió López y Magé [6].



**Figura 32.** Diagrama de pasos para la tarea Desarrollar - actividad Ejecutar proyecto

## Paso 1: Definir Rol

<b>Descripción</b>	Cada uno de los estudiantes del grupo define su rol, dividiendo las responsabilidades y comprometiéndose a realizar uno o más módulos del requerimiento.
<b>Entradas</b>	Ninguna
<b>Salidas</b>	Definición de roles y responsabilidades

Tabla 21. Descripción formal del paso Definir rol – tarea Desarrollar

## Paso 2: Programación por rol

<b>Descripción</b>	Los estudiantes desarrollan las líneas de código necesarias para implementar cada una de las funcionalidades que fueron analizadas y diseñadas en las actividades anteriores; esta tarea será realizada por rol, es decir cada estudiante asume la responsabilidad del módulo asignado dependiendo del rol definido.
<b>Entradas</b>	Entregable de Diseño, Definición de roles y responsabilidades
<b>Salidas</b>	Fuentes del requerimiento
<b>Roles</b>	Docente (supervisor), Estudiante (ejecutor)

Tabla 22. Descripción formal del paso Programación por rol – tarea Desarrollar

Para ver la descripción formal de los otros pasos de esta tarea referirse al Anexo H Modelo de Procesos Refinado.

## 4.15 EVALUACIÓN DE LA MEJORA

### 4.15.1 Evaluación de la Mejora desde la perspectiva de los Indicadores de Mejora definidos

En la fase de Diagnóstico se establecieron las métricas y los indicadores de mejora respectivamente. Estos indicadores son analizados respecto a las decisiones y acciones aplicadas al proceso del curso de LSD y los resultados obtenidos después de la aplicación de la Mejora; dichos indicadores señalan si se ha cumplido o no los objetivos del proyecto de mejora.

A continuación se evalúa la mejora con los indicadores establecidos, comparando los resultados del diagnóstico y los resultados actuales.

Métrica	Indicador de Mejora	Resultados del Diagnóstico	Mejora con Agile SPI-Process	Análisis
<b>Tiempo promedio de desarrollo</b>	Tiempos promedio de desarrollo después de la mejora	Los tiempos registrados en el diagnóstico correspondiente al II periodo del 2011 del LSD, mostraron que los estudiantes invierten, en promedio, en la realización de la interfaz 21 minutos, en la realización del cliente 47 minutos y en el servidor 42 minutos. Lo anterior indica que en la mayoría de las prácticas, los estudiantes gastan menos tiempo en la realización de la interfaz, seguida del tiempo empleado en el servidor y por último en el cliente (ver registro de tiempos en Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 2).	<p>En el I periodo del 2012, los tiempos registrados muestran que los estudiantes invierten, en promedio, en la realización de la interfaz 14 minutos, en la realización del cliente 43 minutos y en el servidor 52 minutos. A diferencia de los otros tiempos, el módulo en el que mayor tiempo emplean es el servidor.</p> <p>En el II periodo del 2012, los estudiantes, en promedio invierten en la realización de la interfaz 19 minutos, en la realización del cliente 47 minutos y en el servidor 38 minutos.</p> <p>(Ver registro de tiempos en Anexo F Digital Resultados Implementación de Mejoras, Sección 23 y Sección 25 respectivamente).</p>	Después del proyecto de Mejora, los estudiantes tienen mayor productividad en la realización del módulo interfaz y el cliente.
<b>Porcentaje de grupos con registro de tiempos incompletos y tiempos completos</b>	Porcentaje de grupos que finalizaron la práctica	De acuerdo al registro de tiempos en el diagnóstico, sólo un grupo registra la finalización de todas sus prácticas del LSD, esto equivale al 8,33% del total de los grupos.	En el I periodo del 2012 y II periodo del 2012, dos grupos sobresalen en la finalización de sus prácticas, esto equivale al 25% de los grupos del LSD para cada periodo.	Según los resultados, la mejora aumentó la productividad de los estudiantes en el laboratorio, al igual que su desempeño para terminar las prácticas

<p><b>Promedio de notas</b></p>	<p>Porcentaje de estudiantes que aprobaron la materia</p> <p>El 70,833% de los estudiantes tuvieron notas finales superiores o iguales a 3,0 lo que corresponde a los estudiantes que aprobaron la materia. Por otra parte, el 29,166% de los estudiantes tuvieron notas finales inferiores a 3,0 lo cual se refiere al porcentaje de estudiantes que reprobaron la materia (ver discriminación de las notas en el Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 4).</p>	<p>Para el I periodo del 2012, el 88,2% de los estudiantes aprobaron la materia. Por otro lado, el 5,9% de los estudiantes reprobaron la materia y el 5,9% de los estudiantes no presentaron ninguna de las prácticas ni el requerimiento asignado para el corte.</p> <p>Para el II periodo del 2012, el 87,5% de los estudiantes aprobaron la materia y el 12,5% de los estudiantes reprobaron la materia.</p> <p>(Ver discriminación de las notas en el Anexo F Digital Resultados Implementación de Mejoras, Sección 29 y Sección 30 respectivamente).</p>	<p>asignadas.</p> <p>Existe un mayor porcentaje de estudiantes que han aprobado la materia en la actualidad. Esto indica que la mejora ha disminuido el índice de deserción de estudiantes en el LSD.</p>
<p><b>Porcentaje de reporte de asesorías</b></p>	<p>Porcentaje de reporte de asesorías después de la mejora</p> <p>De acuerdo a los resultados en el diagnóstico, el porcentaje para cada categoría de asesoría es el siguiente:                  Entorno: 7,69%                  Datos: 20,51%                  Diseño: 2,56%                  Arquitectura: 15,38%                  Funcionalidad: 30,77%                  Ejecución: 15,38%</p>	<p>Para el II periodo del 2012, el reporte de asesorías en porcentajes fue el siguiente:                  Entorno: 0%                  Datos: 19,05%                  Diseño: 23,81%                  Arquitectura: 9,52%                  Funcionalidad: 47,62%                  Ejecución: 0%                  Lenguaje: 0%</p>	<p>En comparación con los resultados del diagnóstico, el curso después de la mejora ha solventado dudas respecto a algunas categorías de asesorías (Entorno, Ejecución, Lenguaje, Compilación). Sin embargo, aún</p>

<b>Porcentaje de reporte de errores</b>		<p>Lenguaje: 5,13%                  Compilación: 2,56%</p> <p>(Ver la información tabulada que permitió obtener estos porcentajes en el Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 5).</p>	<p>Compilación: 0%</p> <p>(Ver la información tabulada que permitió obtener estos porcentajes en el Anexo F Digital Resultados Implementación de Mejoras, Sección 31).</p>	<p>prevalece las asesorías sobre Funcionalidad, y a diferencia del estado del LSD antes de la Mejora, el Diseño presenta mayor porcentaje que las asesorías respecto a Datos.</p>
	Porcentaje de reporte de errores después de la mejora	<p>El 53,33% de los errores en esta práctica son de tipo “Revisión Diseñador”.</p> <p>El 46,66% de los errores corresponden al tipo “Revisión Cliente”.</p> <p>(Ver la información tabulada que permitió obtener estos porcentajes en el Anexo C Digital Medidas realizadas en el Diagnóstico, Sección 6).</p>	<p>Para el II periodo del 2012, el 30% de los errores en esta práctica son de tipo “Revisión Diseñador”; el 70% de los errores corresponden al tipo “Revisión Cliente”.</p> <p>(Ver la información tabulada que permitió obtener estos porcentajes en el Anexo F Digital Resultados Implementación de Mejoras, Sección 32).</p>	<p>El curso de LSD presentaba un mayor índice en los errores de tipo “Revisión Diseñador” antes de la mejora, es decir, los estudiantes cometían más errores de código e implementación.</p> <p>Actualmente, la valoración de la mejora muestra que estos errores han disminuido, pero al mismo tiempo se ha elevado el número de errores “Revisión Cliente”, correspondientes a los errores presentes en la ejecución de la práctica.</p>

**Tabla 23.** Mejora desde la perspectiva de los Indicadores de Mejora definidos.

#### 4.15.2 Evaluación de la Mejora desde la perspectiva de las deficiencias encontradas

Las oportunidades de Mejora identificadas a partir de la información recolectada en la fase de Diagnóstico surgieron a partir de las deficiencias encontradas. A continuación se analiza si dichas deficiencias fueron superadas con la mejora realizada, de acuerdo a los resultados actuales y la experiencia en el curso de LSD.

Deficiencia	Herramienta de mejora utilizada	Experiencia	Actualidad
<ul style="list-style-type: none"> <li>• Falta de conocimiento y poca experiencia en comandos Linux.</li> <li>• Retraso en el envío de las prácticas.</li> </ul>	Tutorial comandos Linux	<p>Se proporcionó una clase del LSD para que los estudiantes del curso estudien el tutorial sobre comandos Linux (ver tutorial en Anexo E: Herramientas de Mejora, Sección 4: Tutorial comandos Linux).</p> <p>Al final de la lectura del tutorial, se les realizó un ejercicio donde aplicaban los conocimientos adquiridos.</p>	<p>Actualmente en el curso, los estudiantes están utilizando este mecanismo previo a enfrentarse a los requerimientos de las prácticas en las cuales se utiliza el entorno Linux. De esta forma, ellos recuerdan los principales comandos de Linux y se muestran más seguros en la utilización del entorno para la tecnología RPC.</p> <p>Además las evaluaciones de conocimiento antes y después del tutorial, demuestran que los estudiantes han adquirido los conocimientos deseados (ver resultados de evaluación de conocimiento en Anexo F Digital Resultados Implementación de Mejoras, Sección 4 y Sección 6).</p>
Falta de conocimiento y poca experiencia en la compilación y ejecución de un programa en C para Linux.	Tutorial de compilación y ejecución de un programa en C	Se proporcionó una clase del LSD con el propósito que los estudiantes del curso estudien el tutorial de ejecutables en Linux para la compilación y ejecución de un programa en C (ver tutorial en Anexo E:	Los estudiantes del curso tienen más destreza en el manejo de la compilación y ejecución de un programa en C (ver resultados de evaluación de conocimiento en Anexo F Digital Resultados

		<p>Herramientas de Mejora, Sección 5: Tutorial compilación y ejecución de un programa en C).</p> <p>Terminando la lectura del tutorial, se les realizó un ejercicio donde aplicaban los conocimientos adquiridos, teniendo en cuenta además los conocimientos del tutorial anterior.</p>	Implementación de Mejoras, Sección 9 y Sección 11).
<ul style="list-style-type: none"> <li>• Uso inadecuado de la gestión de memoria dinámica en C.</li> </ul>	Tutorial de conocimientos en lenguaje C	<p>Dentro de una de las clases de LSD se dio el espacio para que los estudiantes leyeran el tutorial (ver tutorial en Anexo E: Herramientas de Mejora, Sección 6: Tutorial punteros en C). Al igual que los otros casos en la aplicación de tutoriales, al finalizar la lectura se les realizó a los estudiantes un ejercicio que evaluara su lectura y aprendizaje.</p>	Los estudiantes registran menores problemas con el correcto uso de datos en este lenguaje, pero aún persisten algunos errores sobre gestión de la memoria dinámica en C (ver resultados de evaluación de conocimiento en Anexo F Digital Resultados Implementación de Mejoras, Sección 14 y Sección 16).
<ul style="list-style-type: none"> <li>• Errores de sintaxis en la implementación de las prácticas referentes al lenguaje de programación Java.</li> </ul>	Tutorial de conocimientos en lenguaje Java	<p>Se brindó una clase del LSD para que los estudiantes leyeran el tutorial de Java (ver tutorial en Anexo E: Herramientas de Mejora, Sección 7: Tutorial Java). Al igual que los otros casos en la aplicación de tutoriales, al finalizar la lectura se les realizó a los estudiantes un ejercicio que evaluara los conocimientos adquiridos.</p>	Los estudiantes registran menores problemas con el correcto uso de datos en Java, pero aún persisten algunos errores en la compilación (ver resultados de evaluación de conocimiento en Anexo F Digital Resultados Implementación de Mejoras, Sección 19 y Sección 21).
Muchos estudiantes no llegan puntualmente al inicio de la clase y pierden la explicación	Video RPC, Video RMI, Video Callback RMI, Video CORBA,	Se facilitó a los estudiantes el video de cada una de las tecnologías usadas en las prácticas de LSD, donde se explicaba la manera de compilar y ejecutar un programa	Los registros de asesorías después de la mejora no indican índices en las asesorías por compilación y ejecución, lo que nos permite asegurar que los estudiantes han

<p>proporcionada por el profesor.</p>	<p>Video Callback CORBA</p>	<p>en cada caso.  Cada video se subió a la plataforma de la asignatura como recursos para los estudiantes. La responsabilidad de utilizar los videos fue de cada estudiante, dado que estos constituían un recurso del curso y no una herramienta impuesta.</p>	<p>solventado la necesidad de buscar al docente para resolver problemas de cómo compilar y ejecutar un programa (ver reporte de asesorías en Anexo F Digital Resultados Implementación de Mejoras, Sección 31).</p>
<ul style="list-style-type: none"> <li>• Trabajo no colaborativo en el grupo.</li> <li>• En la pareja, un estudiante toma toda la responsabilidad de la práctica.</li> <li>• Dificultad para apropiarse los requerimientos.</li> </ul>	<p>Servidor y Clientes de Prueba</p>	<p>Para todas las prácticas, se subió a la plataforma el servidor y el cliente de prueba. Cada estudiante tuvo un rol: un estudiante se encargaba de la codificación del servidor y otro del cliente; por tanto, cada estudiante utilizó el ejecutable de prueba para evaluar su código y corregir sus posibles errores.</p>	<p>La forma de trabajo de los estudiantes en el laboratorio actualmente es dividida por rol: un estudiante se encarga del módulo servidor y otro del módulo cliente. Por tanto, se califica a cada estudiante del grupo por separado dependiendo del desempeño en su propio rol.  Por otro lado, en el reporte de errores se han presentado menos errores por lógica, lo que evidencia que los ejecutables de prueba son una ventaja para que el estudiante comprenda fácilmente el requerimiento de cada práctica (ver reporte de errores en Anexo F Digital Resultados Implementación de Mejoras, Sección 32).</p>
<p>Presencia de los mismos errores en diferentes prácticas.</p>	<p>Documento de Retroalimentación</p>	<p>El docente señaló a los estudiantes las principales observaciones de las inconsistencias, errores y dificultades en las prácticas.</p>	<p>El docente realiza un documento de los comentarios sobre el desempeño de los estudiantes en una práctica cuando lo considera pertinente. Esto con el propósito de que los estudiantes corrijan errores frecuentes o debilidades en el desarrollo</p>

			de los módulos.
Dificultad para apropiarse los requerimientos de cada práctica	Modelo de secuencia	<p>Se solicitó a los estudiantes realizar un modelo de secuencia en algunas de las prácticas de menor complejidad, teniendo en cuenta el tiempo que implica realizar esta parte del diseño antes de llegar a la implementación.</p> <p>Los estudiantes realizaron el modelo de secuencia pertinente y constituyó un entregable de la práctica.</p>	El docente solicita a los estudiantes un modelo de secuencia en las prácticas que él considere necesarias, teniendo en cuenta la disponibilidad de tiempo y la complejidad de las mismas.
Retraso en la implementación de las prácticas debido a la falta de configuración de las plataformas y entornos necesarios.	Guía y demo de configuración de equipos.	<p>Antes del inicio de las prácticas de la tecnología RPC en el LSD, se le suministró al monitor de la sala una guía de configuración de las plataformas y los entornos necesarios para la elaboración de las prácticas en esta tecnología (ver Anexo E: Herramientas de Mejora, Sección 3: Guía de configuración, demo de configuración).</p> <p>Se acordó una jornada con el monitor donde no la sala del LSD estuviera disponible; en esta jornada se supervisó al monitor para que configurara en su totalidad todos los equipos y para la resolución de dudas de la guía.</p>	<p>Actualmente el monitor configura los equipos con el entorno y las plataformas necesarias en una forma estricta y previa al desarrollo de las prácticas.</p> <p>No se reportan asesorías de tipo Entorno, donde el estudiante tenga problemas o inquietudes al respecto (ver reporte de asesorías en Anexo F Digital Resultados Implementación de Mejoras, Sección 31).</p>
Repetición de partes de las guías y demora en su creación.	Generación de guías en PDF sobre una aplicación web.	En la aplicación web de apoyo y soporte a la mejora se realizó una sesión para el docente. En esta sesión, el docente pudo generar los PDFs necesarios dependiendo	En la fase de Diagnóstico, la medición del tiempo que tardaba el docente en la realización de las prácticas en PDF fue en promedio de 90 minutos. Actualmente, a

		<p>de la tecnología seleccionada (ver Anexo E: Herramientas de Mejora, Sección 2: Aplicación Web).</p> <p>En un principio fue necesaria la capacitación del docente para la utilización de la herramienta y se presentaron algunas dudas e inconsistencias que aportaron su mejoramiento.</p>	<p>favor de su productividad, el docente sólo tarda 20 minutos en la elaboración de las prácticas donde se incluye el tiempo que tarda en la elaboración de las gráficas, redacción de texto necesario y la generación del PDF por la aplicación web.</p>
<p>Falta de experiencia y acercamiento con cada una de las tecnologías realizadas en las prácticas.</p>	<p>Repositorio de las mejores prácticas</p>	<p>En la página o plataforma disponible para el curso se subió el repositorio de las mejores prácticas de los semestres anteriores dependiendo de la tecnología.</p> <p>La responsabilidad de utilizar el repositorio fue de cada estudiante, dado que estos constituían un recurso del curso y no una herramienta impuesta.</p>	<p>El docente selecciona las mejores prácticas de semestres anteriores y las carga en la plataforma disponible para el curso. Los estudiantes cuentan con este repositorio disponible según la tecnología requerida.</p>
<p>Retraso en la entrega oportuna de las notas por parte del docente.</p>	<p>Script (ejecutable) para evaluación</p>	<p>Se suministró al docente diferentes scripts para la evaluación de los módulos de las diferentes prácticas desarrolladas por los estudiantes (ver Anexo E: Herramientas de Mejora, Sección 8: Scripts para evaluar prácticas).</p> <p>El docente apropió dichos scripts como método de evaluación.</p>	<p>El docente tardaba en promedio 150 minutos en la evaluación de las prácticas de laboratorio según la información recolectada en la fase de Diagnóstico. Actualmente, el docente sólo tarda 90 minutos con el método de evaluación mediante Scripts. La diferencia entre el tiempo empleado antes y después de la mejora, indica una mejora significativa en la producción de la labor del docente.</p>

**Tabla 24.** Mejora desde la perspectiva de las deficiencias encontradas.

## Resultados Estudio de Caso

A partir de los resultados obtenidos en la aplicación de Agile SPI-Process académico en un curso (estudio de caso), se obtienen las pautas para responder a la pregunta ¿Agile SPI – Process adaptado para el ámbito académico es útil y aplicable para este contexto, permitiendo así la mejora en sus procesos?. Estas pautas se describen a continuación.

### 4.16 RESULTADOS

#### 4.16.1 Nivel de Utilidad

- El análisis de resultados que se muestra en la Tabla 23 especifica los datos obtenidos antes y después de cada una de las mejoras aplicadas en el curso; el impacto de cada una de estas mejoras sobre el desempeño de los estudiantes se refleja en la nota final que estos obtienen.

En resumen, el porcentaje de estudiantes que aprobaron el curso antes de la mejora fue del 70,8%, mientras que después de la aplicación de la mejora este porcentaje fue de 88,2% para el I periodo de 2012 y de 87,5% para el II periodo de 2012.

- A continuación se muestra el resultado de la encuesta realizada al docente que permite determinar su percepción sobre la utilidad del proceso de mejora en el curso:

	Nivel de Impacto Positivo		
	Bajo (1 a 2)	Medio (3)	Alto (4 a 5)
<b>Número de Preguntas</b>	0	1	4

**Tabla 25.** Resumen de resultados encuesta docente percepción de utilidad

La tabla 25 muestra que el 80% de las preguntas contestadas por el docente está en un nivel alto de impacto positivo sobre el curso (ver encuesta y resultados en el Anexo F Digital Resultados Implementación de Mejoras, Sección 33).

- A continuación se muestra el resultado de la encuesta realizada a los estudiantes que permite determinar si las herramientas de mejora son un apoyo positivo para el desarrollo del curso:

	Nivel de Utilidad		
	Baja (1 a 2)	Media (3)	Alta (4 a 5)
<b>Promedio de porcentajes</b>	3%	11%	86%

**Tabla 26.** Resumen de resultados de la percepción de los estudiantes sobre el nivel de utilidad de las herramientas de mejora.

La tabla 26 muestra que en promedio el 86% de los estudiantes clasifica las herramientas de mejora en un nivel alto de utilidad (ver pregunta 14, Anexo F Digital Resultados Implementación de Mejoras, Sección 28).

- Los resultados de la aplicación de la mejora en cada una de las iteraciones. (Ver este Capítulo sección 4.15 Evaluación de la Mejora)

#### 4.16.2 Nivel de Aplicabilidad

- A continuación se muestra el resultado de la encuesta realizada al docente que permite determinar su percepción sobre la aplicabilidad del procedimiento formal en el curso:

	Nivel de facilidad de aplicación		
	Bajo (1 a 2)	Medio (3)	Alto (4 a 5)
<b>Número de Preguntas</b>	0	2	3

**Tabla 27.** Resumen de resultados encuesta docente percepción de aplicabilidad

La tabla 27 muestra que el 60% de las preguntas contestadas por el docente está en un nivel alto de facilidad de aplicación del procedimiento formal sobre el curso (ver encuesta y resultados en el Anexo F Digital Resultados Implementación de Mejoras, Sección 34).

- El esfuerzo que implicó la aplicación de Agile SPI-Process académico sobre un curso para los ingenieros de procesos abarcó diferentes etapas:
  - Capacitación sobre mejora de procesos en empresas.
  - Capacitación sobre Agile SPI-Process.
  - Diseño del procedimiento formal preliminar de Agile SPI-Process académico.
  - Ejecución de estudio de caso (aplicación de Agile SPI-Process académico en un curso).

<b>Esfuerzo</b> (semanas trabajadas por persona)			
Capacitación sobre mejora de procesos	Capacitación sobre Agile SPI-Process	Diseño del procedimiento formal preliminar	Aplicación de Agile SPI-Process académico en un curso
<b>5</b>	10	28	18

**Tabla 28.** Esfuerzo general del proyecto

La tabla 28 muestra que el esfuerzo necesario específicamente para la aplicación de Agile SPI-Process académico fue en promedio de 18 semanas por persona, según consideraciones de los ingenieros de procesos.

#### 4.17 ANÁLISIS DE RESULTADOS

Los resultados demuestran que después de la aplicación de las mejoras el porcentaje de estudiantes que aprobó el curso aumentó, consolidando dichas mejoras como mecanismos útiles para incrementar el buen desempeño de los estudiantes en el curso. Además, los estudiantes percibieron que las herramientas que se les proporcionó a raíz de la mejora constituyeron un apoyo para el desarrollo de sus actividades. Por otro lado, desde la perspectiva del docente es posible clasificar este proceso de mejora como útil teniendo en cuenta el impacto positivo que se generó sobre el curso a partir de sus actividades realizadas.

En resumen, en cuanto a la utilidad de Agile SPI-Process académico se puede determinar que a partir de los resultados mencionados anteriormente, se cumple con todas las pautas definidas para concluir que Agile SPI-Process académico es realmente útil para realizar un proceso de mejora en este contexto.

La aplicabilidad de Agile SPI-Process académico está dada a partir de la percepción del docente respecto a este indicador, y los resultados especifican que no se clasifica el proceso como fácilmente aplicable pero sí provee mecanismos para su aplicación en este contexto. Así mismo, desde la perspectiva de los ingenieros de procesos quienes definieron el proceso adaptado y su respectiva aplicación, se puede observar que para cumplir con todas las etapas se necesita de un gran esfuerzo en términos de tiempo por persona, pero que el esfuerzo requerido para llevar a cabo específicamente la ejecución de Agile SPI-Process académico cumple con el tiempo requerido para un periodo académico.

Los resultados obtenidos de las métricas para determinar la aplicabilidad de Agile SPI-Process académico establecen entonces que su aplicación es medianamente fácil para estos contextos.

#### **4.18 COMPARACIÓN DE AGILE SPI-PROCESS ACADÉMICO VS AGILE SPI-PROCESS**

El nivel de adopción de Agile SPI – Process en un entorno académico se ha identificado mediante la experiencia en el LSD. Dicha experiencia indica que adaptar las fases de Agile SPI – Process en el proyecto de Mejora en este contexto es posible. Sin embargo, los pasos sobre el cómo hacerlo no están claramente definidos, es decir, no es posible identificar cuál es el camino apropiado para la realización de un diagnóstico, cómo plantear el plan de mejora resultante de las observaciones de esta fase, cómo formular, realizar la mejora y evaluarla. Por lo tanto, el experto debe buscar este “cómo” en base a otras metodologías o en su experiencia adecuando las actividades o pasos a los propósitos y la definición que establece en cada fase Agile SPI-Process.

A continuación se muestra una tabla comparativa entre los pasos establecidos por Agile SPI-Process, lo realizado en la experiencia con el curso académico de desarrollo software y lo propuesto en el procedimiento formal definido en el capítulo 3 (recomendaciones).

La Prioridad indica el nivel de importancia y de aplicación en el proceso de mejora del curso académico de desarrollo software (utilizando una escala de 0 a 3, 1 en el caso de haber sido muy influyente en el proyecto de mejora del curso, 2 para un paso que influyó en la mejora del curso, 3 para una influencia débil, y 0 para ninguna influencia o un paso que no se aplicó en la experiencia).

Prioridad	Agile SPI-Process	Experiencia	Agile SPI-Process adaptado al contexto académico
<b>Fase de Instalación</b>			
2	En la fase de instalación del Programa se identifican las necesidades del negocio que conducen la mejora. También se definen los roles en el proceso de mejora, se planea y se definen los elementos que deberán ir dentro de la propuesta de mejora.	<p>Dentro de la experiencia, se tomaron los resultados del trabajo de López y Magé [6], el cual incluye el modelo de procesos para un contexto académico. También se adaptaron los principales roles a este contexto (estudiantes, docente, monitor), dado que es diferente a los roles especificados en una empresa.</p> <p>Los elementos que deberán incluir dentro de la propuesta de mejora se fueron perfeccionando y agregando a medida que se realizaban las iteraciones.</p>	Agile SPI-Process y Agile SPI-Process adaptado al contexto académico concuerdan en este sentido. Aunque la adaptación es mucho más concreta sobre este proceso.
1	En la fase de instalación del Programa se construye una propuesta de mejora a partir de estas necesidades del negocio que conducen la mejora.	Dentro de la experiencia, se tomaron los resultados del trabajo de López y Magé [6], que incluían la propuesta de mejora. Sin embargo dentro de lo propuesto se agregó la definición de un estándar que apoyara las labores de mejora; en este caso se utilizó la ISO 29110.	En la construcción de la propuesta de mejora, Agile SPI-Process adaptado al contexto académico se propone definir un estándar de calidad que brinde un conjunto de mejores prácticas para VSE, específicamente la ISO 29110.
3	Las tres últimas actividades en la fase de instalación constituyen obtener la aprobación de la propuesta de	En la experiencia se evaluó con el docente la propuesta de mejora y respecto a sus observaciones se adaptó, iniciando con el programa de	En este aspecto, Agile SPI-Process y Agile SPI-Process adaptado al contexto académico se comportan de forma similar, pero adaptando los

mejora, adecuar dicha mejora. propuesta respecto a las observaciones y finalmente lanzar el programa.

roles, ya que quien aprueba la propuesta de mejora en este caso es el docente (jugando el papel de la gerencia en una organización).

### Fase de Diagnóstico

1	Agile SPI-Process no define una actividad previa a <i>valorar</i> , referente a la planificación del diagnóstico.	En la experiencia se planificó el Diagnóstico, definiendo los objetivos, las métricas e indicadores de mejora y las fuentes de información para el desarrollo del estudio de caso en el LSD.	Agile SPI-Process en el contexto académico define una actividad <i>planificar</i> que planee el diagnóstico, que oriente esta fase y el sentido de la selección de información dentro del proyecto de mejora.
1	Agile SPI-Process describe que en la fase de diagnóstico se inicia valorando el estado actual de los procesos de la empresa.	En la experiencia se inició la valoración tomando tiempos de producción de las prácticas de los estudiantes, observando el modo de trabajo, realizando encuestas para identificar dificultades en el laboratorio, registrando los errores y las asesorías que presentaban los estudiantes dentro del laboratorio, registrando las notas parciales y finales de los estudiantes. Todo lo anterior con la intención de estimar el estado actual del curso de desarrollo intensivo de software.	Agile SPI-Process y Agile SPI-Process en el contexto académico concuerdan con esta actividad en donde se valora el estado actual de los procesos.
1	Agile SPI-Process determina que se debe priorizar los casos de mejora en la empresa con base a la valoración realizada.	En la experiencia se definieron los criterios de priorización de las oportunidades de mejora, de acuerdo a las observaciones del docente y las necesidades de los	Agile SPI-Process y Agile SPI-Process en el contexto académico concuerdan con esta actividad en donde se priorizan las oportunidades de mejora a partir de las falencias

		estudiantes.	encontradas.
			Agile SPI-Process en el contexto académico propone una plantilla para identificar y priorizar las oportunidades de mejora.
1	Agile SPI-Process establece que se debe construir una guía general de la mejora dentro de la fase de diagnóstico.	Como resultado de la valoración y priorización en el diagnóstico, en la experiencia se generó un Plan de Mejoramiento adecuado, que permitiera solventar las necesidades encontradas, además de planificarlas en cuestión de tiempo y esfuerzo en el curso de desarrollo intensivo de software. El “cómo” se realizó el plan general de mejora fue a partir de las experiencias de los autores en otros proyectos.	Agile SPI-Process y Agile SPI-Process en el contexto académico concuerdan en esta actividad.
3	Agile SPI-Process en la fase de diagnóstico, define que se debe comunicar el plan general de mejora obtenido.	En la experiencia, se comunicó que se está realizando un proceso de mejora a los estudiantes en el inicio del curso académico de desarrollo de software y en la aplicación web dispuesta para el proyecto.	Agile SPI-Process y Agile SPI-Process en el contexto académico concuerdan en esta actividad.
<b>Fase de Formulación</b>			
3	Agile SPI-Process propone en la fase de Formulación, como primera actividad <i>Evaluar el área</i> , donde se debe escoger un área de la empresa para ser	En la experiencia con el curso académico no se eligió un área específica, ya que sus procesos no son complejos como lo son en una empresa.	Agile SPI-Process en el contexto académico, por el contrario, propone dos actividades Planificar la iteración de mejora y Evaluar y analizar procesos, de tal forma que se

valorada con más profundidad.

definan los objetivos de la iteración y los casos de mejora a implementar en esa iteración.

1

Agile SPI-Process define que se debe formular los casos de mejora en la fase de formulación, esto incluye documentar, analizar, modelar, diseñar y planificar la mejora en el área o aéreas seleccionadas para la mejora.

En la experiencia esta definición fue la más adaptable al curso académico.

Primero, se definieron los objetivos de la fase. Segundo, se describieron todas las técnicas para evaluar y analizar los procesos llevadas a cabo en todas las iteraciones. Finalmente, se diseñaron los procesos teniendo como resultado un conjunto de actividades, roles, productos de trabajo, plantillas y recomendaciones, que permitirán la mejora y serán adaptadas al modelo de proceso en general. Cabe resaltar que la definición de estas actividades se realizaba a medida que se ejecutaban las iteraciones y se identificaban nuevas necesidades.

Agile SPI-Process en el contexto académico define esta actividad como *Diseño de procesos*. En esta actividad se definen los mecanismos para aplicar cada una de las oportunidades de mejora que se llevarán a cabo dentro de los procesos del curso en la iteración; dicha definición incluye las mejores prácticas de ISO 29110. También se propone una plantilla para la descripción de estas oportunidades de mejora.

3

Agile SPI-Process propone ejecutar casos pilotos de mejora mediante pilotos de prueba, es decir cada caso de mejora planeado y diseñado es aplicado en la empresa en pilotos de prueba, esto con el

En la experiencia no se realizaron pilotos de prueba concretamente, por el contrario, cada iteración se aplicó y dependiendo de sus resultados se identificaban nuevas necesidades y se adaptaban las siguientes iteraciones.

En Agile SPI-Process en el contexto académico no se generan casos pilotos de mejora, se aplican las mejoras en los procesos en una iteración y dependiendo de los resultados se realizan modificaciones para las siguientes

	<p>objetivo de evaluar y analizar el comportamiento de los nuevos o mejorados procesos antes de ser institucionalizados o adoptados de una manera formal dentro de la empresa.</p>		iteraciones.
3	<p>Agile SPI-Process determina en la fase de formulación, institucionalizar el área, es decir, tomar el área que ha sido mejorada o creada e implantarla en el proceso de la empresa de una manera formal y documentada. Además es importante que se verifique que el proceso mejorado cuenta con toda la información necesaria para su utilización, puesto que se pretende que luego de capacitar al personal o usuarios de dicho proceso, éste sea utilizado en proyectos a futuro o que estén siendo ejecutados.</p>	<p>En la experiencia no se institucionalizó un área, debido a que como se ha especificado anteriormente no se seleccionó un área específica debido a que en el contexto académico no existen procesos tan complejos como en una empresa. Sin embargo, realizando un equivalente, sí se implementaron modificaciones en el modelo de procesos resultantes de los avances de cada iteración, con la documentación pertinente para su aplicación en proyectos futuros.</p>	<p>Agile SPI-Process para el contexto académico habla de <i>adoptar procesos</i>, pero se hace referencia a que se aplican las mejoras definidas a los procesos del curso en forma preliminar para luego generar los nuevos procesos resultantes.</p>
<b>Fase de Mejora</b>			
1	<p>En la fase de Mejora se propone que se realicen iteraciones con cada una de las áreas de proceso a mejorar o a crear, para cada una de las cuales se realiza un plan del</p>	<p>La fase de Mejora, en la experiencia, consolida todos los resultados de las iteraciones realizadas, a partir de las actividades diseñadas en la fase de formulación.</p>	<p>Agile SPI-Process para el contexto académico define la actividad de <i>Ejecutar Mejora</i>, en ella se ejecutan los mecanismos o herramientas para la aplicación de las oportunidades de mejora, se actualiza el plan de</p>

	<p>caso de mejora basándose en la información recolectada del o de los pilotos de mejora ejecutados en la fase anterior (fase de formulación), esto con el objetivo de realizar estimativos de recursos como personal, tiempo y dinero. En ésta fase las diferentes iteraciones se pueden realizar al igual que en la fase de formulación.</p>	<p>Además en esta fase se actualizaron documentos como el plan de implementación y el informe de implementación cuando se finalizó cada iteración.</p>	<p>implementación en el paso de una iteración a otra y se crea el informe de implementación que evalúa los resultados de la mejora.</p>
<b>Fase de Revisión</b>			
1	<p>En la fase de Revisión de Agile SPI-Process, se propone realizar una retroalimentación donde de acuerdo a la experiencia se identificaron actividades que se pudieron hacer mejor. Esta actividad genera el documento de <i>Reporte de mejora</i>.</p>	<p>En la experiencia, esta retroalimentación se realiza a partir de las conclusiones del proyecto y las lecciones aprendidas referentes al caso de estudio; no se construyó ningún reporte formal sobre la retroalimentación.</p>	<p>Agile SPI-Process para el contexto académico propone seleccionar las experiencias más importantes e influyentes que tuvieron lugar en la ejecución del proyecto de mejora para luego describirlas como lecciones aprendidas en un documento para futuros proyectos.</p>
3	<p>Agile SPI-Process en la fase de revisión define crear una base de conocimiento, que contenga las malas y buenas decisiones realizadas en anteriores ciclos, como soporte a trabajos futuro de mejora.</p>	<p>En la experiencia se construyeron las lecciones aprendidas del caso de estudio (ver Anexo I Digital: Lecciones aprendidas, Sección 1).</p>	<p>Agile SPI-Process para el contexto académico no propone crear una base de conocimiento como tal, sólo las lecciones aprendidas o comentarios en el proceso de mejora.</p>
1	<p>En la fase de revisión, también</p>	<p>En la experiencia sí se realizaron</p>	<p>Agile SPI-Process para el contexto</p>

	<p>se establece la actividad de analizar la fase de la mejora, donde se desarrollan estudios comparativos para determinar si la mejora realizada ha disminuido o aumentado la eficiencia de los procesos de la empresa.</p>	<p>estas comparaciones del antes y después de la mejora, teniendo como referente las deficiencias encontradas las cuales enfocaron el propósito del proyecto de mejora y los indicadores de mejora definidos en el diagnóstico.</p>	<p>académico concuerda con Agile SPI-Process para el contexto industrial en la actividad de <i>analizar el impacto de la mejora</i>. Sin embargo, se especifica realizar este análisis evaluando la mejora a partir de dos perspectivas:</p> <ul style="list-style-type: none"> <li>• Realizar la evaluación de la mejora desde la perspectiva de los Indicadores de Mejora definidos.</li> <li>• Realizar la evaluación de la mejora desde la perspectiva de las deficiencias encontradas.</li> </ul> <p>Para cada caso Agile SPI-Process académico define plantillas de orientación.</p>
<p>3</p>	<p>En la fase de revisión, se analiza el compromiso y patrocinio, es decir, asegurarse de que el compromiso e interés de la gerencia todavía se mantiene reflejado por el patrocinio que existe para el próximo ciclo.</p>	<p>En la experiencia no se tiene un patrocinio establecido en cuanto a un aporte económico, de recursos, etc., por el contexto en el que se está trabajando. Sin embargo, podría realizarse una equivalencia con el apoyo de los recursos de la universidad y la disponibilidad del docente que dicta el LSD que permitieron la aplicación de este proyecto de mejora</p>	<p>Agile SPI-Process para el contexto académico no considera esta actividad por la naturaleza del contexto.</p>
<p>3</p>	<p>Finalmente Agile SPI-Process, en la fase de revisión, establece</p>	<p>En la experiencia la responsabilidad de las ejecutoras del proyecto</p>	<p>Agile SPI-Process académico, al igual que Agile SPI-Process, incluye</p>

	<p>la preparación del siguiente ciclo teniendo en cuenta el análisis del compromiso y patrocinio y las lecciones aprendidas creando una propuesta de mejora más madura en la fase de instalación del próximo ciclo.</p>	<p>abarca sólo el primer ciclo de mejora. No obstante, se elaboraron una serie de recomendaciones y trabajo futuro (ver capítulo 5) que establecen un apoyo para otros ciclos de mejora en el contexto académico y propiamente en el curso de LSD, el caso de estudio.</p>	<p>la preparación del siguiente ciclo de mejora para el curso respectivo.</p>
<p>1</p>	<p>Agile SPI-Process no incluye una actividad de refinar el modelo de procesos en la fase de revisión.</p>	<p>En la experiencia se refinó el modelo de procesos del curso de LSD durante esta fase.</p>	<p>Agile SPI-Process académico propone que en la fase de revisión se refine el modelo de procesos del curso, adicionando los procesos adaptados a las mejoras e incluyendo los mecanismos, herramientas y demás actividades agregadas al curso.</p>

**Tabla 29.** Comparación de las definiciones de Agile SPI-Process, la experiencia en el LSD y Agile SPI-Process Académico.

# Capítulo 5 – Conclusiones, Recomendaciones y Trabajo Futuro

---

Agile SPI-Process a través del estudio presentado en esta tesis ha resultado adecuado en la implementación de proyectos de mejora de procesos en un contexto académico, debido a que el desarrollo de este trabajo resultó de gran utilidad y aplicabilidad. Así mismo, Agile SPI es una metodología flexible, lo cual se ve reflejado en la capacidad de adaptación que esta posee, dejando que se agreguen, eliminen o actualicen algunas actividades, pasos y artefactos. La experiencia indicó que adaptar las fases de Agile SPI – Process en el proyecto de Mejora en este contexto es posible. Sin embargo, los pasos sobre el cómo hacerlo no están claramente definidos en el modelo original, es decir, no es posible identificar cuál es el camino apropiado para la realización de cada una de sus fases. Por lo tanto, el experto debe buscar este “cómo” en base a otras metodologías o en su experiencia adecuando las actividades y la definición que establece en cada fase Agile SPI-Process. Por ello esta tesis ha propuesto y evaluado una adaptación específica de Agile SPI-Process en el contexto académico.

Algunas conclusiones más puntuales obtenidas de la experiencia, así como las recomendaciones y el trabajo futuro se destacan a continuación:

## 5.1 CONCLUSIONES

- Después del análisis de las actividades definidas por Agile SPI – Process se puede determinar que no muestra detalles de cómo implementar las mejoras en forma concreta; en busca de ello, se estableció un procedimiento definido de cómo implementar un plan de mejora adoptando el uso de herramientas que implementen las mejores prácticas y permitan la aplicación completa de dicho plan sobre un contexto académico. La aplicación de este procedimiento evidencia que existe la necesidad de tener una guía específica que conduzca el proyecto de mejora.
- Al realizar el diagnóstico propuesto se definió que las actividades referentes a priorizar y evaluar oportunidades de mejora constituyen una actividad importante en el desarrollo del proyecto de mejora, ya que se determinan cuáles oportunidades tienen mayor relevancia e impacto ante las necesidades del curso. Esto evita que se implementen mejoras innecesarias que conlleven al desperdicio de esfuerzo perdiendo así el objetivo del proceso.

- Poder determinar cuáles son los procesos dentro de un curso académico y sus oportunidades de mejora, implementando así un plan que permita la mejora de los mismos demuestra que esta experiencia puede ser aplicada en otro curso académico a partir de los resultados satisfactorios del impacto del proyecto de mejora realizado en este trabajo; siendo así una evidencia empírica más en esta línea de investigación.
- Agile SPI-Process no define de manera clara como determinar los indicadores y métricas necesarios en la fase de diagnóstico para determinar los procesos y oportunidades de mejora, por lo tanto se debieron establecer de acuerdo a la experiencia, las necesidades del curso manifestadas por el docente y a lo que se quería obtener en dicha fase. Si bien un marco de medición no es posible de organización a organización, resulta viable en el dominio de cursos de desarrollo de software por tener un contexto de ejecución muy similar.
- A partir del estudio de caso, los valores obtenidos en las encuestas realizadas a los estudiantes, docentes e ingenieros de procesos y los resultados después de la ejecución de la mejora (Ver Capítulo 4, evaluación de la mejora y resultados del estudio de caso), se pudo determinar que Agile SPI-Process académico es útil y que su aplicación es medianamente fácil para un curso dentro del periodo académico respectivo. Útil porque provee las actividades necesarias para mejorar los procesos del curso y medianamente fácil de aplicar porque, si bien expone una guía de su aplicación, requiere un alto nivel de esfuerzo evaluado en el número de horas necesarias para su implementación.
- La experiencia de un estudio de caso para la aplicación de Agile SPI – Process Académico permite concluir que la ISO 29110 utilizada durante el proceso de mejora es adaptable al contexto académico. Este estándar está dirigido a VSE, por lo que la adaptación de alguna de sus definiciones y prácticas es accesible al ingeniero de procesos gracias a la simplicidad con la que esta norma las presenta.
- Para realizar una adaptación de las mejores prácticas definidas en la ISO 29110 se debe realizar un proceso de ingeniería evaluando cada una de las mejores prácticas y seleccionando aquellas que son relevantes respecto a las necesidades y a las restricciones del curso, haciéndolas más concretas para ser aplicadas a un contexto más reducido.
- La adopción de mejores practicas y el uso de herramientas aplicadas a los procesos que se llevan a cabo dentro del curso con Agile SPI – Process Académico, permitieron obtener resultados (Ver Capítulo 4, evaluación de la mejora) que demuestran que se cumplieron con los objetivos de la mejora, mostrando un índice de mejoramiento en los procesos sobre los cuales se realizó la investigación.

## 5.2 RECOMENDACIONES

A continuación se definen algunas recomendaciones por los autores para la realización de proyectos de ese tipo:

- Se definieron métricas de manera empírica y enfocadas al contexto en el que se está realizando el proyecto, pero Agile SPI-Process podría ser mejorado o ajustado de manera que se establecieran medidas generales que sirvan para cualquier contexto.
- Se sugiere que al implementar Agile SPI – Process Académico se realicen mejoras tempranas que permitan refinar el plan de mejora construido en la fase de diagnóstico antes de ser puesto en ejecución sobre los procesos de un curso contando con la experiencia de estas mejoras ya implementadas.
- Ejecutar las mejoras que se han propuesto aquí de tal forma que se observe una evolución del proceso, en la que los estudiantes del curso académico sean los más beneficiados en cada uno de los periodos que continúen de dicho curso.
- Determinar otras oportunidades de mejora en otras áreas del curso que permitan abordar todos los actores de la misma manera, involucrados en los diferentes procesos del curso.
- Dependiendo del contexto sobre el cual se vaya a implementar un proceso de mejora, se hace necesario analizar a profundidad un estándar de calidad adecuado que permita obtener actividades y mejores prácticas adaptables a este contexto.

## 5.3 TRABAJO FUTURO

A partir de este trabajo se pueden emprender trabajos futuros como los descritos a continuación:

- Experimentar el procedimiento para aplicar Agile SPI-Process en el contexto académico, definido en este trabajo, en otros cursos académicos de desarrollo intensivo de software. Dicha experiencia podría generar nuevas adaptaciones o recomendaciones sobre Agile SPI-Process Académico.
- El estudio de caso de este trabajo y el Agile SPI-Process Académico propuesto no abarca pruebas sobre los procesos. De tal forma que se podría ejecutar un trabajo futuro que se enfoque en las pruebas de los procesos llevados a cabo tanto por los estudiantes como por el docente; de manera que se haga entrega de baterías de prueba para cada uno de las

prácticas y requerimientos ejecutados en el curso, lo que le permitiría a los estudiantes probar sus módulos realizados y al docente agilizar la calificación de los mismos.

- El enfoque de mejora de procesos es en la actualidad uno de los más usados y el que ha obtenido buenos resultados, de manera que es importante continuar con el desarrollo de proyectos que estén relacionados.

Para ver las lecciones aprendidas de este trabajo diríjase al Anexo I Digital Lecciones aprendidas, Sección 2.

## Capítulo 6 - Bibliografía

---

[1] LONDOÑO Luis Fernando, “Recomendaciones para la Formación de una Empresa de Desarrollo de Software Competitiva en un País como Colombia”, Centro de Investigación AVANSOFT S.A., 2005.

[2] PRESSMAN, Roger, Ingeniería Del Software, “Un Enfoque Práctico”. McGraw - Hill. 5ª Edición”, 2002.

[3] Institute of Electrical and Electronics Engineers (IEEE), IEEE Standards Association, “610-1990 - IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries”, Sitio Web, Disponible en: <http://standards.ieee.org/findstds/standard/610-1990.html>

[4] PARDO Cesar, HURTADO Julio Ariel, COLLAZOS Cesar, “Mejora de procesos de software ágil con agile SPI process”, Universidad del Cauca, 2009, Sitio Web, Disponible en: <http://revistas.unal.edu.co/index.php/dyna/article/viewFile/25595/26067>

[5] MAS Antonia, AMENGUAL Esperanza, “La mejora de los procesos de software en las pequeñas y medianas empresas (PYME). Un nuevo modelo y su aplicación a un caso real”, Revista Española de Innovación, Calidad e Ingeniería del Software, 2005, Sitio Web, Disponible en: <http://redalyc.uaemex.mx/pdf/922/92210203.pdf>

[6] LÓPEZ, Andrea, “Modelado y Propuesta de Mejora de Procesos de Desarrollo para un entorno académico”, Tesis Pregrado, Universidad del Cauca, 2012.

[7] ISO/IEC 29110, “Software Engineering -- Lifecycle Profiles for Very Small Entities (VSEs)”, 2011, Sitio Web, Disponible en: <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>

[8] Technical Report - ISO/IEC TR 29110-1, “Software engineering – Lifecycle profiles for Very Small Entities (VSEs)”

[9] International Organization for Standardization (ISO), “ISO/IEC 12207:2008”, Sitio Web, Disponible en: [http://www.iso.org/iso/catalogue\\_detail?csnumber=43447](http://www.iso.org/iso/catalogue_detail?csnumber=43447)

[10] “ISO/IEC 12207, IEEE/EIA 12207”, Sitio Web, Disponible en: <http://www.slideshare.net/Yabizyta/iso-12207>

[11] El portal de la norma ISO 15504 en castellano, Sitio Web, Disponible en: <http://www.iso15504.es/>

[12] International Organization for Standardization, Sitio Web, Disponible en: <http://www.iso.org/>

[13] “CMMI Guía para la integración de procesos y la mejora de productos”, Segunda edición, Sitio Web, Disponible en: <http://www.sei.cmu.edu/cmmi>

[14] HURTADO Julio Ariel, VIDAL Juan Carlos, PARDO Cesar, Fernández Luis, “Proyecto SIMEP-SW Agile SPI - Process”, Universidad del Cauca, 2006.

[15] HURTADO Julio Ariel y BASTARRICA Cecilia, “Proyecto SIMEP – SW, Trabajo de Investigación: Hacia una Línea de Procesos Agiles Agile SPsL”, Versión 1.0, 2005, Sitio Web, Disponible en: [http://swp.dcc.uchile.cl/TR/2005/TR\\_DCC-2005-008.pdf](http://swp.dcc.uchile.cl/TR/2005/TR_DCC-2005-008.pdf)

[16] Comunidad MoProSoft, “Modelo de Procesos para la Industria Software MoProSoft”, Versión 1.3, 2005, Sitio Web, Disponible en: [http://www.comunidadmoprosoft.org.mx/COMUNIDAD\\_MOPROSOFTADM/Documentos/V1.3\\_MoProSoft.pdf](http://www.comunidadmoprosoft.org.mx/COMUNIDAD_MOPROSOFTADM/Documentos/V1.3_MoProSoft.pdf)

[17] Proyecto COMPETISOFT, “COMPETISOFT, Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica”, Versión 1.0, 2008, Sitio Web, Disponible en: [http://alarcos.inf-cr.uclm.es/competisoft/publico/downloads/VersModelos/4\\_Nov\\_2008%20COMPETISOFT%20Modelo%20de%20Procesos%20V%201\\_0\\_Definitivo.pdf](http://alarcos.inf-cr.uclm.es/competisoft/publico/downloads/VersModelos/4_Nov_2008%20COMPETISOFT%20Modelo%20de%20Procesos%20V%201_0_Definitivo.pdf)

[18] JACCHERI Mariz Letizia, LAGO Patricia, “Applying Software Process Modeling and Improvement in Academic Setting”, Dipartimento di Automatica e Informatica, Politecnico di Torino.

[19] HAIRUL Mohd, AALYAA Nur, SANIM Shukor, HASHIM Mohd, “Implementation of the Personal Software Process in Academic settings and current support tools”, en *Software Process Improvement and Management: Approaches and Tools for Practical Development*, Sanim Shukor, Nizam Hairul, Ramli Nuraminah, Sahibuddin Shamsul, 2011.

[20] DE AMESCUA Antonio, GARCÍA Javier, SÁNCHEZ María Isabel, MARTÍNEZ Paloma, LLORENS Juan, “An experience of software Process Improvement Applied to Education: The personal work planning technique”, Universidad de Madrid Carlos III, España, 2002.

[21] PEREZ Beatriz, PEDRANA Lucia, BELLINI Marcelo, “MUM – Proceso de Desarrollo de software Modularizado, Unificado y Medible”, Universidad de la

Republica Montevideo, Uruguay, Sitio Web, Disponible en: <http://www.fing.edu.uy/~bperez/public/Proceso%20de%20desarrollo%20de%20Software%20Modularizado%20Unificado%20y%E2%80%A6.pdf>

[22] GARCIA Gabriel, RODRIGUEZ Josefina, “Aplicación del modelo de procesos en un curso de Ingeniería del Software”, Departamento de Ciencias de la Computación, Centro de Investigación Científica y de Educación Superior de Ensenada, México, Sitio Web, Disponible en: <http://redie.uabc.mx/contenido/vol3no2/contenido-mireles.pdf>

[23] CERVANTES Humberto, MARTINEZ Alfonso, "El proceso de desarrollo del programa de mejora UAMISoft", Universidad Autónoma Metropolitana-Iztapalapa, México, Sitio Web, Disponible en: [www.humbertocervantes.net/papers/TIS2005.pdf](http://www.humbertocervantes.net/papers/TIS2005.pdf)

[24] RUNESON, Peter, HÖST Martin, “Guidelines for conducting and reporting case study research in software engineering”, Empirical Software Engineering, 2009. 14 (2): p. 131-164. Sitio Web, Disponible en: [http://www.cse.chalmers.se/~feldt/advice/runeson\\_2009\\_emse\\_case\\_study\\_guidelines.pdf](http://www.cse.chalmers.se/~feldt/advice/runeson_2009_emse_case_study_guidelines.pdf)

[25] BENBASAT Isak, GOLDSTELN David K., MEAD Melissa, “The case research strategy in studies of information systems. MIS Q.”, 1987. 11(3): p. 369-386. Sitio Web, Disponible en: <http://people.cs.aau.dk/~jans/courses/hci-courses/fm-2010/Benbasat87-CaseResearch.pdf>