



**PROCESO PARA LA EVOLUCIÓN Y MANTENIMIENTO DE SOFTWARE EN
LAS MIPYMES Y LOS DEPARTAMENTOS DE DESARROLLO DE SOFTWARE
EN LA CIUDAD DE PEREIRA (MANTELASOFT)**

ANDRÉS FELIPE MONTOYA RÍOS

LINA MARÍA SUÁREZ VÁSQUEZ

UNIVERSIDAD AUTÓNOMA DE MANIZALES

FACULTAD DE INGENIERÍAS

MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE

MANIZALES

2019

**PROCESO PARA LA EVOLUCIÓN Y MANTENIMIENTO DE SOFTWARE EN
LAS MIPYMES Y LOS DEPARTAMENTOS DE DESARROLLO DE SOFTWARE
EN LA CIUDAD DE PEREIRA (MANTELASOFT)**

ANDRÉS FELIPE MONTOYA RÍOS

LINA MARÍA SUÁREZ VÁSQUEZ

**Proyecto de grado para optar al título de Magíster en Gestión y Desarrollo de
Proyectos de Software**

Tutor

JAIRO IVÁN VÉLEZ BEDOYA

UNIVERSIDAD AUTÓNOMA DE MANIZALES

FACULTAD DE INGENIERÍAS

MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE

MANIZALES

2019

RESUMEN

En libros y artículos como “*Software Engineering: A Practitioner's Approach*” (Pressman R. S., 2008) “*Ingeniería del Software*” (Sommerville, 2011), se vienen hablando del tema de mantenimiento de software, situándolo entre el 67% y 90% del costo total del ciclo de vida del desarrollo de un sistema de información. A pesar del esfuerzo de estos y otros estudiosos se reportan pocas investigaciones o producción de datos sobre el tema.

Hoy por hoy, el mantenimiento de software es uno de los procesos más necesarios e importantes para las empresas, pero, simultáneamente, uno de los más complicados y costosos por lo que, en el presente trabajo, se propone una guía denominada “Mantelasoft”, enfocada en el proceso de evolución y mantenimiento de software para las MiPymes y departamentos de desarrollo de software, corresponde a una combinación entre el proceso Ágil Mantema y los procesos de apoyo planteados por la norma ISO/IEC 12207:2008, la cual incluye un modelo con fases, actividades, métricas y técnicas.

En la primera fase se recolectan datos, por medio de una encuesta y un grupo focal a empresarios, sobre cómo algunas empresas de Pereira realizan el mantenimiento de software a sus productos. Luego de analizar la información obtenida y revisar guías de mantenimiento de software ya existentes, se genera y propone “Mantelasoft”. Por último, se aplica “Mantelasoft” en dos empresas de Pereira, donde antes y después de la aplicación se hace una evaluación del mantenimiento de software que allí se realiza, para posteriormente saber las fortalezas y puntos por mejorar de “Mantelasoft”.

Palabras clave—Mantenimiento de Software, SPI, Calidad de Software, Mejora de Procesos, MiPyme, Mantelasoft.

ABSTRACT

In books and articles such as “*Software Engineering: A Practitioner's Approach*” (Pressman R. S., 2008), “*Software Engineering*” (Sommerville, 2011), which since the 70s, have been talking about the issue of software maintenance, placing it among the 67 % and 90% of the total cost of the life cycle of the development of an information system. Despite the efforts of these and other scholars, few investigations or production of data on the subject are reported.

Today, software maintenance is one of the most necessary and important processes for companies, but, simultaneously, one of the most complicated and costly, so, in the present work, a methodological guide is proposed that has been called “Mantelasoft”, focused on the process of evolution and maintenance of software, specifically for MSMEs and software development departments, which corresponds to a combination between the Agil Mantema process and the support processes proposed by ISO / IEC 12207: 2008, which includes a model with phases, activities, metrics and techniques.

Where in the first phase data is collected, through a survey and a focus group of entrepreneurs, on how some Pereira companies perform software maintenance on their products. After analyzing the information obtained and reviewing existing software maintenance guides, “Mantelasoft” is generated and proposed. Finally, “Mantelasoft” is applied in two companies in Pereira, where before and after the application an evaluation is made of the software maintenance performed there, to later know the strengths and points to improve of “Mantelasoft”.

Keywords - Software maintenance, software quality, maintenance metrics, maintenance tools, Mantelasoft.

Contenido

1	PRESENTACIÓN.....	13
2	ANTECEDENTES.....	15
3	ÁREA PROBLEMÁTICA.....	23
4	JUSTIFICACIÓN.....	32
5	REFERENTE TEÓRICO.....	35
5.1	ESTADO ACTUAL DEL MANTENIMIENTO EN PEQUEÑAS ORGANIZACIONES Y MIPYMES EN LA CIUDAD DE PEREIRA	35
5.2	CONCEPTOS	35
5.2.1	Ingeniería de software	35
5.2.2	Proceso	36
5.2.3	Mantenimiento de software	37
5.2.4	Tipos de Mantenimiento.....	38
5.2.5	Evolución del software	39
5.2.6	Proceso Ágil	39
5.2.7	Scrum.....	40
5.2.8	Calidad del software	41
5.2.9	Mantenibilidad.....	41
5.2.10	Modularidad	42
5.2.11	Modelo de calidad de proceso	42
5.2.12	Competisoft	43
5.2.13	Índice de Madurez del Software (IMS).....	43
5.2.14	Leyes de Lehman.....	44

5.2.15	Norma ISO/IEC 12207	45
5.2.16	Ágil Mantema	46
5.2.17	Relación y Comparación entre Agil_MANTEMA y MANTELASOFT 50	
5.3	NORMATIVIDAD	54
5.3.1	ISO/IEEE 14764 - Software Life Cycle Processes - Maintenance (IEEE, 2006) 54	
5.3.2	ISO/IEC 29110 - Systems and software engineering — Lifecycle profiles for Very Small Entities (VSEs).....	55
6	OBJETIVOS.....	56
6.1	Objetivo General.....	56
6.2	Objetivos Específicos	56
7	ALCANCES Y LIMITACIONES	57
8	METODOLOGÍA	58
8.1	TIPO DE ESTUDIO	58
8.2	UNIDAD DE ANÁLISIS	62
8.3	TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN	62
8.4	DESARROLLO DEL PROYECTO	67
8.4.1	Diagnóstico inicial del Mantenimiento de Software	67
8.4.2	Elaboración de MANTELASOFT.....	71
8.4.3	Aplicación de MANTELASOFT en dos MIPYMES desarrolladoras de software 72	
9	RESULTADOS	98

9.1	CARACTERIZACIÓN DE LAS PRÁCTICAS DEL MANTENIMIENTO DE SOFTWARE EN DEPARTAMENTOS DE SISTEMAS Y MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA.	98
9.1.1	Tabulación de la información de la caracterización de las empresas en mantenimiento de software.....	99
9.1.2	Análisis de resultados de la encuesta de caracterización.....	130
9.1.3	Conclusiones de la encuesta de caracterización	131
9.2	Tabulación y gráfico de la información Encuesta Post-Aplicación de la guía	132
9.2.1	Análisis de resultados de la encuesta post-aplicación de MANTELASOFT.....	142
9.2.2	Conclusiones de la encuesta post-aplicación de MANTELASOFT....	142
10	DISCUSIÓN DE RESULTADOS.....	145
11	CONCLUSIONES.....	148
12	RECOMENDACIONES	150
13	BIBLIOGRAFÍA	152
14	ANEXOS	156

Lista de Tablas

Tabla 1. Comparación entre MANTELASOFT y Agil_Mantema.....	53
Tabla 2. Respuestas al cumplimiento de cada actividad de la guía de mantenimiento	64
Tabla 3. Rangos de interpretación del cumplimiento de la guía de mantenimiento .	65
Tabla 4. Evaluación inicial mantenimiento correctivo urgente – Empresa 1.....	74
Tabla 5. Evaluación inicial mantenimiento correctivo no urgente, perfectivo y preventivo – Empresa 1	75
Tabla 6. Evaluación inicial mantenimiento adaptativo –Empresa 1	76
Tabla 7. Evaluación final mantenimiento correctivo urgente –Empresa 1	83
Tabla 8. Evaluación final mantenimiento correctivo no urgente, perfectivo y preventivo –Empresa 1	84
Tabla 9. Evaluación final mantenimiento adaptativo –Empresa 1	85
Tabla 10. Evaluación inicial mantenimiento correctivo urgente –Empresa 2.....	87
Tabla 11. Evaluación inicial mantenimiento correctivo no urgente, perfectivo y preventivo –Empresa 2	88
Tabla 12. Evaluación inicial mantenimiento adaptativo –Empresa 2	89
Tabla 13. Evaluación final mantenimiento correctivo urgente –Empresa 2	95
Tabla 14. Evaluación final mantenimiento correctivo no urgente, perfectivo y preventivo –Empresa 2	96
Tabla 15. Evaluación final mantenimiento adaptativo –Empresa 2.....	97
Tabla 16. Resultados y Productos Logrados	143

Lista de Figuras

Figura 1. Comparación de costos entre Mantenimiento y Nuevos Desarrollos	23
Figura 2. Distribución del esfuerzo en los diferentes cambios	25
Figura 3. Población de Software de EE. UU. en Desarrollo y Mantenimiento.....	26
Figura 4. Estructura general de Agil_MANTEMA.....	47
Figura 5. Niveles de servicios definidos en Agil_MANTEMA.....	48
Figura 6. Proceso de Agil_MANTEMA	49
Figura 7. Relación entre MANTELASOFT y Agil_MANTEMA	51
Figura 8. Fases y actividades en el ciclo PHVA	59
Figura 9. Tarjeta de invitación al grupo focal	67
Figura 10. Listado de empresas interesadas en de participar en el proceso	68
Figura 11. Listado de asistencia al grupo focal.....	69
Figura 12. Asistentes al grupo focal.....	69
Figura 13. Asistentes al grupo focal.....	70
Figura 14. Años de creación de la empresa.....	99
Figura 15. Actividad de la empresa.....	100
Figura 16. Certificación de calidad	101
Figura 17. Realización de mantenimiento de software	102
Figura 18. Modelo de guía para el mantenimiento.....	103
Figura 19. Conocimiento sobre guías de mantenimiento.....	103
Figura 20. Conocimiento sobre mantenimiento de software.....	104
Figura 21. Tipos de mantenimiento de software que realiza.....	105
Figura 22. Uso de la aplicación de un proceso de mantenimiento.....	105
Figura 23. Tercerización del mantenimiento de software	106
Figura 24. Garantía del software	107
Figura 25. Tiempo de garantía del software.....	108
Figura 26. Planificación de operaciones y recursos	108
Figura 27. Realización del plan de mantenimiento	109
Figura 28. Tiempo de duración del mantenimiento	110

Figura 29. Presupuesto de mantenimiento de software.....	111
Figura 30. Estimación de los costos de mantenimiento de software.....	111
Figura 31. Técnica utilizada para estimar los costos.....	112
Figura 32. Costo del mantenimiento de software.....	112
Figura 33. Estimación de costos de operación del hardware	113
Figura 34. Tiempo para entender cambio, mejora o corrección.....	113
Figura 35. Tiempo de solución de defecto	114
Figura 36. Registro de los defectos	115
Figura 37. Número de defectos corregidos por mes.....	116
Figura 38. Medición de la calidad del código	116
Figura 39. Conocimiento del ÍMS.....	117
Figura 40. Recursos para el mantenimiento	118
Figura 41. Realización del impacto de la corrección o mejora del producto	119
Figura 42. Responsable de realizar el análisis.....	119
Figura 43. Responsable de realizar correcciones	120
Figura 44. Ocurrencia de daños al realizar correcciones	121
Figura 45. Realización del plan de retiro del producto	122
Figura 46. Realización de retiro de software.....	122
Figura 47. Realización de documentación de acciones de mantenimiento	123
Figura 48. Principales errores en el mantenimiento de software	124
Figura 49. Estado del personal de mantenimiento	125
Figura 50. Existencia de herramientas para registro de defectos	126
Figura 51. Existencia de herramientas para la gestión de la configuración y versiones	127
.....	
Figura 52. Existencia de herramientas para la gestión del conocimiento	127
Figura 53. Realización de mejoras y evolución por cuenta propia	128
Figura 54. Tiempo de perfeccionamiento	129
Figura 55. Conocimiento de las Leyes de Lehman	129
Figura 56. Conocimiento y dominio de temas de la guía.....	133
Figura 57. Claridad y orden de la guía	133

Figura 58. Suficiencia de contenido de la guía	134
Figura 59. Utilidad de la información de la guía.....	135
Figura 60. Ejecución de la guía.....	135
Figura 61. Generación de aportes por la guía.....	136
Figura 62. Uso de la guía en actividades futuras.....	137
Figura 63. Necesidad de añadir o eliminar pasos de la guía	138
Figura 64. Ejecución de la guía para calidad del software.....	138
Figura 65. Ayuda de la guía para encontrar problemas.....	139
Figura 66. Ayuda de la guía para el conocimiento de los costos de mantenimiento	140
Figura 67. Planeamiento de la guía para realizar mejoras a los productos	141

Lista de Anexos

Anexo A. Encuesta para el diagnóstico inicial del Mantenimiento de Software	157
Anexo B. Formato de la Encuesta de Evaluación Inicial y Final de la aplicación de MANTELASOFT.....	160
Anexo C. Formato de la Encuesta de Evaluación Post Aplicación de MANTELASOFT.....	161
Anexo D. Costo total de la investigación.....	162

1 PRESENTACIÓN

El presente trabajo está enfocado, específicamente, en el mantenimiento de software, la situación que viven las Mipymes de Pereira frente a este proceso y una propuesta de solución. El mantenimiento de software es la modificación de un producto de software después de la entrega, en cuyo caso se requiere corregir errores, mejorar el rendimiento, u otros atributos (IEEE, 2006).

El presente trabajo se realizó con la convicción que es necesario dotar a las empresas de métodos que faciliten el proceso de mantenimiento de software, por lo que el objetivo general se direccionó hacia el diseño de un proceso para la evolución y mantenimiento de software, lo que se realizó desde dos grandes plataformas, la primera, la adaptación del proceso Ágil Mantema y los procesos de apoyo de la norma ISO/IEC 12207 y la segunda, la aplicación de una encuesta utilizada para caracterizar las prácticas del mantenimiento de software de las Mipymes de Pereira.

Cabe aclarar aquí que, históricamente, las empresas que requieren aplicar procesos de mantenimiento de software utilizan la guía de mantenimiento Ágil Mantema, sin embargo, los empresarios, en diferentes espacios académicos, han manifestado que esta herramienta es compleja, contiene demasiados pasos y exige un número de colaboradores con dedicación exclusiva a su implementación, incrementando los costos de dicho mantenimiento y complicando su aprovechamiento. Por tanto, para la presente investigación se propuso utilizar algunas bases de la guía Mantema, pero ajustada a los requerimientos, necesidades y características de las Mipymes de la región.

Es así como, los objetivos específicos se dirigieron a caracterizar las prácticas del mantenimiento de software en los departamentos de sistemas y las Mipymes desarrolladoras de software de la ciudad de Pereira, formular el proceso para la evolución y mantenimiento de software llamado “Mantelasoft”, aplicación de este proceso en las Mipymes y departamentos y validación de su pertinencia.

Es necesario aclarar aquí que la fase inicial correspondió a un estudio exploratorio, realizado para conocer el estado de mantenimiento de software en la ciudad de Pereira, el

tipo de mantenimiento que se realiza, los cambios que se presentan, los tipos de incidencias y los procesos que aplican, entre otros.

En esta primera búsqueda de información se llevó a cabo un grupo focal con los responsables o encargados de los departamentos de desarrollo y de las Mipymes desarrolladoras de software de Pereira que están registradas en la Cámara de Comercio y que, tras invitación, manifestaron interés en participar en este estudio. A los empresarios convocados que no asistieron al grupo focal, se les envió una encuesta para completar el análisis.

Con la información obtenida en la etapa de investigación y desde el ciclo de mejoramiento de calidad de procesos PHVA de Deming, se abordan las diferentes fases del trabajo. En la fase “Planificación” se elaboró un nuevo y creativo proceso de mantenimiento, se planeó su aplicación y se definieron las conclusiones. Se puede decir aquí que todo este trabajo investigativo y de aplicación arrojó como resultado una guía, que como se comentó antes, fue denominada “MANTELASOFT”, la cual describe el proceso para la evolución y el mantenimiento de software de los departamentos de desarrollo de software de las Mipymes de la ciudad de Pereira.

En la fase “Hacer” se aplicó la nueva guía de mantenimiento en dos empresas de la ciudad de Pereira. Esto se revisó en la fase “Verificar”, en la que se analizaron los resultados obtenidos de dos evaluaciones, Pre/Post, la primera a través de una encuesta realizada al iniciar el ejercicio para saber cómo estaba la empresa en el tema de mantenimiento antes de la aplicación de la guía y la segunda, correspondiente a una encuesta diseñada para analizar la efectividad del proceso y su validez para ser aplicado en otras empresas. Los datos de la encuesta realizada al finalizar la intervención se convirtieron en insumo fundamental para plantear modificaciones y perfeccionar la propuesta.

Cabe anotar que se desarrolla el ciclo de mejoramiento PHVA, ya que es un ciclo dinámico que permite hacer un análisis medible de cada paso donde P (planear) se diseña la guía, se decide qué medir y cómo será aplicada, H y V (hacer, verificar) permiten aplicar la guía y medir los indicadores que se tuvieron en cuenta en la planificación y A (actuar) hacer los ajustes a la guía y a las mediciones para futuros trabajos.

2 ANTECEDENTES

De orden Nacional se da una propuesta en el Crecimiento de la industria del software en Colombia: un análisis sistémico (Martínez Marín, Arango Aramburo, & Robledo Velásquez, 2015). En esta investigación se analiza la alta competitividad industrial y sectorial, arrojando resultados que facilitaron la construcción de la matriz DOFA, con ellas se desarrollaron encuestas con el objetivo de evaluar el proceso de toma de decisiones estratégicas en las empresas. Los interrogantes que conforman el cuestionario nacen de los resultados obtenidos en la matriz DOFA, punto en el que se evidencia la necesidad de aumentar las capacidades relacionadas con la formación y especialización del recurso humano, tanto a nivel técnico-operacional, como estratégico-gerencial.

Estos resultados muestran que el deficiente direccionamiento estratégico y gestión empresarial en las empresas nacionales que trabajan en el desarrollo de software, es el resultante del estado actual de inmadurez y desarticulación de la industria, sumado a que el proceso de decisiones estratégicas se encuentra sesgado por el área de formación profesional de sus dirigentes. Con base en lo anterior, se propone como metodología de solución en la ejecución de este proyecto, el desarrollo de un modelo de simulación con dinámica de sistemas, que permita la evaluación de estrategias de innovación relacionadas con la capacidad de investigación y desarrollo (I+D), direccionamiento estratégico y marketing, el proceso de caracterización de las prácticas del mantenimiento de software, en los departamentos de sistemas o MIPYMES desarrolladoras de software de la ciudad de Pereira, mediante la aplicación de una encuesta; y poder así comprender aún más el estado actual de éstas.

En uno de los informes encontrados sobre la caracterización del sector de software y tecnologías de la información en Colombia (Ministerio de las Tecnologías de la Información y la Comunicación, 2015). El objetivo fundamental de esta investigación es realizar una caracterización de la industria de Software y Tecnologías de la Información a partir de un análisis de la evolución de las principales variables económicas y del comportamiento de las empresas que concurren en el mercado.

El informe contiene cuatro partes: en la primera se analizan los resultados del estudio de MinTIC “Conformación del Directorio de Empresas Activas de la Industria del Software y Servicios Asociados con TI de Colombia – 2014” en el que se incluye información sobre el comportamiento de las empresas en varios niveles: económico, talento humano, investigación y desarrollo, entre otros aspectos. En la segunda parte, se analizan las principales variables económicas y financieras del sector, para lo que se utilizaron fuentes de información secundarias, tales como la base de datos de declaraciones de IVA y renta suministrados por la DIAN, las cifras de exportaciones del DANE, y los estados financieros reportados a la Superintendencia de Sociedades por las empresas de Software y Tecnologías de la Información. En el tercer capítulo se presentan los resultados de la encuesta aplicada a las empresas del sector de software y tecnologías relacionadas en el 2015, en el marco del estudio de caracterización ocupacional del sector de Teleinformática, Software y TI en Colombia, realizado por FEDESOFTE con el apoyo del SENA y MinTIC.

En la parte final se efectúa una estimación del personal ocupado por el sector de Software y Tecnologías de la Información, en la actualidad y su proyección de crecimiento para los próximos años. Para el proyecto, esta investigación permite dar a conocer el conocimiento sobre la industria de Software y Tecnologías de la Información en Colombia, y la vez sirve de base para llevar a cabo el proceso de caracterizar las prácticas del mantenimiento de software, en departamentos de sistemas o MIPYMES desarrolladoras de software de la ciudad de Pereira.

Por otro lado, a nivel regional se da la Caracterización del proceso de desarrollo de software en Colombia: una mirada desde las PYMES productoras (Peláez Valencia, Toro Lazo, López Restrepo, & Ramírez Medina, 2012)

Con este trabajo investigativo, se socializan los resultados obtenidos al caracterizar la calidad del proceso de desarrollo de software en Colombia, donde se encontraron falencias en la aplicación de buenas prácticas de ingeniería que permitieron a los productores nacionales alcanzar mejores calificaciones frente a modelos de calidad ampliamente reconocidos. Para lograr el propósito, se formulan las cuatro fases hasta llegar a las pruebas del software.

Los problemas del mantenimiento del software se presentan, en gran medida, por factores relacionados con la continuidad del cambio, el incremento de la complejidad, la evolución del programa, la conservación de la estabilidad organizacional y la conservación de la familiaridad. Para el proyecto planeado, este trabajo contribuye con la tarea propuesta de caracterizar las prácticas del mantenimiento de software en las empresas ya que, otorga los resultados que se obtuvieron del estudio aplicado en la población empresarial sobre la calidad del proceso de desarrollo de software.

En la misma línea se encuentra la Caracterización de la industria del software en el Triángulo del Café Colombia (Joyanes Aguilar, Cuesta Meza, & López Trujillo, 2010), para lograr la caracterización de la actual Industria del Software en el Triángulo del Café – Colombia, se realizó un trabajo de campo durante el año 2009, dirigido en primera instancia a localizar e identificar las empresas que en su objeto social hacen explícito el desarrollo de software, la prestación de servicios en informática o la venta de productos de software desarrollados en la región.

Se logró trabajar con una muestra total de 48 empresas desarrolladoras de software en el Triángulo del Café: 20 en la ciudad de Manizales, 16 en Pereira y 12 en Armenia. Posteriormente se hizo el trabajo de campo en cada una de las 48 empresas de la región con la colaboración del Grupo de Investigación de Ingeniería de Software de la Universidad Cooperativa de Colombia Seccional Manizales, el Grupo de Investigación Grande de la Universidad Tecnológica de Pereira y el Grupo de Investigación Sinfoci de la Universidad del Quindío.

Los resultados muestran que los productos y los servicios que ofrecen las empresas desarrolladoras de software de la región requieren un alto nivel de especialización, con el fin de ubicar nichos de mercados específicos, producto de una adecuada segmentación de dicho mercado puesto que en la actualidad el Software a la medida es el producto o servicio que ocupa el primer lugar en oferta. De igual manera, el nivel de ventas de las empresas de software propias de la región no impacta considerablemente la economía regional, haciéndose notoria la falta de adopción de modelos internacionales de calidad en la producción de productos y servicios de software como principal debilidad de la industria

local, puesto que no permite penetrar mercados nacionales e internacionales donde se exige como mínimo esa garantía.

De igual manera al trabajo anterior, esta investigación contribuye con la tarea propuesta de caracterizar las prácticas del mantenimiento de software en las empresas ya que, otorga los resultados que se obtuvieron del estudio aplicado en la población empresarial sobre la calidad del proceso de desarrollo de software.

En la misma línea también se ha investigado y se hizo un estudio de las prácticas de calidad del software implementadas en las Mipymes desarrolladoras de software de Pereira (Ramírez Aguirre & Ramírez Arias, 2010). En esta investigación se llevó a cabo un estudio sobre el grado de implementación de los modelos de calidad en el desarrollo de software de las Mipymes del sector TI en la ciudad de Pereira utilizando el modelo IT Mark para el Laboratorio de Calidad de software de la Universidad Tecnológica de Pereira, identificando el nivel de la calidad del software de las empresas de la región mediante el análisis de los datos recopilados en un formato de encuesta que se elaboró a partir del modelo IT Mark; a fin de establecer pautas a seguir, para que el Laboratorio de Calidad continúe con su misión de que las Mipymes del sector TI puedan llegar a obtener una certificación en IT Mark.

Otro estudio del orden iberoamericano, se proponen varios modelos o metodologías para el proceso de mantenimiento de sistemas de información, El documento de COMPETISOFT, Ágil Mantema (2008) (Pino, Ruiz, & Salas, 2008), metodología que puede verse como *“el conjunto de todas las operaciones que se realizan sobre el software para implementar las modificaciones solicitadas por el cliente y donde se detalla qué debe realizarse, cuándo, cómo y por quién, de tal manera que cada intervención de mantenimiento que se lleve a cabo sea conforme a un proceso de mantenimiento predefinido”* (Osorio Martínez, Irrazabal, Garzás, & Marcos, 2015); busca guiar paso a paso el proceso de mantenimiento del software en pequeñas organizaciones de software.

En su contenido se encuentra la estructura básica para la definición del proceso de mantenimiento a partir de esta guía, presentando gráficamente la estructura general del proceso y explicando la integración de procesos de mantenimiento de software, así como los diferentes tipos de mantenimiento que se consideran y la identificación de las

organizaciones que intervienen en el proceso. Finalmente, se describe las diferentes partes del proceso deteniéndose en cada actividad y tarea, detallando cuidadosamente cada una de ellas (Polo, Piattini, Ruiz, & Calero, 2000), compuestas específicamente por las tareas y actividades iniciales comunes mencionadas anteriormente, las del mantenimiento planificable y no planificable, junto con cada una de las técnicas y métricas propuestas para la ejecución de las dichas tareas y las métricas específicas para el producto.

Es importante tener este antecedente presente para el trabajo pues es una aproximación iberoamericana enfocada específicamente en las pequeñas empresas, que es precisamente el sector en el cual se pretende enfocar esta investigación. Además, tiene una estructura diferente a los antecedentes anteriores, pues está enfocada en proceso ágil como Scrum con procesos definidos y detallados para todas las etapas. Es por estas razones, que “MANTELASOFT” será una adaptación entre el proceso Ágil Mantema, la norma ISO/IEC 12207 y la encuesta utilizada para caracterizar las prácticas del mantenimiento de software en departamentos de sistemas o Mipymes desarrolladoras de software de la ciudad de Pereira.

Otra propuesta es la Aproximación a la Mejora de la Metodología Ágil Mantema para la Adopción del Modelo de Calidad ISO/IEC 15504-ISO/IEC 12207 en PYMEs. El artículo presenta una mejora realizada a la metodología Ágil Mantema. Dicha mejora se realizó comparándola con las normas ISO/IEC 15504 y ISO/IEC 12207 y de este análisis de comparación se determinó los puntos débiles de Ágil Mantema y se describieron las actividades y tareas propuestas. Adicional, se implementó la nueva propuesta en una PYME de México, dando resultados favorables (Osorio Martínez, Irrazabal, Garzás, & Marcos, 2015) Mantema se utilizará para el desarrollo del proceso Mantelasoft.

Otro artículo publicado en una revista española sobre la Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas (Pino, García, & Piattini, 2006), ha presentado una revisión sistemática sobre los esfuerzos SPI llevados a cabo en Pymes desarrolladoras de software, que permite tener una visión muy completa de la situación actual.

La formalidad con la que se lleva a cabo la revisión sistemática permite validar sus resultados, los cuales están soportados y avalados por el protocolo de la revisión, el cual

consta de cinco partes generales: formulación de la pregunta, selección de las fuentes, selección de los estudios, extracción de información y resumen de los resultados. Esta investigación enfoca sus esfuerzos para que los estándares del ISO y SEI puedan ser aplicados de manera apropiada en las micro, pequeñas y medianas empresas ya que actualmente dichas empresas adaptan y utilizan tales estándares para emprender sus esfuerzos de mejora, aunque por lo general no consiguen una certificación.

Otro dato relevante obtenido es que, de los diferentes tipos de estándares disponibles, relacionados con las mejoras de procesos, los más utilizados por las Pymes Desarrolladoras de Software son: CMMI como modelo de procesos, ISO/IEC 15504 como método de evaluación e IDEAL como modelo para guiar la mejora. El artículo sirve como base para la realización del proyecto, ya que sirve como base para la formulación del proceso para la evolución y mantenimiento de software, mediante la formulación de un instrumento basado en los procesos de apoyo planteados por la norma ISO/IEC 12207.

De la misma manera esta investigación sobre el modelo, *Software Maintenance Maturity Model (SMmm): The software maintenance process model* (Alain, Huffman Hayes, Alain , & Reiner , 2004), la cual aborda la evaluación y mejora de la función de mantenimiento de software proponiendo mejoras a los estándares de mantenimiento de software y la introducción de un modelo de madurez propuesto para las actividades diarias de mantenimiento de software denominado Modelo de madurez de mantenimiento de software (SMmm), el cual aborda las actividades únicas de mantenimiento de software manteniendo una estructura similar a la del modelo de madurez CMMI. Este modelo se basa en la experiencia de los practicantes, los estándares internacionales y la literatura sobre mantenimiento de software, presentándose el propósito, alcance, fundamento, arquitectura del modelo y su validación inicial.

Esta investigación aporta en gran medida al segundo y tercer objetivo del proyecto, pues permite formular una guía basada en fases, actividades, métricas, técnicas y herramientas, similares a las ofrecidas por el Modelo de madurez de mantenimiento de software (SMmm), para ser utilizadas en los departamentos de sistemas y Mipymes desarrolladoras de software de la ciudad de Pereira.

De igual manera el trabajo, *Critical Success Factors in Software Maintenance A Case Study* (Sneed & Brössler, 2003), tiene como objetivo identificar los factores críticos para el éxito de una operación de mantenimiento en general y aplicarlos a un proyecto de mantenimiento en particular. Se muestra el mantenimiento y la evolución de un sistema de solicitudes bancarias muy grande y complejo para el procesamiento de valores que ha estado en marcha desde hace varios años, definiendo y evaluando factores del mantenimiento del software como tener los requerimientos claros, aclarar los objetivos, tener un cronograma realista, conocimientos en gestión de proyectos efectivos, soporte desde la dirección del proyecto, involucrar al cliente y al usuario, comunicación efectiva y retroalimentación, entre otros.

Cada uno de estos factores se califica de acuerdo con una escala métrica dada, dejando al evaluador el peso y evaluación de la importancia de los factores individuales. El estudio se basa en datos empíricos recopilados durante la duración del proyecto y su objetivo es contribuir al conocimiento general del mantenimiento del software.

En este caso específico, el artículo presenta su principal aporte para el proyecto en cuanto a la definición de factores básicos y necesarios en una operación de mantenimiento de software, y que se requieren para las fases de la guía de mantenimiento de software propuesta.

Finalmente, se encuentra dentro de los referentes, Métrica V.3, propuesta del gobierno español. “Proceso de mantenimiento de sistemas de información (Ministerio de Administraciones Públicas de España, 2001). Esta propuesta realizada por el Ministerio de Hacienda y Administraciones Públicas del Gobierno de España es un proceso de planificación, desarrollo y mantenimiento de sistemas de información para la sistematización de actividades del ciclo de vida de los proyectos de software en el ámbito de las administraciones públicas.

En éste se describe la estructura de procesos principales para el ciclo de vida de los Sistemas de Información la cual se compone de productos que se deben realizar o entregar, técnicas o prácticas para implementar en el proceso y los participantes que actúan allí; además se encuentran interfaces las cuales son transversales a los procesos. Para el proyecto será de gran utilidad el Proceso de Mantenimiento de Sistemas de Información

(MSI) porque es un proceso que realiza el registro de las peticiones de mantenimiento recibidas, que se complementa con Agil_MANTEMA pues ya tuvo en cuenta los procesos de Métrica V3; el cual tiene como fin llevar el control de estas y de proporcionar, si fuera necesario, datos estadísticos de peticiones recibidas o atendidas en un determinado periodo, sistemas que se han visto afectados por los cambios, en qué medida y el tiempo empleado en la resolución de dichos cambios. Además, es referente para las Mipymes colombianas, ya que brinda las tareas que se deben realizar para el mantenimiento, los productos de entrada y salida, las prácticas y los responsables de dichas tareas. La guía de mantenimiento de software propuesta contiene en su proceso el análisis del error, la intervención correctiva, preventiva, adaptativa y perfectiva, y el cierre de la intervención, cada una compuesta por una serie de actividades a desarrollar.

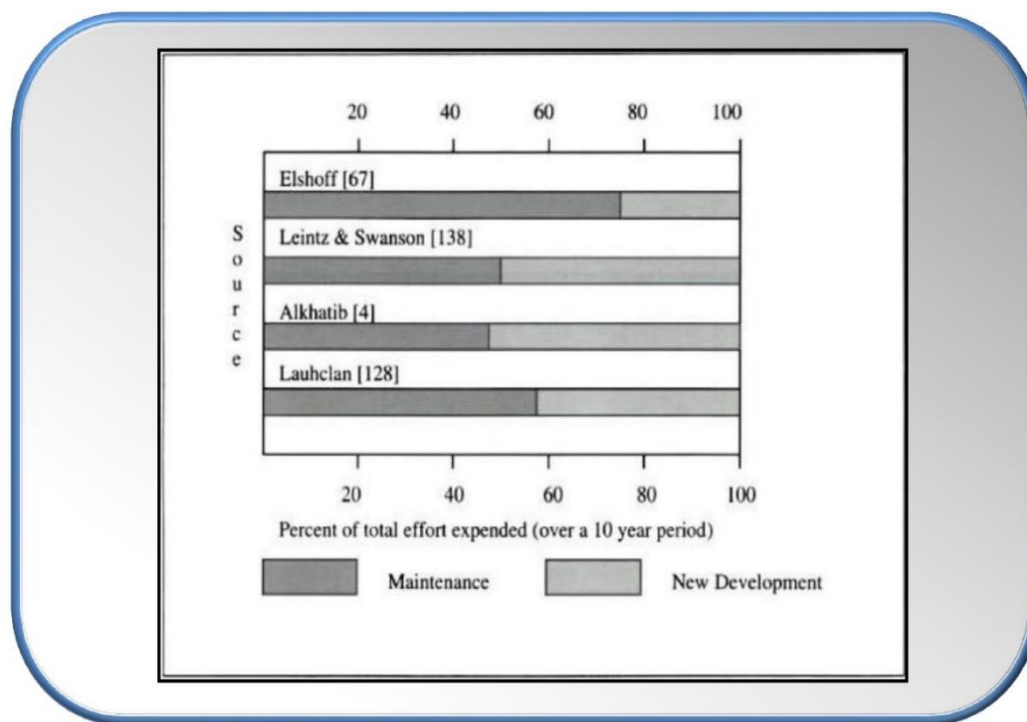
3 ÁREA PROBLEMÁTICA

Desde su nacimiento, el grupo Entre Ciencia e Ingeniería cuenta con el apoyo de un importante número de empresas de la ciudad en las que realiza los diferentes proyectos y de las que se seleccionaron las que se dedican al desarrollo de software siendo 47 empresas las convocadas a participar y 13 las que finalmente respondieron los instrumentos.

La hipótesis en la que se basa la presente indagación está sustentada en la idea que la mayor parte del mantenimiento de software que se realiza en las Mipymes es de tipo correctivo y gran porcentaje de los recursos se destinan al desarrollo de este producto, no dejando pie a la inversión en otros procesos importantes.

En la Figura 1, se muestra un comparativo entre los nuevos desarrollos y el mantenimiento de software, en la cual aparecen los resultados de cuatro autores donde los costos del mantenimiento de software van desde un 49% a un 75%.

Figura 1. Comparación de costos entre Mantenimiento y Nuevos Desarrollos



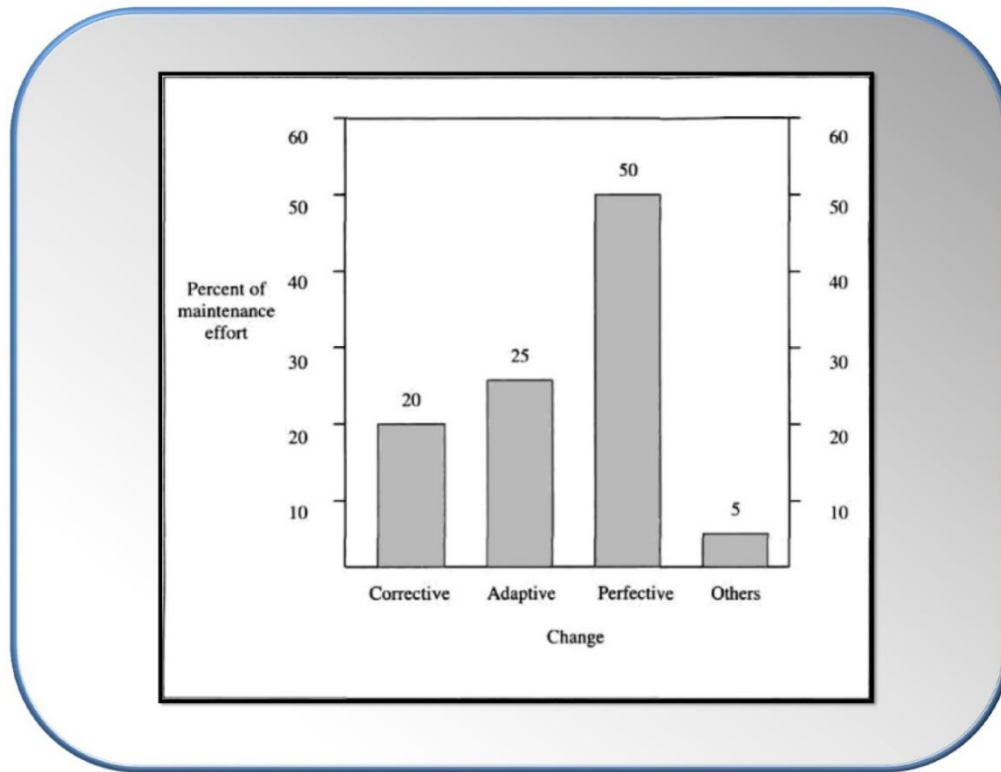
Fuente: (Grubb & Takang, 2003, pág. 49).

En un estudio realizado por Lientz y Swanson, en 1982, en el que se reportaron los cambios que se registraban en 487 organizaciones de procesamiento de datos, se realizó un comparativo que se plasma en la Figura 2. En esta se evidencia que solo el 20% del esfuerzo de mantenimiento es para mantenimiento correctivo, el cual centra sus esfuerzos en corregir defectos que se presentan cuando el producto ya está en operación. Por su parte, el 50% se dedica al mantenimiento perfectivo, que se enfoca en la modificación del producto luego de estar en operación.

El 25% es para mantenimiento adaptativo, el cual se realiza cuando surge un cambio en el ambiente de operación después de ser entregado y, por último, el 5% restante para mantenimiento preventivo, donde se busca encontrar y solucionar errores antes de que el producto entre en operación o se hagan actividades relacionadas como pruebas y auditorías, entre otras.

Hoy por hoy, la situación se sigue presentando de manera similar, como lo expresan Singh & Goel, 2007, quienes coinciden con Lientz y Swanson (1982), tras investigaciones como *A Step Towards Software Preventive Maintenance* (Singh & Goel, 2007, pág. 1). De estos resultados y a partir de un primer acercamiento con los empresarios de la ciudad de Pereira, se detecta, entonces, la problemática descrita, teniendo en cuenta que unos empresarios aducen centrar sus esfuerzos en corregir los defectos cuando el producto ya fue entregado, mientras que otros afirman invertir sus esfuerzos en corregir estos defectos antes de entregar su producto, lo cual puede apuntar a que el producto sea entregado con un mayor grado de calidad.

Figura 2. Distribución del esfuerzo en los diferentes cambios



Fuente: (Grubb & Takang, 2003). (Lientz y Swanson, 1982, citados por Grubb & Takang, 2003)

Varios estudios, sugieren una magnitud similar a la distribución presentada en la gráfica anterior; lo que estos estudios muestran es que las incorporaciones de nuevos requerimientos de usuario pueden ser el problema principal para el mantenimiento y la evolución del software y para el mantenimiento no correctivo (Bennett & Rajlich, 2000), necesitando involucrar más empleados y la carga prestacional que esto implica.

En la Figura 3, se puede evidenciar la variación que ha habido de personal para el desarrollo y para el mantenimiento desde el año 1950, con una proyección al año 2025.

Figura 3. Población de Software de EE. UU. en Desarrollo y Mantenimiento

Year	Development Personnel	Maintenance Personnel	Total Personnel	Maintenance Percent
1950	1,000	100	1,100	9.09%
1955	2,500	250	2,750	9.09%
1960	20,000	2,000	22,000	9.09%
1965	50,000	10,000	60,000	16.67%
1970	125,000	25,000	150,000	16.67%
1975	350,000	75,000	425,000	17.65%
1980	600,000	300,000	900,000	33.33%
1985	750,000	500,000	1,250,000	40.00%
1990	900,000	800,000	1,700,000	47.06%
1995	1,000,000	1,100,000	2,100,000	52.38%
2000	750,000	2,000,000	2,750,000	72.73%
2005	775,000	2,500,000	3,275,000	76.34%
2010	800,000	3,000,000	3,800,000	78.95%
2015	1,000,000	3,500,000	4,500,000	77.78%
2020	1,100,000	3,750,000	4,850,000	77.32%
2025	1,250,000	4,250,000	5,500,000	77.27%

Fuente: (Jones, *The Economics of Software Maintenance in the Twenty First Century*, 2006, pág. 4).

También se puede ver un constante aumento de personal para el mantenimiento y para el proceso de desarrollo desde 1950 hasta 1995 pero, desde ese año en adelante, el personal de desarrollo disminuyó y el de mantenimiento aumentó casi el doble para el 2000, aunque no debido al problema en los sistemas de información denominado Y2K, que se tenía predicho por el cambio de milenio, sino por la tendencia de aumentar la automatización del proceso de desarrollo y por consiguiente, del mantenimiento de software, lo que llevó a prevenir, más que a corregir defectos, al incremento en el uso de *frameworks* para el desarrollo de software, a la gestión de la configuración y la madurez de aplicaciones genéricas que se presentan de manera vertical y horizontal en las organizaciones, como ERP's, CRM's.

Hay estudios que revelan que el coste del mantenimiento de software es más elevado que cualquier otra de las fases del ciclo de vida del software, pues supone más del doble que los costos de su desarrollo y la tendencia cada vez va en aumento. Actualmente el panorama de los costes no se encuentra muy alejado del inicio de la década del 2000, cada

vez son más las empresas desarrolladoras de software y los costes han aumentado en función de la seguridad y funcionalidad de las aplicaciones elaboradas (Jones, Estimación de costos y administración de proyectos de Software, 2008).

En el libro *The Economics of Software Quality* (Jones & Bonsignour, The Economics of Software Quality, 2012) se menciona que a partir del 2011 el costo promedio por punto de función, el cual es utilizado para medir el tamaño del software (ISO/IEC, 2007) en Estados Unidos, es de aproximadamente \$1,000 dólares que son usados para construir aplicaciones de software y otros \$1,000 dólares dedicados al mantenimiento y al soporte por cinco años, siendo, en total, \$2,000 dólares por punto de función (Jones & Bonsignour, The Economics of Software Quality, 2012, pág. 3).

En Colombia, según PIB, la industria del software cuenta con una participación del 2.6% del total de las empresas relacionadas con el sector del Software y Tecnologías de Información y, conforme al informe realizado por PROEXPORT COLOMBIA, (Procolombia, 2014), se visualiza como uno de los de mayor proyección y crecimiento en los próximos años.

Actualmente, al Eje Cafetero han llegado empresas multinacionales extranjeras como INDRA, VERITRANS entre otras y Mipymes de desarrollo de software, lo cual demanda de esta región generar estrategias de desarrollo de software, que cumplan con los estándares de calidad en el ámbito nacional e internacional, para el desarrollo y mantenimiento de software como son la norma ISO/IEC 12207, la cual presenta el estándar para los procesos de vida del software de la organización; la norma ISO/IEC 25000 que es una familia de normas, la cual evalúa los productos de software indicando las características de calidad, los lineamientos para su uso y sus métricas asociadas; y el modelo CMMI-DEV, que es un modelo de mejora de procesos de construcción de software y mantenimiento que provee los elementos necesarios para determinar su efectividad. Todas las anteriores con el fin de suplir la necesidad del mercado y contribuir de manera activa al incremento de la competitividad regional.

Es necesario recalcar aquí que la competitividad y la calidad del software es difícil de medir como se demuestra en el proyecto denominado “Propuesta metodológica para estandarizar el proceso de construcción y evaluación del producto software que permita a la

MIPYMES medir la calidad del software” (Becerra Sanchez, 2007) desarrollado por el grupo de investigación de la Universidad Católica de Pereira.

Este proyecto destaca que es difícil caracterizar la calidad del software desde el momento de su fabricación, cuando las empresas que lo construyen están convencidas de lo bondadoso de sus procesos, lo que las restringe, en cierta forma, de participar de manera objetiva en un diagnóstico de la situación actual con miras a un plan de mejora.

Pressman (2008) sostiene que “la calidad del software es 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”. Es por esta razón que las empresas deben adoptar metodologías y marcos de trabajo que les permitan entregar sus desarrollos con alto grado de calidad y con similares características de los productos desarrollados por las grandes empresas.

Capers Jones Sostiene que,

Un mejor control de calidad de software en las formas de prevención de defectos y una eliminación más efectiva de defectos podría liberar aproximadamente 720.000 empleados de software para un trabajo más productivo que sólo reparaciones de fallos, reduciendo fácilmente los costos de desarrollo y mantenimiento de software en un 50%. (Jones & Bonsignour, The Economics of Software Quality, 2012, pág. 76)

Por tanto, es prioritario enfocarse en producir software a la medida con la finalidad de reducir costes en el mantenimiento y desarrollo, sin embargo, hay que anotar que el mantenimiento de software es una labor que debe estar presente para mantener el software vigente.

El gran desarrollo de la industria en general, apalancado por el proceso de globalización, la estandarización y los tratados de libre comercio, han permitido que la industria del desarrollo de software concentre gran parte de sus esfuerzos en el desarrollo de aplicaciones para distintas plataformas o entornos, incrementando la disponibilidad, confiabilidad e integridad de la información y la demanda de profesionales calificados que

contribuyan a desarrollos de software de alta calidad en estos ambientes (Nasir & Abbasi, 2010).

Es por esto que se viene desarrollando una labor complementaria al proyecto de investigación “Desarrollo de software aplicando una propuesta metodológica diseñada en la Universidad Católica de Pereira”, presentado por el Grupo de Investigación Entre Ciencia e Ingeniería, de la Facultad de Ciencias Básicas e Ingeniería, de la Universidad Católica de Pereira, teniendo en cuenta que el modelo comprende una serie de fases y actividades alrededor del desarrollo y la gestión: requerimientos, arquitectura, construcción, pruebas, implantación, planificación y estimación, SQA, configuración de versiones, reporte de incidencias y documentación.

Este trabajo es un complemento a la metodología de desarrollo, que como resultado generó una solución que permite la gestión de paz y salvos en instituciones educativas por medio del seguimiento detallado de la metodología propuesta por el grupo de investigación, que tiene como fin atribuir la calidad del producto ya que se desarrolló y gestionó de manera disciplinada.

Tal motivo conlleva a realizar un esfuerzo conjunto que permita compartir un conocimiento específico que tenga validez en el ámbito pragmático, pues se ha de considerar que la sistematización de procesos en relación con los requerimientos de software, tiene una vinculación con el mantenimiento del software en cuanto a que los requerimientos funcionales y de rendimiento establecidos, tengan concordancia con las características implícitas que se espera de todo software desarrollado profesionalmente, en procura de que el mantenimiento sea realmente un desarrollo evolutivo donde se preserve el valor del software sobre el tiempo (April, Huffman, Abran, & Dumke, 2004).

Cabe mencionar aquí el estudio de caracterización de la calidad del software (Merchán & Urrea, 2007) el cual tomó como población total 256 Mipymes registradas en FEDESOFTE (Federación Colombiana de Industriales del Software y Tecnologías Relacionadas) de las cuales 153 fueron encuestadas, de éstas, el 32% fueron ubicadas en el Eje Cafetero. Al indagar en lo relacionado con el desarrollo y con la gestión de sus proyectos, se evidencia en este estudio que el proceso en la mayoría de estas pequeñas

organizaciones es realizado por la intuición, las herramientas del momento y las consideraciones personales de quien tenga a cargo un proyecto.

Por la misma línea, en marzo de 2015, según CERTICAMARA S.A. (Sociedad Cameral de Certificación Digital), la certificación CMMI, la habían obtenido 56 empresas en Colombia, de ese total 44 estaban en nivel 3 y 12 en nivel 5, el más alto. Esta certificación es un modelo que se enfoca en evaluar la gestión y el rendimiento de las empresas y que permite asegurar que las compañías tengan costos predecibles, cumplan los cronogramas establecidos y sus negocios sean de cierta calidad.

Según datos de FITI - iniciativa del Ministerio TIC a través del Plan Vive Digital que busca fortalecer la industria de tecnologías de la información mediante acciones específicas haciendo énfasis en las áreas de Red de Valor donde se identifican necesidades de transformación para generar impacto y aumentar la competitividad de una empresa que a su vez permita fortalecer la industria TI nacional -en septiembre de 2015, el país dio un salto hasta las 84 empresas en total, 64 en el nivel 3 y 15 en el máximo nivel, convirtiéndose en uno de los líderes mundiales en cantidad de empresas certificadas. Por su parte, según IT Mark, certificación en métodos técnicos y de negocio, enfocado hacia la mejora de procesos en PYMEs del sector de tecnologías de información que trabaja en componentes tales como la gestión de negocios, ingeniería de software, sistemas y servicios y gestión de seguridad que desarrollan estrategias comerciales, financiera y de mercado, que solo se usa en España, Europa del Este y Colombia. Al 2015 se han certificado 174 empresas de 16 países del mundo, y Colombia lidera el listado de empresas con 80, muchas de ellas apoyadas por el Ministerio de las TIC.

Estas cifras muestran que, si bien es cierto que Colombia no está del todo mal en procesos de certificación para desarrollo de software, no ha conseguido el 100% de empresas certificadas. Cabe resaltar que en los últimos años en la región se han certificado algunas de ella. Por ejemplo con certificación IT MARK: DATACENTER, HEINSOHN, GEMINUS, INDESAP, JAIVANA, OCEANIC, SINGLECLIC, PROSOF Y HEISSON en CMMI nivel 3, estas del número total de empresas en la región, por lo que se hace necesario reforzar el apoyo y facilitar los procesos de certificación, generando mayor eficiencia y calidad para las empresas, lo que se pretende lograr con la guía aquí propuesta,

la cual enfatiza en la forma en que las empresas desarrolladoras de software ejercen sus actividades.

La guía se interesa también por el mantenimiento del producto de software que adquieren los clientes, que bajo todo concepto, se debe hacer énfasis en esta tarea ya que si es bien administrada, no se debe observar como un problema sino más bien como una fuente de ingresos que capitalizaría cualquier empresa desarrolladora, y si a esto se le adhiere un plan o guía de trabajo, se va a reforzar el concepto de calidad, puesto que aumenta la eficiencia con que se realizan las labores y tanto la empresa desarrolladora como sus clientes van a obtener beneficios mutuos, ya que la labor del software es cambiante constantemente (Xiaoping,, Fang, & Yi, 2014).

Los proyectos de desarrollo de software suelen enfrentarse a un ámbito de trabajo muy cambiante, especialmente en cuanto a las especificaciones del producto y a las prioridades de las características solicitadas, se hace un proceso intensivo en comunicación, entre el equipo y con el cliente (Bwch & Kung, 1997). Según el artículo *Software Maintenance Process Model and Contrastive Analysis* (2011) las técnicas y herramientas genéricas para llevar a cabo el mantenimiento del software resultan claramente ineficaces para enfrentar estos desafíos; las metodologías ágiles destacan esta situación pero la resuelven de una manera excesivamente simplista, utilizando roles muy genéricos y confiando en la habilidad de cada uno de los miembros del equipo para que, verbalmente y/o con soportes no informatizados, realicen el mantenimiento continuo (Yongchang & Zhongjing, 2011).

Por todo lo presentado anteriormente, nace la siguiente pregunta de investigación:

¿Cómo pueden las Mipymes y sus departamentos de desarrollo de software de la ciudad de Pereira, mejorar las operaciones, la gestión y la calidad del producto/teniendo como base la metodología Ágil Mantema y la norma ISO/IEC 12207?

4 JUSTIFICACIÓN

La propuesta se sustenta sobre los datos obtenidos en la encuesta y el grupo focal aplicado a los empresarios de la región en el primer momento de este proyecto, en los cuales se pudo evidenciar que pocas empresas de software de Pereira siguen una metodología de mantenimiento de software, y que además, en las investigaciones realizadas por el grupo de investigación Entre Ciencias e Ingeniería de la Universidad Católica de Pereira, tampoco se han obtenido resultados de estudios que revelen la situación del mantenimiento de software por parte de las empresas de la ciudad de Pereira.

Todas estas razones generan la necesidad de crear “MANTELASOFT”, una guía simple de mantenimiento de software, con la cual las empresas puedan evaluar el estado del mantenimiento de sus desarrollos sin que les tome mucho tiempo ejecutarla, y que cada empresa pueda ajustarla a sus necesidades. Esta propuesta tiene como base una metodología el proceso Ágil Mantema, la cual a su vez se apoya en otra metodología llamada MANTEMA y buenas prácticas de la norma ISO/IEC 12207.

Por tanto, se reconocen los aportes de estas dos herramientas importantes, sin embargo, MANTELASOFT nace como una propuesta diferente que, aunque retoma algunos principios de MANTEMA e ISO/IEC, se plantea como una guía nueva y original que obedece a las necesidades planteadas por las empresas abordadas en esta investigación.

MANTELASOFT, es una guía pensada para responder a las necesidades propias y mantener un control detallado del mantenimiento, con características especiales que lo hacen factible para las empresas de desarrollo de software. Con esta propuesta se pretende lograr que las empresas o departamentos de desarrollo incorporen procesos de calidad en la fase de mantenimiento para el desarrollo de sus productos, con el fin de garantizar el funcionamiento del software al cliente.

El fundamento de este proyecto es proporcionar una guía a las empresas, especialmente a las Mipymes y sus departamentos de desarrollo de software de la ciudad de Pereira para su mejoramiento y perfeccionamiento, razón por la que se plantea “MANTELASOFT”, proceso para la evolución y mantenimiento de software diseñado con características especiales basadas en la información recolectada en las encuestas aplicadas

durante la realización del estudio de buenas prácticas, procedimientos y técnicas que utilizan estas empresas para hacer el mantenimiento de software.

“Mantelasoft” tiene conceptos y procedimientos de algunas de las guías y marcos de trabajo reconocidos del sector de la ingeniería de software, como son la Swebok V3, la Métrica V3, el libro *Software Evolution and Maintenance* (Polo Usaola, 2000) y el documento de COMPETISOFT, reconociéndolos como unos antecedentes positivos y útiles en la obtención de los resultados buscados.

Este trabajo pretende lograr que las Mipymes y sus departamentos de desarrollo de software obtengan una optimización de sus costos, de la producción y de la calidad del producto, a partir de mantener en excelente funcionamiento el software en el momento que se necesite; dándoles un carácter más competitivo desde la región hacia el resto del país y el mundo.

La guía presenta un proceso para mantenimiento de software que integra todas las actividades relacionadas con este proceso. Su objetivo es convertir el mantenimiento de software en un proceso controlable y mensurable mediante la identificación de los elementos que intervienen en él.

Su contenido es una síntesis del proceso Ágil Mantema, los procesos de apoyo planteados por la norma ISO/IEC 12207 y su sustento es la encuesta de caracterización de las prácticas del mantenimiento de software en departamentos de sistemas de Mipymes desarrolladoras de software de la ciudad de Pereira.

El presente trabajo se muestra importante para las empresas desarrolladoras de software de la región, pues hasta el momento de finalización de este proyecto, las publicaciones encontradas en el tema de evolución y mantenimiento de software no se ajustan del todo al contexto de la región, considerando que trece (13) empresas que hicieron parte del estudio declararon, específicamente en el grupo focal, que al intentar aplicar alguna de las guías existentes, el proceso se torna costoso, confuso y difícil de aplicar.

Este proyecto busca complementar la línea de investigación *"Buenas prácticas en la gestión para mejorar la calidad de los sistemas de información"* del Grupo de Investigación Entre Ciencia e Ingeniería de la Universidad Católica de Pereira, que fue creado en el año 2005, teniendo como objetivo general incentivar la cultura científica para

generar y realizar proyectos relacionados con las tecnologías de la información y las telecomunicaciones específicamente en el tema “Mantenimiento de Software”, teniendo en cuenta que, a la fecha, este tema no ha sido abordado por ninguno de los investigadores, aunque sí se han trabajado contenidos como calidad y requerimientos de software, los cuales son publicados en la revista Entre Ciencia e Ingeniería.

Finalmente, a las empresas que colaboraron con la investigación, se les mostrará el resultado obtenido para que, de primera mano, puedan tenerlo y utilizarlo en sus empresas. Aquellas que no participaron, pero que son parte de la población objetivo, se enviará un correo compartiendo “Mantelasoft” para que lo utilicen si lo consideran necesario.

5 REFERENTE TEÓRICO

5.1 ESTADO ACTUAL DEL MANTENIMIENTO EN PEQUEÑAS ORGANIZACIONES Y MIPYMES EN LA CIUDAD DE PEREIRA

Actualmente en la ciudad de Pereira, se están consolidando empresas de Desarrollo de Software reconocidas a nivel nacional y mundial. Sin embargo, ha sido difícil encontrar información de estudios o investigaciones acerca del Mantenimiento de Software en las empresas de la ciudad. Es por esta razón que el primer objetivo de esta investigación fue “Caracterizar las prácticas del mantenimiento de software, en departamentos de sistemas o MIPYMES desarrolladoras de software de la ciudad de Pereira, mediante la aplicación de una encuesta. Y de esta manera contribuir con un estudio que pueda servir de insumo para futuras investigaciones e incluso para las empresas de la ciudad y la región.

5.2 CONCEPTOS

5.2.1 Ingeniería de software

En SWEBOK se define la Ingeniería de Software como la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software (IEEE Computer Society, 2014, pág. xxxi). Esta disciplina trasciende la actividad de programación, que es el pilar fundamental a la hora de crear una aplicación, el ingeniero de software se encarga de toda la gestión del proyecto para que éste se pueda desarrollar en un plazo determinado y con el presupuesto previsto. Por lo tanto, la Ingeniería de Software incluye el análisis previo de la situación, el diseño del proyecto, el desarrollo del software, las pruebas necesarias para confirmar su correcto funcionamiento y la implementación del sistema

5.2.2 Proceso

La definición que brinda SWEBOK del Proceso de Software es una serie de actividades y tareas interrelacionadas que transforman productos de trabajo de entradas en salidas (IEEE Computer Society, 2014, págs. 1-2). También se mencionan que los Procesos de Mantenimiento proveen actividades necesarias y entradas/salidas detalladas a esas actividades (IEEE Computer Society, 2014, págs. 5-7). Por otra parte, Tripathy y Naik mencionan en su libro “Software Evolution and Maintenance” que el proceso de producción de software comprende una serie de actividades comenzando desde la concepción hasta el retiro. Hay muchos procesos de software que se diferencian principalmente en su clasificación de fases y actividades (Tripathy & Naik, 2015, pág. 83).

El Área de Conocimiento Modelos y Métodos de la ingeniería del software incluye ambas, métodos de la ingeniería del software y modelos de la ingeniería del software. La subárea de Modelos de la ingeniería del software utiliza la misma estructura que la guía en sí misma, con un tema por cada una de las otras nueve Áreas de Conocimiento de la ingeniería del software. Se añade un tema adicional: cuestiones varias sobre modelos como técnicas de integración de herramientas que son potencialmente aplicables a todo tipo de herramientas, como son: Requerimientos de las herramientas de sw, herramientas de diseño sw, de construcción sw, de pruebas de sw, de mantenimiento de sw, de dirección de configuración de sw, de dirección en la ingeniería de sw, de proceso de ingeniería de sw, de calidad de sw y cuestiones de herramientas compuestas. La subárea de Métodos de la ingeniería del software se divide en tres subsecciones: métodos heurísticos que tratan aproximaciones informales, métodos formales basados en aproximaciones matemáticas y métodos de prototipado tratando varias formas de prototipados.

Desarrollar un software es construirlo a partir de los requerimientos y necesidades que tiene el cliente, pero para asegurar que el software siga satisfaciendo las exigencias del usuario es necesario llevar a cabo el mantenimiento, aplicable al software desarrollado y realizado para corregir defectos, mejorar el diseño, el interfaz con otros sistemas, adaptar programas con diferente hardware y diferente software, emigrar software y retirar el software (IEEE Computer Society, 2014, pág. 26).

5.2.3 Mantenimiento de software

El Mantenimiento de Software es la totalidad de actividades necesarias para proveer un soporte rentable de un sistema de software. Las actividades se realizan durante la etapa preentrega y post-entrega (IEEE, 2006, pág. 4). Es una de las fases en el ciclo de vida de desarrollo de sistemas (SDLC), que se aplica al desarrollo de software. La fase de mantenimiento es la fase que viene después del despliegue o implementación del software en el campo. El mantenimiento es realmente un desarrollo evolutivo y que las decisiones de mantenimiento son ayudadas por entender lo que sucede a los sistemas y al software con el tiempo (Herrera Caldas, 2015).

Muchas actividades de mantenimiento son similares a aquellas de desarrollo de software. Los mantenedores realizan el análisis, el diseño, la codificación, las pruebas y la documentación. Ellos deben rastrear exigencias en sus actividades tal cual, hechas en el desarrollo, y la documentación de actualización como el cambio de líneas de fondo. Hay un número de subprocessos del mantenimiento de software, como son (IEEE Computer Society, 2014, pág. 98)

- Transición: una secuencia controlada y coordinada de actividades durante las cuales el software es transferido cada vez más del desarrollador al mantenedor.
- La Aceptación/Rechazo de Petición de Modificación: el trabajo de petición de modificación sobre un cierto tamaño/esfuerzo/complejidad puede ser rechazado por mantenedores y desviado a un desarrollador.
- La petición de Modificación y la Mesa de Ayuda de Informe de Problema: una función de apoyo de usuario final que provoca la evaluación, la ordenación, y de presupuesto de solicitud de modificación.
- El Análisis de Impacto.
- El Apoyo de Software: ayuda y aconseja a usuarios que solicitan información (por ejemplo, reglas de gestión, validación, datos que quieren decir y ad hoc solicita/hace un informe).
- Los Acuerdos de Nivel de Servicio (SLAs) y los contratos de mantenimiento especializados (específicos de dominio) que son responsabilidad de los mantenedores.

5.2.4 Tipos de Mantenimiento

Los autores consultados como SWEBOK (IEEE Computer Society, 2014, págs. 4-5), IEEE 14764 (IEEE, 2006), IEEE 1219 (IEEE, 1998, pág. 4) y otros (Tripathy & Naik, 2015, pág. 5) (Pino, Ruiz, & Salas, 2008, pág. 12), concuerdan con los siguientes tipos de Mantenimiento de Software:

Mantenimiento no planificable: Es el mantenimiento de software que no puede ser programado debido a su carácter de emergencia, en este tipo de mantenimiento se encuentra el Correctivo urgente.

Mantenimiento planificable: Es el mantenimiento de software que permite ser programado debido a su carácter de no emergencia, en este tipo de mantenimiento se encuentra el Correctivo no urgente, Perfectivo, Preventivo y Adaptativo.

Correctivo: Modificación reactiva de un producto de software después de su entrega para corregir problemas encontrados. Cabe mencionar el mantenimiento de emergencia el cual es una modificación para mantener un sistema operando mientras se realiza el mantenimiento correctivo. Es aquella situación donde al presentarse un error en el producto, bloquea la aplicación o el proceso de funcionamiento de la empresa y debe ser resuelta con brevedad; o se produce el fallo y no es crítico para el funcionamiento, por lo que da un tiempo más para su solución. Es mantenimiento Correctivo urgente, cuando un error bloquea la aplicación o el proceso de funcionamiento de la empresa, y se debe resolver rápidamente. Y es mantenimiento Correctivo no urgente, cuando un error no es crítico pero tal vez impida el funcionamiento de la aplicación o de la empresa en un período de tiempo corto.

Adaptativo: Modificación de un producto de software realizado después de la entrega para mantenerlo usable si hay cambios en el ambiente. Por ejemplo, el sistema operativo debe ser actualizado o el cambio del sistema de gestión de base de datos.

Perfectivo: Modificación de un producto de software después de su entrega para proveer mejoras para los usuarios, mejorar la documentación del programa y recodificar para optimizar el rendimiento del software, mantenibilidad u otros atributos del software. Se encarga de adicionar nuevas características o funcionalidades, generalmente solicitadas por el cliente.

Preventivo: Modificación de un producto de software después de su entrega para detectar y corregir fallos latentes antes de que éstos lleguen a ocasionar fallas operativas, buscando que en un futuro el esfuerzo del mantenimiento sea menor.

5.2.5 Evolución del software

El término fue definido a finales de 1960 por Lehman y sus colaboradores (Lehman, Ramil, Wernick, Perry, & Tursky, 1997). Basado en estudios empíricos, formularon varias observaciones y ellos las introdujeron como las “Leyes de la Evolución. Las leyes han evolucionado de tres en 1974 a ocho en 1997.

La evolución de sistemas de software significa crear nuevos diseños, pero relacionados con los ya existentes. Los objetivos son dar soporte a nuevas funcionalidades, hacer que los sistemas funcionen mejor y hacer que los sistemas corran en diferentes sistemas operativos (Tripathy & Naik, 2015, pág. 3). El mantenimiento es un desarrollo evolutivo y las decisiones de mantenimiento son ayudadas por la comprensión de lo que sucede en el software sobre el tiempo. Un software de gran tamaño nunca es completado y continúa evolucionando; y a medida que evoluciona, crece más su complejidad a menos que se tomen varias acciones para reducirla. (IEEE Computer Society, 2014, págs. 5-3). *“La evolución es la fase donde es posible hacer cambios significativos a la arquitectura y la funcionalidad del software. Durante el servicio, los únicos cambios que se realizan son relativamente pequeños”*. (Sommerville, 2011, pág. 236).

Se es consciente de que el software se encuentra en constante evolución, es por ello por lo que hay cientos de lenguajes de programación en desarrollo y librerías que actualizan los ya existentes, es precisamente por estos cambios que el software a desarrollar requiere de adaptabilidad y por ende de un estudio constante (Mitchell, 2004).

5.2.6 Proceso Ágil

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término ágil, aplicado al desarrollo de software. Esto pretende ofrecer una alternativa que permita a los

equipos desarrolladores, desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto (Bohem, 2003).

Es de aquí en adelante donde surgen diferentes metodologías que dan respuesta a modelos ágiles ya existentes.

El desarrollo de software ágil es un concepto usado en el desarrollo de software para describir las metodologías de desarrollo incrementales (Cohen, Lindvall, & Costa, 2003), con el fin de fortalecer los procesos de comunicación en relación con la documentación y generando que equipos de trabajo multidisciplinar tengan mayor colaboración.

5.2.7 Scrum

Según la definición de Gutiérrez Joaquín, (Gutierrez, 2007), define Scrum como un marco para gestión de proyectos, que se ha utilizado con éxito en los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se resumen en:

El desarrollo de software se realiza mediante iteraciones, denominadas Sprint, con una duración de 30 días. El resultado de cada Sprint es un incremento ejecutable que se muestra al cliente. La segunda son las reuniones a lo largo del proyecto, del grupo de desarrolladores para integrar, coordinar y hacer seguimiento, se reunión diaria de solo 15 minutos, regularmente a primeras horas de la mañana.

Este proceso genera trabajo colaborativo, mayor desempeño, innovación, competitividad y mejores resultados de un proyecto, este proceso hace entregas parciales y periódicas del proyecto, esto desencadena obtención de resultados prontamente

Scrum es una metodología que permite que en proyectos que no se este dando cumplimiento a las necesidades del cliente, resolverlas de forma rápida, evitar incremento en costos, tiempo, cuidando disminuir la calidad del producto, mejorar la reacción frente a la competencia, incrementando la motivación de los equipos de trabajo o cuando se requiere de que se desarrolle un producto con un proceso especializado.

5.2.8 Calidad del software

Dentro del contexto de Ingeniería de Software, la calidad del software propuesta por la organización internacional de estándares (ISO/IEC DEC 9126 y la 25000), es la totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas. Se puede decir que el software tiene calidad si cumple o excede las expectativas del usuario en cuanto a:

1. Funcionalidad (que sirva un propósito),
2. Ejecución (que sea práctico),
3. Confiabilidad (que haga lo que debe),
4. Disponibilidad (que funcione bajo cualquier circunstancia) y
5. Apoyo, a un costo menor o igual al que el usuario está dispuesto a pagar.

La calidad de software se refiere a: “Los factores de un producto de software que contribuyen a la satisfacción completa y total de las necesidades de un usuario u organización”.

5.2.9 Mantenibilidad

Según la norma ISO 25010, Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas. Esta característica se subdivide a su vez en las siguientes sub-características:

- Modularidad. Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- Reusabilidad. Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.
- Analizabilidad. Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.

- Capacidad para ser modificado. Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- Capacidad para ser probado. Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

5.2.10 Modularidad

La modularidad es la manifestación más común de la división de problemas. El software se divide en componentes con nombres distintos y abordables por separado, en ocasiones llamados módulos, que se integran para satisfacer los requerimientos del problema.

El software monolítico (un programa grande compuesto de un solo módulo) no es fácil de entender para un ingeniero de software. El número de trayectorias de control, alcance de referencia, número de variables y complejidad general haría que comprenderlo fuera casi imposible. En función de las circunstancias, el diseño debe descomponerse en muchos módulos con la esperanza de que sea más fácil entenderlos y, en consecuencia, reducir el costo requerido para elaborar el software (Pressman R. S., 2010).

5.2.11 Modelo de calidad de proceso

Es un conjunto de normas para calidad enfocada al mundo del Software. Estas se aplican a los diferentes procesos que hay que llevar a cabo para lograr producir software con calidad, es muy importante mencionar que igual que las normas ISO 90003, este modelo dice qué hay que hacer, y no cómo hay que hacerlo.

El modelo de calidad permite:

- Describir los componentes del modelo y sus relaciones.
- Comprender las áreas de proceso.
- Localizar información relevante en el modelo.
- Aplicar los conocimientos a su entorno de trabajo y en un equipo de evaluación de componentes y sus relaciones de un modelo

5.2.12 Competisoft

Es un proyecto financiado por CYTED, programa internacional de cooperación científica y tecnológica multilateral, de ámbito iberoamericano que tiene como propósito incrementar el nivel de competitividad de las Pymes iberoamericanas productoras de software mediante la creación y difusión de un marco metodológico común que, ajustado a sus necesidades específicas, llegue a ser la base sobre la que se pueda establecer un mecanismo de evaluación y certificación de la industria del software reconocido en toda Iberoamérica (Pino, Ruiz, & Salas, 2008).

El proyecto buscó recoger el conocimiento de más de 100 investigadores de países como España, México, Brasil, Argentina, Uruguay, Colombia, Ecuador, Costa Rica, Chile, Perú, entre otros. En el proyecto se plantearon los siguientes objetivos específicos:

- Generar un marco metodológico común iberoamericano.
- Difundir la cultura de procesos mediante la formación de investigadores, docentes y profesionales.
- Incidir en los diferentes organismos de normalización y certificación de los países iberoamericanos, para que asuman que los principios metodológicos, objeto de este proyecto puedan ser la base para establecer un mecanismo común y mutuamente reconocido de evaluación y certificación de la industria del software iberoamericana.

5.2.13 Índice de Madurez del Software (IMS)

Brinda una visión general sobre la estabilidad del producto de software, con base en los cambios que se puedan presentar desde la liberación de este. (IEEE, 1988)

Para calcularlo se puede utilizar la siguiente fórmula:

$$IMS = \frac{[Mt - (Fa + Fm + Fe)]}{Mt}$$

Donde:

Mt = número de módulos en la versión actual.

F_m = número de módulos en la versión actual que han sido modificados.

F_a = número de módulos en la versión actual que han sido añadidos.

F_e = número de módulos de la versión anterior que se han eliminado en la versión actual

A medida que el resultado de la fórmula anterior se aproxima a 1, el software tiende a estabilizarse y, por ende, requerirá menor esfuerzo para su mantenimiento.

5.2.14 Leyes de Lehman

Basado en estudios empíricos, Lehman y sus colaboradores formularon algunas observaciones y las introdujeron como leyes de la evolución. (Tripathy & Naik, 2015)

Las "leyes" mismas han "evolucionado" de tres en 1974 a ocho en 1997. Esas leyes son el resultado de estudios sobre la evolución de los sistemas de software de código cerrado o de propiedad privada a gran escala. Las ocho leyes se explican brevemente de la siguiente manera:

- Cambio continuo. A menos que un sistema se modifique continuamente para satisfacer las necesidades emergentes de los usuarios, el sistema se vuelve cada vez menos útil.
- Complejidad creciente A menos que se realice un trabajo adicional para reducir explícitamente la complejidad de un sistema, el sistema se volverá cada vez más complejo debido a los cambios relacionados con el mantenimiento.
- Autorregulación. El proceso de evolución se autorregula en el sentido de que las medidas de los productos y procesos, que se producen durante la evolución, se aproximan a las distribuciones normales.
- Conservación de la estabilidad organizacional. Es la cantidad promedio de esfuerzo adicional necesaria para producir una nueva versión es casi la misma.
- Conservación de la familiaridad. Como un sistema evoluciona todo tipo de personal, es decir, los desarrolladores y usuarios, por ejemplo, deben obtener

un nivel deseado de comprensión del contenido y el comportamiento del sistema para lograr una evolución satisfactoria. Un gran crecimiento incremental en una versión reduce esa comprensión. Por lo tanto, el crecimiento incremental promedio en un sistema en evolución sigue siendo casi el mismo.

- Crecimiento continuo. A medida que pasa el tiempo, el contenido funcional de un sistema se incrementa continuamente para satisfacer las necesidades del usuario.
- Calidad decreciente. A menos que el diseño de un sistema se ajuste con precisión y se adapte a los nuevos entornos operativos, las cualidades del sistema se percibirán como decrecientes a lo largo de la vida útil del sistema.
- Sistema de retroalimentación. El proceso de evolución del sistema implica retroalimentación multinivel, multiagente y multinivel entre diferentes tipos de actividades. Los desarrolladores deben reconocer esas interacciones complejas a fin de evolucionar continuamente un sistema existente para ofrecer más funcionalidades y niveles más altos de cualidades.

5.2.15 Norma ISO/IEC 12207

La Norma ISO/IEC 12207 comprende todo lo que al ciclo de vida del software se refiere, iniciando desde la idea del software hasta la retirada. Consta de procesos para la adquisición, suministro y servicios de software, estableciendo modelos para su control y mantenimiento.

Su objetivo es definir una estructura común para que todos los involucrados en el desarrollo del software usen un lenguaje en común.

Procesos de Soporte o Apoyo

“Un proceso de apoyo es el que apoya a otro proceso como parte esencial del mismo, son un propósito bien definido y contribuye al éxito y calidad del proyecto de software” (ISO/IEC, 2008)

Los procesos de apoyo son:

- Proceso de documentación: Define las actividades para el registro de la información producida por un proceso del ciclo de vida
- Proceso de gestión de la configuración: Define las actividades de la gestión de la configuración.
- Proceso de gestión de aseguramiento de la calidad: Define las actividades para asegurar, de una manera objetiva, que los productos de software y los procesos son conformes a sus requerimientos especificados y se ajustan a sus planes establecidos. Revisión conjunta, auditoría, verificación y validación pueden ser utilizados como técnicas de Aseguramiento de la Calidad.
- Proceso de verificación: Define las actividades para verificar hasta un nivel de detalle dependiente del proyecto de software, los productos software.
- Proceso de validación: Define las actividades para validar los productos software del producto software
- Proceso de revisión conjunta: Define las actividades para evaluar el estado y productos de una actividad. Este proceso puede ser empleado por cualquiera de las dos partes, donde una de las partes revisa a la otra parte, de manera conjunta.
- Proceso de auditoría: Define las actividades para determinar la conformidad con los requerimientos, planes y contrato.
- Proceso de solución de problemas: Define las actividades para analizar y eliminar los problemas que sean descubiertos durante la ejecución del proceso de desarrollo, operación, mantenimiento y otros procesos.

5.2.16 Ágil Mantema

Para el presente proyecto, se tuvo referencia del Informe Técnico No: IT 23 de Competisoft, llamado Ágil Mantema del 28 de febrero de 2008.

Ágil MANTEMA está creado a partir de la agilización de MANTEMA a través de la aplicación de Scrum.

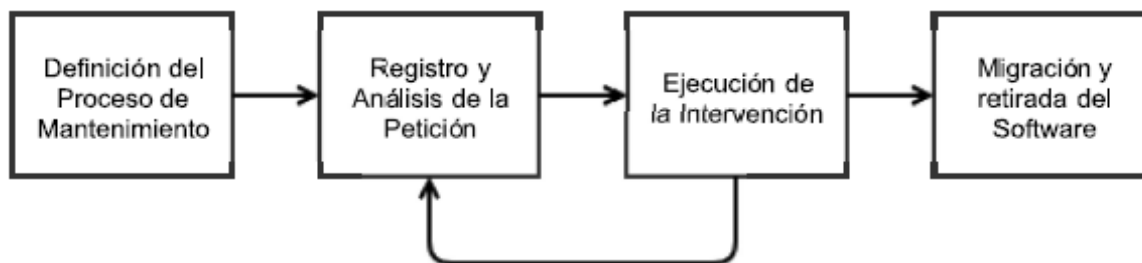
La metodología MANTEMA muestra la visión del proceso de mantenimiento desde el mayor nivel de abstracción, en el que probablemente no interesa el contenido de las instrucciones ni los campos de los archivos, aunque sí las mejores técnicas para entenderlos y modificarlos. (Pino, Ruiz, & Salas, 2008)

Es por esto que MANTEMA describe qué se debe realizar, cuándo, cómo y por quién, de tal manera que cada intervención de mantenimiento que se lleve a cabo sea conforme a un proceso de mantenimiento predefinido.

Ágil MANTEMA está enfocada a pequeñas organizaciones y pretende definir un proceso de mantenimiento, detallando qué debe realizarse, cuándo, cómo y por quién, es decir, busca guiar paso a paso el proceso de mantenimiento del software en este tipo de organizaciones.

La estructura general de Ágil MANTEMA está pensada en ayudar a las pequeñas organizaciones que desean disponer de una guía metodológica para llevar a cabo el proceso de mantenimiento de software. Dicha estructura, presentada en la Figura 4, se basa en la estructura de MANTEMA y en las indicaciones del estándar ISO/IEC 12207.

Figura 4. Estructura general de Agil_MANTEMA



Fuente: (Pino, Ruiz, & Salas, 2008)

La estructura general, o modelo de proceso, presentada en la figura anterior se complementa con los siguientes elementos, que también se incluyen en Agil_MANTEMA:

- Niveles de servicio extraída de Métrica V3 y adaptada a esta metodología.
- Nivel de capacidad del proceso basado en la norma ISO/IEC 15504:2006 [4], y adaptada también a esta metodología.

La estructura general de Agil_MANTEMA se complementa con la utilización del concepto de nivel de servicio, basado con adaptaciones en el mismo concepto de Métrica V3.

Agil_MANTEMA define tres niveles de servicio: el nivel básico abarca el mantenimiento correctivo urgente; el nivel intermedio añade al anterior el correctivo no urgente y el perfectivo, y el nivel avanzado abarca todos los tipos de mantenimiento, incorporando a los del nivel anterior los mantenimientos adaptativo y preventivo.

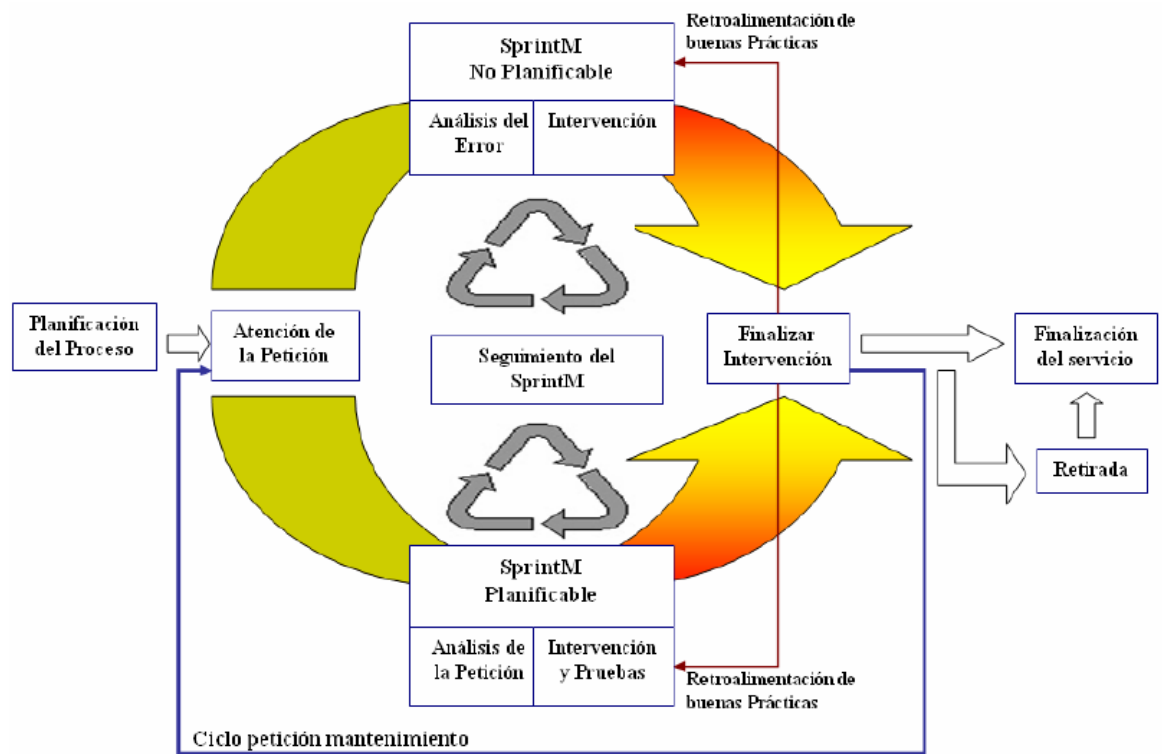
Figura 5. Niveles de servicios definidos en Agil_MANTEMA

	Nivel Básico	Nivel Intermedio	Nivel Avanzado
Tipos de Mantenimiento	- Correctivo Urgente	- Correctivo Urgente - Correctivo No Urgente - Perfectivo	- Correctivo Urgente - Correctivo No Urgente - Perfectivo - Adaptativo - Preventivo
Interfaz fundamental	- Soporte al cliente - Gestión de resolución de problemas	- Soporte al cliente - Gestión de resolución de problemas - Gestión de la Configuración - Aseguramiento de la Calidad	- Soporte al cliente - Gestión de resolución de problemas - Gestión de la Configuración - Aseguramiento de la Calidad - Gestión de cambio de requisitos - Gestión de proyectos

Fuente: (Pino, Ruiz, & Salas, 2008)

Con todas las descripciones anteriores, se genera el proceso de mantenimiento Agil_Mantema el cual se puede observar en la Figura a continuación. Cada una de estas actividades tiene definida sus tareas específicas que se deben realizar, junto con sus entradas, salidas, técnicas, roles y niveles de servicio.

Figura 6. Proceso de Agil_MANTEMA



Fuente: (Pino, Ruiz, & Salas, 2008)

Como se puede ver en el proceso de Agil Mantema, comienza con la Planificación del Proceso donde se asignan los responsables, se adquiere el conocimiento de la aplicación a mantener, se preparan los entornos de pruebas y se definen los procedimientos para las peticiones. Luego, se realiza la Atención de la Petición, allí se recibe la petición de modificación y se decide qué tipo de mantenimiento es.

Después se decide si se atiende la petición como urgente (no planificable) o no urgente (planificable). Si es urgente, se analizan las causas, se realizan las correcciones necesarias y se realizan las pruebas de la corrección. si no es urgente, se analiza y elige la opción mas adecuada de la solución, se realiza ejecuta la intervención y se realizan todas las pruebas necesarias como unitarias, funcionales, regresión.

En el seguimiento se realizan reuniones habituales de avance, problemas que se han tenido, dudas, etc, y se revisan los cambios realizados y comprobar si han quedado bien corregidos.

Y por último se Finaliza la intervención donde se verifica y valida la corrección realizada, se hace el paso a producción, se realizan manuales si es necesario y se informa al cliente de lo realizado.

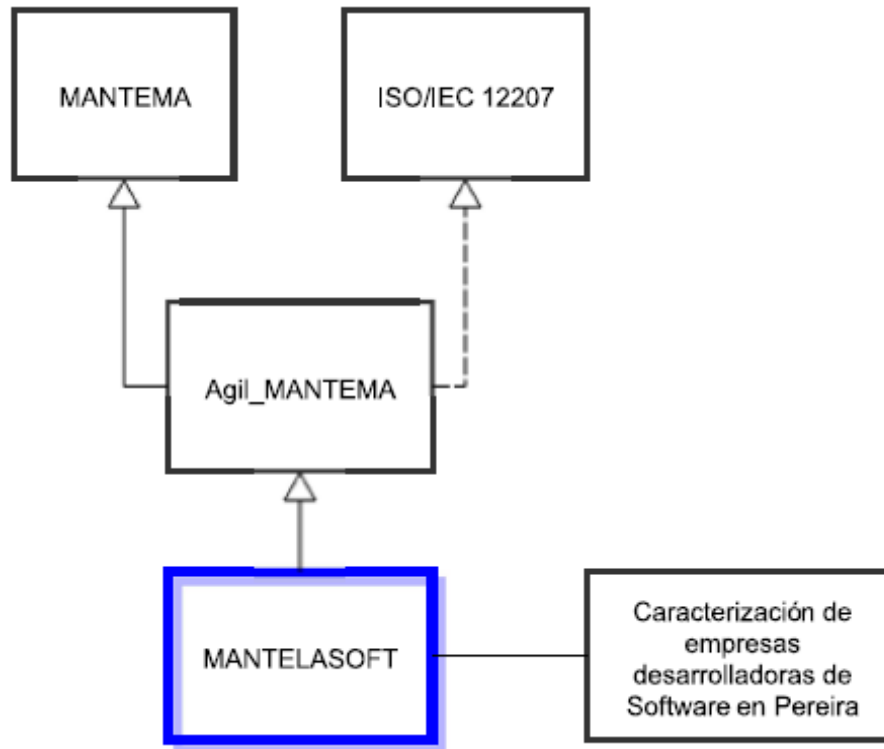
5.2.17 Relación y Comparación entre Agil_MANTEMA y MANTELASOFT

MANTELASOFT nace de la necesidad de dar un bosquejo general del mantenimiento de software a las empresas, como también que, a través de una guía, se pueda evaluar el mantenimiento que las empresas hacen a sus desarrollos, en especial las MiPymes, de forma simple, y dejar de lado procesos, tareas y actividades que pueden tomar mucho tiempo y recursos para su implementación.

MANTELASOFT se basa en Agil_MANTEMA, ya que se enfoca en pequeñas organizaciones, tiene sus procesos bien definidos junto con la descripción de cada una de las actividades que se deben realizar. Además, porque, esta a su vez, se basa en otra metodología para el mantenimiento como MANTEMA e implementa las indicaciones y buenas prácticas de la norma ISO/IEC 12207, como también los procesos de apoyo que se encuentran descritos en la norma anteriormente mencionada. Por otra parte, se tuvieron en cuenta las necesidades que tienen las empresas desarrolladoras de software a través de la caracterización que se realizó, la cual se detalla más adelante.

Es así como MANTELASOFT es una síntesis de las metodologías, normas y necesidades de las empresas para evaluar los proyectos de software en la fase de mantenimiento, y de esta forma los desarrollos de software puedan aumentar su calidad y puedan aplicar una guía de mantenimiento donde no sea muy costoso en tiempo y recursos.

Figura 7. Relación entre MANTELASOFT y Agil_MANTEMA



Fuente: Los autores

Con lo mencionado anteriormente, de Agil_Mantema se adopta las siguientes características:

- La clasificación de los tipos de mantenimiento y la agrupación de estos, razón por la cuál en MANTELASOFT se encuentran tres guías, cada una con acciones específicas para cada uno de los grupos de mantenimiento.
- Entre las más importantes actividades descritas en los procesos “Planificación del Proceso”, “Atención de la Petición”, “Análisis de la Petición”, “Análisis del Error”, “Intervención y Pruebas”, “Finalizar Intervención”, “Retirada”, se encuentran aquellas que tienen que ver con:

“Asignar Responsables”, “Recibir petición de modificación”, “Decidir el tipo de mantenimiento”, “Analizar y elegir la solución”, “Migración y retirada del Software”

- Para cada una de las actividades descritas en MANTELASOFT, también se tomó como base los roles que aplican para cada tarea y los documentos que pueden servir como soporte para la misma.

Por otra parte, para MANTELASOFT no se tuvieron en cuenta las siguientes consideraciones de Agil_Mantema:

- Gestión de la Configuración, no se tuvo en cuenta como proceso según se especifica en Agil_Mantema, ya que en sí mismo es un proceso que involucra un incremento sustancial en tiempo, costos y personal para su implementación.
- No se tuvo en cuenta los niveles de servicio que se proponen para cada tipo de mantenimiento, ya que no todas las empresas de desarrollo tienen este tipo de estructura.

Además de las categorías y actividades de soporte tomadas y sintetizadas en MANTELASOFT, se implementan las siguientes actividades como complemento y que resultan relevantes para este estudio:

- Métricas del Mantenimiento: un conjunto de indicadores que le permitirá a la empresa tener una referencia de los atributos que puede medir, para conocer el estado y evolución que va teniendo el producto a través del tiempo, con números que lo respalde.
- Estimación de Costos de Mantenimiento: Esta categoría surgió de los [resultados de la encuesta de caracterización](#) realizada a las empresas en esta investigación, pues allí se evidencia que la gran mayoría de las empresas encuestadas no lo hacen y ésta sería una guía para que puedan tener mejor estimación de sus productos y desarrollos.

- Personal de Mantenimiento: Es una actividad donde se quiere involucrar al equipo de trabajo, conocer sus habilidades, conocimientos y motivaciones, pues el personal es la parte fundamental de toda empresa y se debe velar por su bienestar.
- Evolución y Mejora del Producto: Es una fase de reflexión para la empresa, donde se indica que el mantenimiento no solo es para corregir errores, sino también para mejorar los productos que se tienen.

A continuación, se muestra una comparación entre Agil_Mantema y MANTELASOFT:

Tabla 1. Comparación entre MANTELASOFT y Agil_Mantema

MANTELASOFT	AGIL_MANTEMA
Número de Fases: 8	Número de Fases: 0
Número de Actividades: 19	Número de Actividades: 10
Número de Tareas: 64-75	Número de Tareas: 27
Número de Métricas: 35-40	Número de Métricas: 1
Temas a tratar: Estudio Inicial Planificación del Mantenimiento Atención de la Petición Proceso de Mantenimiento Análisis del Error Intervención y Pruebas Métricas del Mantenimiento* Estimación de Costos* Personal de Mantenimiento* Evolución o Mejora del Producto* Actividades Finales del Mantenimiento Finalizar Intervención Retirada	Temas a tratar: Planificación del Proceso Atención de la Petición Análisis del Error Análisis de la Petición Intervención y Pruebas Seguimiento del Sprint Finalizar Intervención Retirada Finalización del Servicio
Proceso menos controlado, con pocos principios	Proceso menos controlado, con pocos principios
Tiene un punto de equilibrio entre la No-	Tiene un punto de equilibrio entre la No-

documentación y Demasiada-documentación	documentación y Demasiada-documentación
Evita la burocracia y brinda resultados	Evita la burocracia y brinda resultados
Brinda cambios y resultados continuos	Brinda cambios y resultados continuos
Provee una herramienta para evaluar el estado del mantenimiento	N/A

Fuente: Los autores

5.3 NORMATIVIDAD

5.3.1 ISO/IEEE 14764 - Software Life Cycle Processes - Maintenance (IEEE, 2006)

El estándar ISO/IEEE 14764, Software Life Cycle Processes - Maintenance (2006) presenta una guía donde se explica el proceso de aplicar el mantenimiento en el ciclo de vida del software y establece definiciones para los diferentes tipos de mantenimiento. En su contenido propone un plan como parte de la estrategia de mantenimiento, donde se manifiesta la necesidad de hacer el mantenimiento, contiene documentación necesaria para identificar quién efectúa ese trabajo, cómo se hace y responsabilidades de todos los involucrados. Además, se incluyen los recursos que hay disponibles, dónde se hace y cuándo comienza. Cabe mencionar que el capítulo de mantenimiento de SWEBOK, se basa en esta norma. Es importante contar con estos atributos de calidad porque nos permite tener conceptos y definiciones diferentes a los otros ya mencionados, pero en especial nos brinda un enfoque diferente del mantenimiento, así que permite visualizar el mantenimiento no solo como una definición sino como un modelo para llevarlo a la práctica dentro del ciclo de vida del software, y no como una etapa aislada del desarrollo de este.

Estos son los aspectos fundamentales en cuanto a la estrategia de mantenimiento que propone el estándar ISO/IEEE 14764. Los elementos que comprenden el proceso de mantenimiento, y que son usados para el proyecto, se detallan a continuación:

- Implementación del proceso.
- Análisis de modificaciones y problemas.
- Implementación de modificaciones.
- Revisión y aceptación del mantenimiento.

- Migración y
- Retiro.

5.3.2 ISO/IEC 29110 - Systems and software engineering — Lifecycle profiles for Very Small Entities (VSEs)

Esta serie de normas pueden ser aplicadas en cualquier fase del desarrollo de software dentro de un ciclo de vida. Esta serie de estándares están destinadas a ser usadas por Pequeñas Empresas que no tienen experiencia en adaptar otras normas y también en cualquier ciclo de vida como cascada, iterativo, incremental, evolutivo o ágil. Desde la perspectiva de Pequeñas Empresas, varios beneficios considerados por el uso de este estándar son buenos procesos internos de desarrollo de software, procesos de gestión de software, mayor confianza y satisfacción del cliente, mayor calidad de productos de sistemas y/o software y disminución del riesgo en el desarrollo. Esos beneficios también podrían ayudar con la competitividad y la cuota de mercado (ISO/IEC, 2016). La norma aporta a la presente investigación, una perspectiva enfocada a las pequeñas empresas y no tan general como las que ya se han mencionado. A pesar de que es para el desarrollo de software, puede servir de guía para saber cómo orientar y estructurar el proceso de mantenimiento que se quiere plantear con base en esta serie de normas y saber la terminología a usar.

6 OBJETIVOS

6.1 Objetivo General

Diseñar un proceso para la evolución y mantenimiento de software, como una adaptación del proceso Ágil Mantema a la norma ISO/IEC 12207:2008 y a la encuesta utilizada para caracterizar las prácticas del mantenimiento de software, para las MIPYMES y los departamentos de desarrollo de software en la ciudad de Pereira.

6.2 Objetivos Específicos

- Caracterizar las prácticas del mantenimiento de software, en departamentos de sistemas o MIPYMES desarrolladoras de software de la ciudad de Pereira, mediante la aplicación de una encuesta.
- Formular el proceso para la evolución y mantenimiento de software llamado “Mantelasoft”, cuya metodología sea producto de una combinación entre el proceso Ágil Mantema, los procesos de apoyo planteados por la norma ISO/IEC 12207:2008 y la encuesta utilizada para caracterizar las prácticas del mantenimiento de software.
- Aplicar el proceso de evolución y mantenimiento de software en dos de los departamentos o MIPYMES desarrolladoras de software.
- Validar la aplicación del proceso para la evolución y mantenimiento de software, por medio de la retroalimentación de los encargados de aplicar este documento, a través del diligenciamiento de la encuesta de evaluación de “Mantelasoft”.

7 ALCANCES Y LIMITACIONES

- Se realizó un estudio de buenas prácticas, procedimientos, técnicas y/o herramientas que utilizan las empresas para hacer el mantenimiento de software. Es un análisis del mantenimiento que cubre la planificación, la estimación de costos, las métricas, el personal y la mejora del producto; que sirvió como insumo para elaborar el proceso para evolución y mantenimiento de software.
- Para el estudio del mantenimiento en las empresas, se contactaron 13 empresas, sin embargo, el estudio se hizo en dos de ellas que manifestaron interés por participar.
- El proceso de mantenimiento que se desarrolló fue conciso y concreto, plasmó lo necesario en el tema de mantenimiento de software, con una metodología producto de una combinación entre el proceso Ágil Mantema, los procesos de apoyo planteados por la norma ISO/IEC 12207:2008 y la encuesta utilizada para caracterizar las prácticas del mantenimiento de software. Dando pie igualmente a próximos proyectos que puedan complementarla o mejorarla.
- Se implementó el proceso de mantenimiento en solo dos empresas o departamentos para tener evidencias de los beneficios y aspectos por mejorar de la misma, aun sabiendo que no es una muestra significativa para validar el proceso de mantenimiento propuesto.

8 METODOLOGÍA

8.1 TIPO DE ESTUDIO

El tipo de estudio realizado es de mejora a los procesos, específicamente en el área de mantenimiento de software, por lo cual el proyecto se enmarca en el esquema SPI (Software Process Improvement) que ha demostrado ser efectivo en proyectos de gran envergadura donde se exige tener un riguroso control de las cosas. Sin embargo, este enfoque no resulta ser el más adecuado para MiPymes, ya que en este tipo de industria el contexto es muy cambiante y necesitan reducir los tiempos de los proyectos y costos relacionados sin alterar la calidad su mismo (Pardo, Hurtado, & Collazos, 2010).

Por otra parte, el ciclo de Mejora Continua PHVA es un modelo liviano que permite abstraer mejor cada una de sus fases, y es éste el que se implementa en las fases de la investigación.

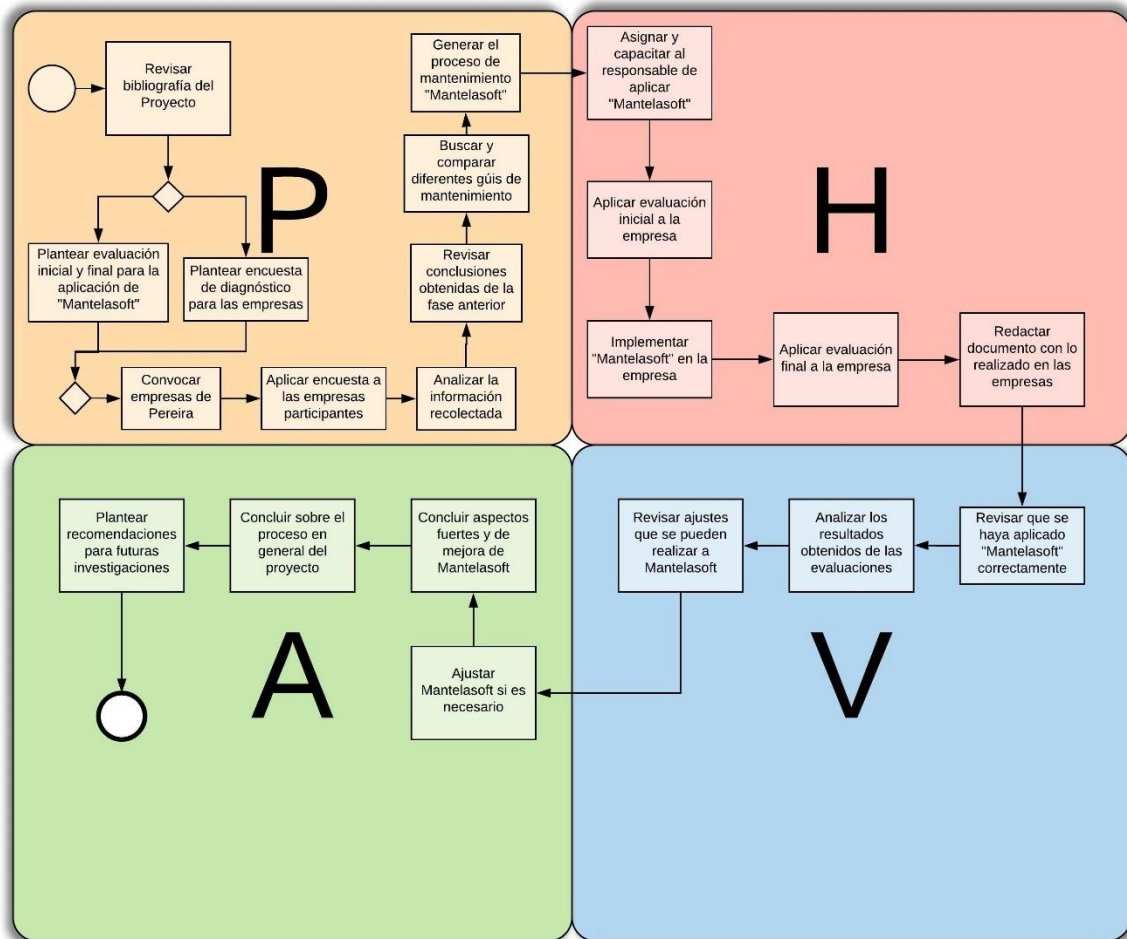
El Ciclo PHVA *“es una herramienta utilizada para la mejora continua que se basa en un proceso de 4 pasos: Planifique, Haga, Verifique, Actúe.”* (Diez Canseco & Mejia Figueroa, s.f), puede brindar el soporte a este proceso con el fin de aplicar ajustes o mejoras en el desarrollo de este. Para llegar a una mejor calidad que satisfaga a los clientes, deben recorrerse constantemente las cuatro etapas, con la calidad como criterio máximo.

Dentro de la fase de Planificación en el ciclo PHVA, se encuentran las fases exploratoria y descriptiva por cuanto aborda una problemática que se puede identificar en el contexto posterior al desarrollo de software y se analizan diferentes aspectos de mantenimiento y necesidades de las empresas a las que acuden los líderes de departamentos y de las MIPYMES desarrolladoras de software, para adecuarse a los procesos evolutivos que por los cambios mismos del software obligan a desarrollar nuevas formas de optimización. Y descriptivo ya que de modo sistemático permite describir las características y el interés que tienen estas empresas sobre el mantenimiento, y posteriormente permite hacer una identificación y análisis de los resultados de las actividades de los desarrolladores en temas de mantenimiento.

Las fases del Ciclo de Mejora PHVA, se ajustan a los objetivos propuestos por los investigadores; donde P (Planear) corresponde al objetivo específico 1 en el cual se obtiene información de las empresas sobre mantenimiento y se planea la elaboración, se diseña el proceso, se ejecuta y se verifica el mismo; H (Hacer) corresponde al objetivo específico 2 en el que se formula el proceso para la evolución y mantenimiento de software; V (Verificar) corresponde al objetivo específico 3, en el que se aplicó “MANTELASOFT” en dos MIPYMES desarrolladoras de software y se validó su aplicación a través del diligenciamiento de la encuesta; y A (Actuar) corresponde a las conclusiones y recomendaciones que se formulen sobre el proceso para la evolución y mantenimiento de software en estas empresas.

En la imagen que se presenta a continuación, se muestra un diagrama donde se modela el proceso que se realizó y cada una de sus actividades, enmarcado en el Ciclo PHVA.

Figura 8. Fases y actividades en el ciclo PHVA



Fuente: Los autores

Las fases y actividades que se llevaron a cabo durante la realización de la presente investigación se detallan a continuación:

Planificar:

- Plan de recolección y análisis de la información

Actividades:

- Elaboración de la encuesta.
- Envío de la encuesta a las empresas interesadas en participar en el proyecto y las registradas en Cámara de Comercio.
- Recolección y tabulación de la información obtenida.

- Análisis y conclusiones de las encuestas.
- Plan de elaboración del proceso de mantenimiento de software

Actividades:

 - Revisar las conclusiones obtenidas de la etapa de recolección de la información.
 - Comparar las diferentes guías que existen de mantenimiento de software.
 - Elaborar el proceso de mantenimiento ya sea nuevo o un complemento a los ya existentes
- Plan de aplicación de la guía en las empresas piloto

Actividades:

 - Definir los atributos que se van a medir antes y después de la aplicación del proceso de mantenimiento.
 - Elaborar evaluación para conocer el estado actual de la empresa y el estado posterior a la aplicación del proceso de mantenimiento.
 - Definir el cronograma y los tiempos para la aplicación del proceso.

Hacer

- Implementación de los planes de mejora
 - Asignación y capacitación del responsable
 - Aplicación de la evaluación antes y después al proyecto piloto de la empresa.
 - Socializar el cronograma de entregas a los responsables del proyecto.
- Implementar el proceso de mantenimiento a un proyecto de software en dos empresas de la ciudad de Pereira.
- Revisar que se haya aplicado el proceso de mantenimiento en su totalidad o al menos las partes que se requieran en el proyecto.
- Redactar un documento donde quede constancia de todo el proceso realizado, los resultados de la evaluación inicial, lo que se presentó durante la aplicación de “Mantelasoft” y la evaluación final.

Verificar

- Revisar que se haya aplicado el proceso de mantenimiento en su totalidad o al menos las partes que se requieran en el proyecto.
- Analizar y comparar los resultados obtenidos luego de aplicar el proceso de mantenimiento, con el fin de conocer las observaciones, ventajas y desventajas en dicha ejecución.

Actuar

- Concluir aspectos fuertes y de mejora de “Mantelasoft”
- Concluir sobre lo que se hizo bien durante todas las fases del proyecto y qué se pudo haber mejorado o hecho diferente
- Plantear recomendaciones que quedan pendientes para futuras investigaciones

8.2 UNIDAD DE ANÁLISIS

El proyecto es un caso de estudio, por lo cual no se hará un análisis estadístico ni se aplicará un método de muestreo, sino que se tendrán en cuenta las empresas que han participado en el grupo de investigación y que han mostrado su interés por estar en el presente proyecto.

Las empresas que mostraron su interés en participar en el proceso fueron trece (13) y en dos (2) de ellas se realizó la aplicación de “Mantelasoft”.

8.3 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN

Encuesta: Se utilizaron encuestas para la primera fase de diagnóstico de mantenimiento en los departamentos de desarrollo y las MIPYMES, y para evaluar la calidad del contenido de la guía antes y después de su aplicación en las empresas.

La encuesta de diagnóstico se dividió por categorías, para que la persona encuestada tuviera una idea de qué aspectos se van a preguntar sobre el mantenimiento y haya una

mayor comprensión a la hora de responderla. Con base en tales referentes se plantearon las siguientes categorías:

- Generalidades del Mantenimiento
- Planificación del Mantenimiento
- Estimación de Costos
- Métricas del Mantenimiento
- Proceso de Mantenimiento
- Personal de Mantenimiento
- Técnicas y Herramientas de Mantenimiento
- Evolución o Mejora del Producto

Mediante la lectura y análisis del capítulo de Mantenimiento de Software de Swebook V3 (IEEE Computer Society, 2014), se plantearon las preguntas 1 a 9 pertenecientes a la categoría de Generalidades del Mantenimiento, las preguntas 18 a 25 pertenecientes a la categoría Proceso de Mantenimiento y las preguntas 29 y 30 pertenecientes a la categoría Evolución o Mejora del Producto.

A través del análisis del Métrica V3, en su apartado de “Mantenimiento de Sistemas de Información” se plantearon las preguntas 13 al 17 pertenecientes a la categoría Métricas del Mantenimiento y se planteó el cuadro de personal perteneciente a la categoría Personal de Mantenimiento.

El Manual de Competisoft fue base para el planteamiento de la pregunta 10 perteneciente a la categoría Planificación del Mantenimiento y las preguntas 26 a 28 perteneciente a la categoría Técnicas y Herramientas de Mantenimiento.

Por último, mediante el estudio y análisis del libro “Software Evolution and Maintenance” (Tripathy & Naik, 2015) y el artículo “Software Maintenance and Process Improvement by CMMI” (Jansson, 2007), fue posible la formulación de las preguntas 11 y 12 perteneciente a la categoría Estimación de Costos. Para algunas de estas referencias, se realizaron mapas mentales para conceptualizar mejor la información, como se puede ver en los Anexos [D](#), [E](#) y [F](#)

Para colaborar con el medio ambiente y agilizar el proceso de diligenciamiento de esta, se hizo la encuesta virtual y se envió el enlace a las empresas que nos ayudaron en este proceso. El contenido de la encuesta se presenta en el [Anexo A](#).

La evaluación de la aplicación de Mantelasoftware en las empresas, comprende 8 fases que son: generalidades del mantenimiento, planificación del mantenimiento, proceso de mantenimiento, métricas del mantenimiento, estimación de costos de mantenimiento, personal de mantenimiento, evolución o mejora del producto y actividades finales del mantenimiento; cada una de las cuales se divide en actividades, 64 en total, y que fueron calificadas según su nivel de cumplimiento por parte del equipo de mantenimiento de la empresa, al momento de realizar las tareas de mantenimiento de software solicitado por los clientes de la compañía.

Las 64 tareas fueron calificadas según los seis tipos de valores para la respuesta:

Tabla 2. Respuestas al cumplimiento de cada actividad de la guía de mantenimiento

RESPUESTAS AL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	
VALOR	ESTADO
0	NO SABE
1	NO SE CUMPLE
2	SE CUMPLE SIN REGISTROS
3	SE CUMPLE CON ALGUNOS REGISTROS
4	SE CUMPLE EN ALTO GRADO CON REGISTROS
5	SE CUMPLE PLENAMENTE CON REGISTROS

Fuente: Los Autores

Para la interpretación del resultado general se tuvo en cuenta la siguiente tabla de rangos:

Tabla 3. Rangos de interpretación del cumplimiento de la guía de mantenimiento

RANGOS DE INTERPRETACIÓN DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	
RANGO	ESTADO
Entre 0,0 y 2,0 Puntos	INSUFICIENTE
Entre 2,1 y 2,9 Puntos	DEFICIENTE
Entre 3,0 y 3,9 Puntos	REGULAR
Entre 4,0 y 4,5 Puntos	BUENO
Entre 4,6 y 5,0 Puntos	EXCELENTE

Fuente: Los Autores

Se realizaron tres formatos de evaluaciones, cada una para cada tipo de mantenimiento que son:

- Mantenimiento Correctivo Urgente
- Mantenimiento Correctivo No Urgente, Perfectivo y Preventivo
- Mantenimiento Adaptativo

El contenido de las encuestas se presenta en el [Anexo B](#).

La encuesta de evaluación de este proceso luego de su aplicación, y que tiene elementos comunes con la primera encuestados, cuenta con dos secciones. En la primera se evaluará la claridad, pertinencia y utilidad del contenido del documento que plantea un proceso y la segunda parte permiten preguntas para conocer puntos fuertes, puntos de mejora y otros atributos que permiten conocer la relevancia de este documento en su aplicación. El contenido de la encuesta se presenta en el [Anexo C](#).

Entrevista / Grupo Focal: Se realizó un grupo focal con las empresas que manifestaron el interés en participar en el proceso.

Para ello se le envió una invitación formal a cada empresa y se realizó una reunión en la Universidad Católica de Pereira donde se le dio a conocer el proyecto de grado y la encuesta que ellos iban a diligenciar. De esta reunión participaron trece (13) empresas de Pereira.

Y por último se dio un espacio para dialogar sobre el mantenimiento de software en la región, cómo lo hacían en sus empresas y expectativas que se generaban con el presente proyecto.

8.4 DESARROLLO DEL PROYECTO

8.4.1 Diagnóstico inicial del Mantenimiento de Software

Se realizó una revisión bibliográfica en temas de mantenimiento de software, para generar las preguntas de la encuesta del diagnóstico del mantenimiento de software en las empresas de la ciudad de Pereira, la cual se presenta en el [Anexo A](#).

La encuesta se realiza en una plataforma virtual para ser enviada por correo y así no incurrir en gasto innecesario de papel y ayudar al medio ambiente.

Contamos con la base de datos de las empresas desarrolladoras de software y afines que proporciona la Cámara de Comercio de la ciudad de Pereira, con un total de 47 registros al 2015, donde se seleccionaron 21 empresas que mostraron interés de participar en el proceso a las cuales se les envió un correo electrónico explicando brevemente el proyecto de grado, el objetivo de la encuesta, el enlace para diligenciarla y la tarjeta de invitación para el grupo focal.

Figura 9. Tarjeta de invitación al grupo focal



Fuente: Los autores

Con la ayuda del grupo de investigación “Entre Ciencia e Ingeniería” de la Universidad Católica de Pereira, siguiendo en su proceso de profundizar en la línea de investigación Ingeniería de Software y siguiendo el desarrollo del objetivo específico No. 1, se desarrolló el día 26 de octubre de 2015 a las 7:30 a.m., salón Oval 1, un grupo focal de empresarios del sector de desarrollo de software. De las 21 empresas interesadas, 13 confirmaron asistencia y asistieron 8 de ellas.

En la reunión se les hizo una introducción sobre el grupo de investigación, una breve exposición del objetivo, alcances del proyecto y se proporcionó la encuesta para su debido diligenciamiento a quienes no la habían diligenciado. Luego, se dio un desayuno y se generó un espacio para hablar sobre el tema de mantenimiento de software de acuerdo a las preguntas de la encuesta y, por último, manifestaron el interés de aplicar la guía y adaptarla a sus procesos de desarrollo.

A continuación, se presentan algunas imágenes sobre el grupo focal. En la Figura 6 se presenta el listado de empresas que mostraron interés en participar de la encuesta y el grupo focal que se realizó. La Figura 7 es el listado de asistencia de las empresas que hicieron parte del grupo focal. Y la Figura 8 y 9 fueron algunas fotos tomadas durante el grupo focal.

Figura 10. Listado de empresas interesadas en de participar en el proceso

		NOMBRE	CARGO	EMPRESA	TELEFONO	CORREO ELECTRONICO
1	INGENIERO	JULIAN BARBERY	GERENTE	COMPUMEDIOS - AVANSIS	3333352	gerencia@avansis.com.co
2	INGENIERO	JAIRO GUTIERREZ	DIRECTOR	LUCACIAN LAB	3400137	jgutierra@lucacian.com
3	INGENIERO	JHON CORTES		LUCACIAN LAB	3137682749	jcortes@lucacian.com
4	INGENIERO	ALBEIRO RIOS NAVARRO		UNE	3247204 - 3247200	arios@unp.com.co
5	INGENIERO	CRISTIAN YOHANNI LOAIZA DUQUE	GERENTE	DATACENTER	3716412051	cristian.loaiza@datacenter.com.co
5.5	INGENIERA	LORENA CARDONA BENJUMEA		DATACENTER		
8	INGENIERO	JAIRO ALBERTO MONROY OSPITIA	GERENTE	INDRA	3186126679	jamonroy@indra.company.com
9	INGENIERA	INGRID GALIANO RUIZ	GERENTE DE PROYECTOS	INDRA	317 5120368	mfrapatac@indra.company.com
10	INGENIERO	JUAN SEBASTIAN SANTACRUZ	GERENTE	IDRA	3164494502	santacruz-juan@hotmail.com , isantacruz@idra.com
11	INGENIERO	ANDRÉS DAVID RIOS		IDRA		arios@idrasoluciones.com
12	INGENIERO	JUAN SEBASTIAN DUQUE	GERENTE	PLUGAR	3103831139	juan.duque@plugar.co
13	INGENIERO	JUAN CARLOS MEJIA	GERENTE	IAIVANA	3163625392	iaivana@gmail.com
14	INGENIERO	JOHANA LOAIZA MESA	DIRECTOR EJECUTIVO	PARQUESOFT	3183795517	acadavid@parquesoftpereira.com
15	INGENIERO	JULIO CESAR CANO GONZALEZ		FISCALIA	3146824933	julio.cano@ucp.edu.co
16	INGENIERO	CARLOS URIEL PAREJA RODRIGUEZ		PROSOFT	3104538955	carlosurrielpareja@gmail.com
17	INGENIERO	FRANCISCO LUIS ALVAREZ		PROSOFT		
18	INGENIERO	TORRES CANO ANDRÉS FELIPE	GERENTE	EHMSOFTWARE SAS	3168746839 - 3266660	info@ehmssoft.com
19	INGENIERO	HUMBERTO PAREJA QUINTERO	COORDINADOR COMERCIAL	PARQUESOFT	3113500651	comercial@parquesoftpereira.com
20	INGENIERO	DANIEL BLANDON	DIRECTOR GESTION TECNOLOGIA	UCP	3124000 EXT. 1023	daniel.blandon@ucp.edu.co
21	INGENIERO	ANDRÉS FELIPE TORO	DIRECTOR	MIKOMOVIL SAS	3215290 - 3217014633	andres_toro@hotmail.com

Fuente: Elaboración propia

Figura 11. Listado de asistencia al grupo focal

UNIVERSIDAD CATOLICA DEL CAUCA		SECRETARÍA GENERAL			
Fecha de elaboración del formato: 22 de mayo de 2008		GESTION DOCUMENTAL			
		FORMATO LISTADO DE ASISTENCIA			
		ID: 2	Página 1 de 1		GD-ADM-05
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍA PROGRAMA: TECNOLOGÍA EN DESARROLLO DE SOFTWARE EVENTO INTERNO: DESAYUNO EMPRESARIOS LUGAR: SALÓN OVAL HORA INICIO: 7:30 AM HORA FINALIZACIÓN: 9:00 AM FECHA: OCTUBRE 26 DE 2016					
N	NOMBRES	CÉDULA	EMPRESA	CORREO	CELULAR
1	Juan Sebastián Salazar Pérez	1088267371	IDea Soluciones	jsalazar@ideasoluciones.com	3164494502
2	Fernando Luis Barrera Duran	10130551	PROSOF	pros@pros@pros@e.co	3113857934
3	Lorena Carolina Benjumea	42.100.2918	Omnesse SA	lorena.carolina@omnesse.com.co	3180888202
4	Johanna Laura Ussa	1.127.6925	Prof. UESFT	duess@prof.uesft@prof.com	3193375517
5	Diana Carolina López Lyce	1088292146	USP - G. Tecnológica	diana.lyce@usp.edu.co	3106771826
6	Julio César Cano Rómulo	10135123	Fiscalia - TIC	ayfca@p@fiscalia.gov.co	3446824953
7	Andrés Felipe Gutiérrez García	9862871	On Technology	agutierrez@onv.com.co	3127529911
8	Julián Andrés García	80409518	Compumedia	garci@compumedia.com.co	3104232503
9					
10					
11					
12					
13					
14					
15					

Fuente: Los autores

Figura 12. Asistentes al grupo focal



Fuente: Tomada por los integrantes del proyecto de grado

Figura 13. Asistentes al grupo focal



Fuente: Tomada por los integrantes del proyecto de grado

En total se lograron recopilar 13 encuestas.

Posterior a la recolección de la información, se tabularon cada una de las preguntas y con esta información se obtuvieron conclusiones que se pueden ver en el capítulo [“Conclusiones de la encuesta”](#).

8.4.2 Elaboración de MANTELASOFT

Con la ayuda de estas conclusiones y la revisión bibliográfica realizada para el proyecto, se formula el proceso de evolución y mantenimiento de software “Mantelasoft”.

La guía propuesta presenta un proceso para el mantenimiento de software que integra todas las actividades relacionadas con él. Su objetivo es convertir el mantenimiento de software en un proceso controlable y medible mediante la identificación de los elementos que intervienen en él.

El proceso para la evolución y mantenimiento de software fue producto de una síntesis del proceso Ágil Mantema, con el apoyo de los procesos de apoyo planteados por la norma ISO/IEC 12207:2008 y la encuesta utilizada para caracterizar las prácticas del mantenimiento de software en departamentos de sistemas y Mipymes desarrolladoras de software de la ciudad de Pereira. Se espera que el proceso se estructure para ser presentado como una alternativa interesante, al menos en el contexto de las MIPYMES y departamentos de desarrollo de software pereiranos.

Está dividida en 8 fases y cada una tiene actividades y tareas específicas que le permiten al lector, saber precisamente que debe realizar o tener en cuenta en su mantenimiento. Además, no requiere de mucho tiempo para poder aplicarla en los productos de software de la empresa, pues fue pensada como lista de chequeo para que fuera de mayor comprensión por las personas que la fueran a utilizar.

El documento final de “Mantelasoft” se encuentra como documento anexo a este proyecto. [Guía de Mantenimiento “Mantelasoft”](#)

8.4.3 Aplicación de MANTELASOFT en dos MIPYMES desarrolladoras de software

A fin de poder llevar a cabo la aplicación del proceso de evolución y mantenimiento de software, se escogieron dos Mipymes desarrolladoras de software de la ciudad de Pereira, que muy amablemente aceptaron ser parte de este estudio.

Con fines de tener confidencialidad de la información, no se pondrán el nombre real de las empresas ni el nombre de los aplicativos evaluados. En cambio, se pondrán nombres diferentes.

Para las dos empresas, las 64 actividades fueron calificadas según los seis tipos de valores, como se describió en el capítulo [*“Técnicas e instrumentos de recolección de información”*](#).

La evaluación comprende 8 fases que son: generalidades del mantenimiento, planificación del mantenimiento, proceso de mantenimiento, métricas del mantenimiento, estimación de costos de mantenimiento, personal de mantenimiento, evolución o mejora del producto y actividades finales del mantenimiento; cada una de las cuales se divide en actividades, 64 en total, y que fueron calificadas según su nivel de cumplimiento por parte del equipo de mantenimiento de la empresa, al momento de realizar las tareas de mantenimiento de software solicitado por los clientes de la compañía.

En cada una de las empresas se realizaron tres visitas en un tiempo de mes, donde en el primer encuentro fue de dos días donde en el primero se les presentó el proyecto de grado, la guía de mantenimiento realizada y el objetivo de cada punto de la esta. Además, se acuerda el producto de software que se va a evaluar. En el segundo día se realiza la evaluación inicial.

En el segundo encuentro, el cual fue a los quince días, se atendieron las dudas que tuvieron durante la aplicación de la guía, el avance que tuvieron y las mejoras que realizaron.

En el tercer encuentro, igualmente se atendieron dudas, y de allí surgieron algunas mejoras que se pueden realizar a la guía. Y, por último, se realizó la evaluación final.

Empresa 1

Evaluación inicial

Inicialmente se realizó una reunión con el equipo de mantenimiento de la empresa, conformado por tres ingenieros de sistemas. Los llevó a cabo una explicación de la tarea investigativa desarrollada, a fin de socializar las etapas que componen la guía de mantenimiento de software propuesta; pasando a aplicar la evaluación inicial del cumplimiento de la guía, la cual fue contestada por el equipo.

“Empresa 1”, presta el servicio de mantenimiento al software propio denominado “Software 1”, el cual es un software administrativo, contable y de gestión acorde a las normas internacionales de información financiera NIIFs, y el manejo de inventarios incluyendo facturación, sistema P.O.S., compras y todas las transacciones que este implica como órdenes de compras, pedidos, cotizaciones, comisiones a vendedores, manejo de cartera y proveedores, medios magnéticos, nómina, recurso humano, presupuesto comercial y público, producción, activos fijos, control y manejo de parqueaderos y un módulo adicional de auditoría que rige en todas las anteriores aplicaciones. Este software trabaja de manera integral y en tiempo real.

A continuación, se detalla el resultado de la evaluación inicial del cumplimiento de la guía de mantenimiento de software de la empresa, para los tipos de mantenimiento en estudio:

- Mantenimiento Correctivo Urgente
- Mantenimiento Correctivo No Urgente, Perfectivo y Preventivo
- Mantenimiento Adaptativo

Tabla 4. Evaluación inicial mantenimiento correctivo urgente – Empresa 1

EVALUACIÓN INICIAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE											
MANTENIMIENTO CORRECTIVO URGENTE											
FASE	No.	TAREA	RESPUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado con registros				
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre solicitud de los servicios de mantenimiento	0	1	2	3	4	5	5,0	5,0	REGULAR
	2	Preparar la propuesta de mantenimiento para el cliente						X	5,0		
	3	Elaborar el contrato de prestación de servicios de mantenimiento						X	5,0		
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades				X			3,0	3,6	
	2	Adquirir conocimiento de la aplicación. Verificar el estado del software - Estudiar la documentación, el código de los programas, referencias cruzadas y sus funcionamiento				X			3,0		
	3	Desmenujar planes. Estimar recursos necesarios (tiempo y personal) - Inventariar el sistema de información - Definir los herramientas para registrar los defectos que se reporten, para la gestión de aprendidas, entre otros		X					2,0		
	4	Definir procedimientos de petición de modificación					X		4,0		
	5	Preparar los entornos de pruebas. Realizar copias del entorno software para el trabajo					X		4,0		
	6	Recibir la petición de modificación. Recibir petición y analizarla					X		4,0		
	7	Decidir el tipo de mantenimiento que se requiere. Rechazar la petición, justificando la razón o aceptar la petición - Decidir las actividades de mantenimiento a aplicar						X	5,0		
PROCESO DE MANTENIMIENTO	1	Investigar y analizar las causas del error: Analizar la petición - Verificar el problema - Estudiar alternativas para implementar modificaciones - Listar elementos software a corregir				X			3,0	3,0	
	2	Realizar las acciones correctivas: Identificar las rutinas y bases de datos afectadas por la intervención		X					2,0		
	3	Delimitar la documentación exigida. Describir los cambios realizados				X			3,0		
	4	Realizar las pruebas unitarias. Completar la verificación de los cambios realizados				X			3,0		
	5	Pasar a producción el software: Enviar al entorno de producción el software corregido para su utilización					X		4,0		
METRICAS DEL MANTENIMIENTO	1	Desmenujar Planes de mantenimiento	1	Tiempo dedicado a la tarea		X			2,0	2,3	
	2		Número de puntos-función		X			3,0			
	3		Número de módulos		X			3,0			
	4		Número de líneas de código		X			3,0			
	5		Complejidad ciclomática media de cada módulo		X			2,0			
	6		Número de variables globales (visibles externamente) de cada módulo		X			3,0			
	7		Número de rutinas públicas (visibles externamente) de cada módulo		X			3,0			
	8	Descripción de mantenimiento	8	Petición aceptada o rechazada		X			2,0		
	9		Causa de la petición: Cambios legales - Evolución del negocio - Mejora del proceso - Cambio en las políticas de negocio - Otra (cual?)		X			2,0			
	10		Área o dominio funcional que ha presentado la petición		X			2,0			
	11	Investigar y analizar causas	11	Tiempo dedicado a la tarea		X			2,0		
	12		Número de puntos-función que previsiblemente se verán afectados por el error			X		3,0			
	13		Origen del error: Cambio en el análisis de requisitos - Cambio en el análisis de localización - Cambio en el análisis de diseño - Codificación			X		3,0			
	14		Causa del error: Desconocimiento del dominio de la aplicación - Desconocimiento del hecho o la implementación del Sistema - Requisitos ambiguos o incompletos - Error sintáctico o semántico del lenguaje de programación - Tronco en la planificación - Fallo durante el ensayo - Descuido			X		3,0			
	15	15	Tiempo dedicado a la tarea		X			2,0			
METRICAS DEL MANTENIMIENTO	16	Realizar acciones correctivas	16	Número de puntos-función alterados		X			2,0		
	17		Número de puntos-función modificados		X			2,0			
	18		Número de puntos-función borrados		X			2,0			
	19		Número de módulos alterados		X			2,0			
	20		Número de líneas de código alterados		X			2,0			
	21		Número de líneas de código modificados		X			2,0			
	22		Número de líneas de código borradas		X			2,0			
	23	Número de defectos encontrados al producto por mes		X			2,0				
	24	Complejidad ciclomática media de cada módulo alterado		X			2,0				
	25	Número de variables globales (visibles externamente) de cada módulo alterado		X			2,0				
	26	Número de rutinas públicas (visibles externamente) de cada módulo alterado		X			2,0				
	27	Medición de la calidad del código		X			1,0				
	28	Índice de calidad del software		X			1,0				
	29	Análisis de impacto de la corrección			X		3,0				
	30	Número de acciones de software alteradas			X	X	3,0				
31	Cada cuánto perfectamos el producto software			X		2,0					
32	Delimitar documentación	32	Tiempo dedicado a la tarea		X			2,0			
33		Número de páginas añadidas a la documentación, según el estándar de la Organización			X		3,0				
34		Especificar los requisitos		X			2,0				
35	Pasos a producción	35	Tiempo dedicado a la tarea		X			2,0			
36		36	36					2,0			
ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	1	Estimar los costos del mantenimiento del software: Definir la técnica utilizada				X			3,0	3,5	
	2	Identificar la técnica para estimar los costos de mantenimiento: Paramétrico - Matemático - Juicio de Experto - Otra (Cual?)					X	4,0			
	3	Definir el costo de ejecución del Hardware donde va a funcionar el producto de Software		X				3,0			
	4	Determinar el costo total del mantenimiento al producto					X	4,0			
PERSONAL DE MANTENIMIENTO	1	Verificar si el equipo de mantenimiento: Se encuentra dentro del desarrollo - Tiene conocimientos de la aplicación - Tiene conocimientos en el uso de técnicas y herramientas - Tiene conocimientos en lenguajes de programación - Está laboralmente estable en la empresa - Se encuentra motivado - Recibe retroalimentación sobre la calidad de su trabajo					X		4,0	4,0	
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Analizar: Se evolucionó o mejoró el producto sin que el cliente reporto los defectos				X			3,0	3,0	
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Registrar intervenciones				X			3,0	2,9	
	2	Actualizar base de datos: Incorporar las métricas a la base de datos histórica		X				1,0			
	3	Ejecutar backups: Realizar operaciones manuales sobre el software que se va a retirar y el software nuevo (si se sustituye)				X		3,0			
	4	Notificar retirada: Notificar al Cliente la retirada del software				X		3,0			
	5	Almacenar datos del entorno antiguo			X			2,0			
	6	Entrega del inventario y de la documentación: Entregar al cliente todos los productos software generados y modificados			X			2,0			
	7	Traspaso de experiencia y formación: Enseñar las nuevas características del sistema al cliente					X	4,0			
	8	Gestión de finaliza del servicio: Finalizar la prestación de los servicios al Cliente						X	5,0		

Tabla 5. Evaluación inicial mantenimiento correctivo no urgente, perfectivo y preventivo – Empresa

1

EVALUACIÓN INICIAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE														
MANTENIMIENTO CORRECTIVO NO URGENTE - PERFECTIVO - PREVENTIVO														
FASE	No.	TAREA	RESPUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN			
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado con registros					Se cumple plenamente con registros		
			0	1	2	3	4	5						
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre solicitud de los servicios de mantenimiento						X	5,0	5,0				
	2	Preparar la propuesta de mantenimiento para el cliente						X	5,0					
	3	Elaborar el contrato de prestación de servicios de mantenimiento						X	5,0					
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades				X			3,0	3,6				
	2	Adquirir conocimiento de la aplicación. Verificar el estado del software - Estudiar la documentación, el código de los programas, referencias cruzadas y su funcionamiento				X			3,0					
	3	Desarrollar planes. Estimar recursos necesarios (tiempo y personal). Inventariar el sistema de información - Definir los herramientas para reparar los defectos que se reporten, para la gestión de la configuración y versiones y para la gestión del conocimiento (Documentación, Lecciones aprendidas, entre otras)			X				2,0					
	4	Definir procedimientos de petición de modificación					X		4,0					
	5	Preparar los ensayos de pruebas. Realizar copias del sistema software para el trabajo					X		4,0					
	6	Recibir la petición de modificación. Recibir petición y analizarla					X		4,0					
	7	Decidir el tipo de mantenimiento requerido: Recharlar la petición, justificando la razón o aceptar la petición - Decidir las actividades de mantenimiento a realizar						X	5,0					
PROCESO DE MANTENIMIENTO	1	Validar la petición. Analizar las características de la petición - Verificar el problema - Listar elementos software a corregir				X			3,0	3,4				
	2	Documentar las posibles soluciones: Documentar la causa del error - Estudiar alternativas para implementar la modificación				X			3,0					
	3	Elegir la alternativa seleccionada (Esta tarea no se realiza para el Mantenimiento Preventivo)					X		4,0					
	4	Ejecutar la intervención: Ejecutar las acciones necesarias para servir la petición de modificación conforme a la alternativa seleccionada				X			3,0					
	5	Ejecutar las pruebas unitarias: Corroborar que la petición de modificación quedó servida				X			3,0					
	6	Ejecutar las pruebas de integración: Comprobar que los elementos software funcionan correctamente de forma conjunta				X			3,0					
	7	Ejecutar paralelamente en software antiguo y nuevo: Ejecutar acciones males en el producto software antiguo y en el modificado para detectar y prevenir posibles errores de proceso					X		4,0					
	8	Verificar y validar correctamente con el cliente					X		4,0					
	9	Reforzar manual de usuario			X				2,0					
	10	Pasar a producción: Instalar el producto software modificado en el entorno de trabajo real					X		4,0					
	11	Realizar revisión (Esta tarea no se realiza para el Mantenimiento Preventivo): Comprobar que el producto modificado funciona correctamente en el entorno de trabajo real					X		4,0					
MÉTRICAS DEL MANTENIMIENTO	1	Demora/Tiempo			X				2,0	2,5				
	2	Tiempo dedicado a la tarea			X				2,0					
	3	Número de pruebas falladas			X				3,0					
	4	Número de pruebas falladas			X				3,0					
	5	Número de líneas de código			X				3,0					
	6	Capacidad de memoria (bytes) de cada módulo			X				3,0					
	7	Número de variables globales (variables externamente de cada módulo)			X				3,0					
	8	Número de rutinas públicas (variables externamente de cada módulo)			X				3,0					
	9	Petición aceptada o rechazada			X				2,0					
	10	Causa de la petición: Cambios legales - Evolución del negocio - Mejora del proceso			X				2,0					
	11	Cambio en las políticas de negocio - Otra (real?)			X				2,0					
	12	Área o dominio funcional que ha presentado la petición			X				2,0					
	13	Tiempo dedicado a la tarea			X				2,0					
	14	Fecha estimada para el inicio de la intervención			X				3,0					
	15	Fecha estimada para la finalización de la intervención			X				3,0					
	16	Número de pruebas falladas que preventivamente se verían afectadas por el error			X				3,0					
	17	Origen del error: Cambio en el análisis de requisitos - Cambio en el análisis de localización - Cambio en el análisis de diseño - Codificación			X				3,0					
18	Causa del error: Desconocimiento del destino de la aplicación - Desconocimiento del objeto a la implementación del Sistema - Requisitos ambiguos o inconsistentes - Error sintáctico o semántico del lenguaje de programación - Tercer error en la planificación - Fallo en el entorno de desarrollo - Descuido			X				2,0						
MÉTRICAS DEL MANTENIMIENTO	18	Elegir la alternativa seleccionada (Esta tarea no se realiza para el Mantenimiento Preventivo)			X				2,0	2,5				
	19	Tiempo dedicado a la tarea			X				2,0					
	20	Número de pruebas falladas			X				2,0					
	21	Número de pruebas falladas			X				2,0					
	22	Número de pruebas falladas			X				2,0					
	23	Número de líneas de código			X				2,0					
	24	Número de líneas de código			X				2,0					
	25	Número de líneas de código			X				2,0					
	26	Capacidad de memoria (bytes) de cada módulo alterado			X				2,0					
	27	Número de variables globales (variables externamente de cada módulo)			X				2,0					
	28	Número de rutinas públicas (variables externamente de cada módulo)			X				2,0					
	29	Tiempo dedicado a la tarea			X				3,0					
	30	Número de errores detectados			X				3,0					
MÉTRICAS DEL MANTENIMIENTO	31	Tiempo dedicado a la tarea			X				3,0	3,5				
	32	Número de errores detectados			X				3,0					
	33	Tiempo dedicado a la tarea			X				3,0					
	34	Número de errores detectados			X				3,0					
	35	Tiempo dedicado a la tarea			X				3,0					
	36	Número de errores detectados			X				3,0					
	37	Tiempo dedicado a la tarea			X				3,0					
	38	Número de páginas añadidas/borradas/modificadas al manual			X				3,0					
	39	Tiempo dedicado a la tarea			X				3,0					
	40	Tiempo dedicado a la tarea (Esta tarea no se realiza para el Mantenimiento Preventivo)			X				3,0					
	ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	1	Estimar los costos del mantenimiento del software: Definir la técnica utilizada				X					3,0	3,5	
		2	Identificar los recursos para estimar los costos de mantenimiento: Paramétrico - Matemático - Juicio de Expertos - Otra (¿ cuál?)				X					4,0		
		3	Definir el costo de operación del Hardware donde va a funcionar el producto de Software				X					3,0		
4		Documentar el costo total del mantenimiento al producto					X		4,0					
PERSONAL DE MANTENIMIENTO	1	Verificar el equipo de mantenimiento: Se encuentra dentro del presupuesto - Tener conocimiento de la aplicación - Tener conocimientos en el uso de técnicas y herramientas - Tener conocimiento en lenguajes de programación - Está laboralmente estable en la empresa - Se encuentra motivado - Recibe reconocimiento sobre la calidad de su trabajo				X			4,0	4,0				
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Analizar: Se evoluciona o mejora el producto sin que el cliente reporte los defectos				X			3,0	3,0				
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Realizar intervención				X			3,0	2,9				
	2	Actualizar base de datos: Incorporar las métricas a la base de datos histórica		X					1,0					
	3	Ejecutar paralelos: Realizar operaciones entre el software que se va a retirar y el software nuevo (o se modifica)				X			3,0					
	4	Notificar retirada: Notificar al Cliente la retirada del software				X			3,0					
	5	Actualizar datos del sistema antiguo			X	X			2,0					
	6	Entrega del inventario y de la documentación: Entregar al cliente todos los productos software generados y modificados			X				2,0					
	7	Tiempo de experiencia y formación: Enseñar las nuevas características del sistema al cliente					X		4,0					
	8	Cerrar definitivo del servicio: Finalizar la prestación de los servicios al Cliente						X	5,0					

REGULAR

3,5

Tabla 6. Evaluación inicial mantenimiento adaptativo –Empresa 1

EVALUACIÓN INICIAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE												
MANTENIMIENTO ADAPTATIVO												
FASE	Nº	TAREA	RESPUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN	
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado con registros					Se cumple plenamente con registros
			0	1	2	3	4	5				
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre solicitud de los servicios de mantenimiento						X	5,0	5,0	REGULAR	
	2	Preparar la propuesta de mantenimiento para el cliente						X	5,0			
	3	Elaborar el contrato de prestación de servicios de mantenimiento						X	5,0			
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades				X			3,0	3,6		
	2	Adquirir conocimiento de la aplicación. Verificar el estado del software - Establecer la documentación, el código de los programas, referencias cruzadas y su funcionamiento				X			3,0			
	3	Desarrollar planes. Estimar recursos necesarios (tiempo y personal) - Iniciar el sistema de información - Definir las herramientas para registrar los defectos que se reportan, para la gestión de la configuración y versiones y para la gestión del conocimiento (Documentación, Lecciones aprendidas, entre otros)			X				2,0			
	4	Definir procedimientos de petición de modificaciones						X	4,0			
	5	Preparar los entornos de pruebas. Realizar copia del entorno software para el trabajo						X	4,0			
	6	Recibir la petición de modificación. Recibir petición y radicarla						X	4,0			
	7	Decidir el tipo de mantenimiento requerido. Rechazar la petición, justificando la razón o aceptar la petición - Escalar las actividades de mantenimiento a aplicar						X	5,0			
PROCESO DE MANTENIMIENTO	1	Realizar el diagnóstico				X			3,0	3,4		
	2	Realizar copia del producto software. Realiza una copia del producto software, los bases de datos y los archivos relacionados						X	4,0			
	3	Ejecutar la intervención. Ejecutar las acciones necesarias para servir la petición de modificación conforme a la alternativa seleccionada				X			3,0			
	4	Ejecutar las pruebas unitarias. Comprobar la que la petición de modificación puede ser validada				X			3,0			
	5	Ejecutar las pruebas de integración. Comprobar la que los elementos software funcionan correctamente de forma conjunta				X			3,0			
	6	Ejecutar paralelamente en software antiguo y nuevo. Ejecutar acciones reales en el producto software antiguo y en el modificado para detectar y prevenir posibles errores de proceso						X	4,0			
	7	Verificar y validar corrección con el cliente				X			3,0			
	8	Redocumentar manual de usuario			X				2,0			
	9	Pasar a producción. Instalar el producto software modificado en el entorno de trabajo real						X	4,0			
	10	Realizar revisión. Funciona correctamente en el entorno de trabajo real						X	4,0			
	11	Archivar datos del producto software inicial. Realizar una copia de seguridad de todo lo relacionado con el producto software que se ha adaptado (programas, datos y documentos)						X	4,0			
MÉTRICAS DEL MANTENIMIENTO	1	Desarrollar Planes	Tiempo dedicado a la tarea			X			2,0	3,5		
	Número de puros-función				X			3,0				
	Número de módulos				X			3,0				
	Número de líneas de código				X			3,0				
	Complejidad ciclomática por módulo				X			2,0				
	Número de variables globales (externas) de cada módulo				X			3,0				
	Número de rutinas públicas (visibles externamente) de cada módulo				X			3,0				
	2	Decisión sobre mantenimiento	Petición aceptada o rechazada			X			2,0			
	3		Causa de la petición. Cambios legales - Evolución del negocio - Mejora del proceso. Cambios en las políticas de seguros - Otra (causa?)			X			2,0			
	4		Área o dominio funcional que ha presentado la petición			X			2,0			
	5	Ejecutar la intervención	Tiempo dedicado a la tarea			X			2,0			
	6		Número de puros-función afectados			X			2,0			
	7		Número de puros-función modificados			X			2,0			
	8		Número de errores detectados			X			2,0			
	9		Número de líneas de código alteradas			X			2,0			
10	Número de líneas de código borradas				X			2,0				
11	Complejidad ciclomática por módulo alterado				X			2,0				
12	Ejecutar las pruebas unitarias	Tiempo dedicado a la tarea			X			3,0				
13		Número de errores detectados			X			3,0				
14		Tiempo dedicado a la tarea			X			3,0				
15	Ejecutar las pruebas de integración	Número de errores detectados			X			3,0				
16		Tiempo dedicado a la tarea			X			3,0				
17		Número de errores detectados			X			3,0				
18	Ejecutar las pruebas de aceptación y nuevos	Tiempo dedicado a la tarea			X			3,0				
19		Número de errores detectados			X			3,0				
20		Tiempo dedicado a la tarea			X			3,0				
21	Ejecutar la verificación y validación con el cliente	Número de errores detectados			X			3,0				
22		Tiempo dedicado a la tarea			X			3,0				
23		Número de errores detectados			X			3,0				
24	Redocumentar manual de usuario	Tiempo dedicado a la tarea			X			3,0				
25		Número de páginas añadidas/borradas/modificadas al manual			X			3,0				
26		Tiempo dedicado a la tarea			X			3,0				
27	Pasar a producción	Tiempo dedicado a la tarea			X			3,0				
28		Tiempo dedicado a la tarea			X			3,0				
29	Realizar revisión	Tiempo dedicado a la tarea			X			3,0				
30		Tiempo dedicado a la tarea			X			3,0				
ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	1	Estimar los costos del mantenimiento del software. Definir la técnica utilizada				X			3,0	3,5		
	2	Identificar la técnica para estimar los costos de mantenimiento: Paramétrico - Método - Juicio de Experto - Otro (Cuál?)					X		4,0			
	3	Definir el costo de operación del Hardware donde se va a funcionar el producto de Software					X		3,0			
	4	Determinar el costo total del mantenimiento al producto					X		4,0			
PERSONAL DE MANTENIMIENTO	1	Verificar si el equipo de mantenimiento. Se encuentra dentro del desarrollo - Tiene conocimiento de la aplicación. Tiene conocimiento en el uso de técnicas y herramientas - Tiene conocimiento en lenguaje de programación. Está laboralmente estable en la empresa - Se encuentra motivado - Recibe retroalimentación sobre la calidad de su trabajo						X	4,0	4,0		
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Analizar. Se evolucionó o mejoró el producto sin que el cliente reporte los defectos				X			3,0	3,0		
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Registrar intervención				X			3,0	2,9		
	2	Actualizar base de datos. Incorporar las métricas a la base de datos histórica		X					1,0			
	3	Ejecutar pruebas. Realizar operaciones reales sobre el software que se va a retirar y el software nuevo (si se sustituye)				X			3,0			
	4	Notificar retirada. Notificar al Cliente la retirada del software				X			3,0			
	5	Archivar datos del entorno antiguo			X				2,0			
	6	Entrega del inventario y de la documentación. Entregar al cliente todos los productos software generados y modificados			X				2,0			
	7	Tiempo de experiencia y formación. Enseñar los nuevos características del sistema al cliente					X		4,0			
	8	Cierre definitivo del servicio. Finalizar la prestación de los servicios al Cliente						X	5,0			

Con la evaluación realizada, se observó que en cuanto al mantenimiento en general del producto, la empresa obtiene la información de la solicitud de mantenimiento, prepara la propuesta y elabora el contrato de mantenimiento, dejando documentación al respecto en su repositorio de archivos.

En la planificación del mantenimiento, tienen una herramienta para registrar los defectos que se reportan y se va gestionando su ciclo de vida. Para la gestión de la configuración y versiones, tienen un sistema manual que van registrando en archivos las versiones nuevas, pero no se documentan los cambios que se realizan. Para la gestión del conocimiento tiene documentación sobre procesos de desarrollo y mantenimiento del sistema de información, pero no tienen documentación de lecciones aprendidas ni documentación técnica que pueda servir para el proyecto. Elaboran los cronogramas de actividades y solo algunos de ellos quedan registrados y guardados en el repositorio de la empresa. Las peticiones que reciben están bien identificadas y quedan radicadas en un sistema interno que tienen. Además de que las decisiones que toman para aceptar o rechazar alguna petición de mantenimiento, que da registrado en el mismo sistema de información.

Con respecto a la estimación de costos, para definir la técnica a utilizar, tienen pocos registros históricos que puedan servir como base para que ayude en una estimación más acertada. Para definir el costo de operación, solo tienen registros del costo de compra de servidores y licencias, pero no tienen registro del costo de la energía que consume los servidores, el aire acondicionado que se necesita para mantenerlo refrigerado, el costo del espacio del hardware.

La empresa cuenta con un grupo de desarrolladores mixto, es decir, personal que tiene amplia experiencia en el lenguaje que desarrollan y buen conocimiento del negocio y también personal que apenas está adquiriendo experiencia. Para el mantenimiento del software, cuando llegan correctivos urgentes, asignan a la persona que no tenga carga laboral para realizarlo, sin importar si tiene o no experiencia. Además, no cuentan con una

trazabilidad suficientemente detallada para saber qué módulos se han corregido y cuáles fueron las correcciones. Para poder saberlo, se tiene que acudir a la persona que lo realizó y que retroalimente de qué fue lo que hizo.

Para la evolución del producto, se realizan pruebas unitarias, de integración y aceptación para minimizar la cantidad de fallos cuando se encuentre en producción. Para la mejora del producto, no se tienen estrategias o mecanismos para que el software tenga nuevas funcionalidades o mejorar las que ya se tienen.

Para las actividades finales del mantenimiento, no incorporan las métricas obtenidas del mantenimiento a una base de datos o repositorio, para llevar datos históricos de lo realizado. Almacenan poca información sobre los cambios realizados y el entorno que se estaba utilizando. Al cliente, le hacen entrega formal del mantenimiento, con el detalle de lo realizado y se realiza formación si es necesario.

Ahora, entrando al detalle de cada tipo de mantenimiento para las secciones del proceso y métricas del mantenimiento correctivo, cuando se recibe el error reportado, se hace su análisis y verificación, pero para ello siempre deben recurrir a la persona que desarrolló esa parte y preguntar que puede estar mal. Esto sucede porque el conocimiento se centra en el desarrollador y no lo comparten con el equipo de trabajo para evitar este tipo de inconveniente. Además, se tienen evidencias de pocas correcciones que se han realizado, donde el qué se hizo, que componentes se modificaron, la persona y la fecha.

Por otra parte, para poder hacer una revisión de lo que implica la corrección, debe acudir al código o hablar con desarrolladores y arquitectos para saberlo, pues, aunque hay documentación del software, no siempre se actualiza cuando se hacen correctivos. Con respecto a las métricas, se tienen pocos registros o evidencias de tiempos que incurren al realizar las tareas desde el análisis hasta las pruebas y evidencias del error reportado. Por otro lado, no tienen mediciones sobre la calidad del código que tienen. Estas métricas que

no se tienen en cuenta, hace evidenciar la falta de un repositorio software que les permita conocer el estado del software.

En cuanto al proceso de mantenimiento correctivo no urgente, preventivo, perfectivo, y adaptativo, durante todo el ciclo de desarrollo de la petición, se dejan pocos registros de lo que se hace, por lo que se hace complicado después volver a revisar algo de lo que ya se había hecho. Solo se tienen registros del código que se sube al repositorio donde se encuentran todos los desarrollos y donde se guardan las peticiones realizadas. Pero no se tiene documentos o diagramas donde se evidencie los componentes implicados en las modificaciones, ni tampoco la calidad del código con el cual se entrega la petición. Por otro lado, muy pocas veces se actualiza la documentación con los nuevos cambios realizados y los que se ajustan, no quedan con el histórico de los ajustes hechos previamente.

Proceso de aplicación y Evaluación Final

Se realizaron 3 jornadas de capacitación con el equipo de mantenimiento de la empresa, durante las cuales se explicaron cada una de las actividades que hacen parte de las ocho fases de la guía de mantenimiento de software propuesta por los dos investigadores.

Una vez capacitados los tres integrantes del equipo de mantenimiento en el cumplimiento de la guía, se llevó a cabo dos visitas para aplicar la evaluación final para los mantenimientos correctivo urgente, correctivo no urgente – perfectivo – preventivo y adaptativo. Para esta tarea, los investigadores interrogaron sobre cada una de las actividades verificando la respuesta obtenida y revisando los cambios realizado para así comprobar la calificación final.

Teniendo en cuenta la evaluación inicial y la capacitación dada, el equipo de mantenimiento, decidió realizar algunas mejoras en su proceso de mantenimiento de software.

Las mejoras encontradas fueron las siguientes:

Para la planificación del mantenimiento, la empresa incorporó las siguientes mejoras:

- Más detalle en la elaboración del cronograma de actividades
- Adicionaron nuevas configuraciones al sistema de registro de defectos para obtener reportes personalizados y poder registrar más detalle de cada defecto
- Para la gestión del conocimiento, implementaron un tipo de blog interno para la documentación técnica que pueda servir a todo el personal, lecciones aprendidas y otra documentación común al proyecto y la empresa.
- Agregaron nuevos detalles al proceso de petición de modificaciones para categorizar el tipo de modificación, que tan grande puede ser y que impacto puede tener.
- Comenzaron a re-documentar los cambios realizados en el análisis, diseño y codificación. Adicionando a los documentos una parte para controlar los cambios realizados y las versiones de cada documento que se van actualizando.
- El equipo comenzó a utilizar librerías que les ayudara a realizar las pruebas unitarias y dejar de lado la subjetividad de la prueba en cada desarrollador.

Para el proceso de mantenimiento, se comenzó a documentar las correcciones realizadas y los elementos o componentes que se corregían, para poder llevar una trazabilidad y saber a futuro que implicaciones se puede tener con otras peticiones. En este punto, se les recomendó utilizar la herramienta “Enterprise Architect” la cual ayuda en el proceso de trazabilidad de casos de uso y componentes involucrados.

Para las métricas del mantenimiento de software, fueron más rigurosos con el registro de los tiempos que tomaba cada tarea, con el fin de que más adelante puedan utilizar esta información para estimar las tareas, ya que por el momento es a juicio de

expertos. Se les recomienda utilizar la herramienta llamada Sonar para medir la calidad de código y a su vez obtener métricas que permitan conocer el estado del desarrollo, del componente y del sistema.

Con los cambios que se han ido implementando, anteriormente mencionados, pudieron analizar mejor las causas del error reportado o de la solicitud de mantenimiento ya que se apoyan en la documentación que han ido actualizando y las herramientas que les permite revisar la trazabilidad y dependencia entre componentes.

Al momento de la estimación de los costes del mantenimiento, les fue de gran utilidad tener seguimiento de los tiempos registrados ya que comenzaron a utilizar una técnica paramétrica que van a ir puliendo con el tiempo, mientras tanto seguirán con el juicio de expertos. Además, ya tienen en cuenta los costos en hardware como servidores y computadores, en software todas las licencias que necesitan para desarrollar el sistema y mantener el sistema y, además, tuvieron en cuenta un porcentaje de gastos generales como la energía que consumen todos estos elementos. Y finalmente con esta información más detallada, pudieron ajustar y tener una visión más amplia del coste total de su mantenimiento.

Para la evolución y mejora del producto, tomaron la iniciativa de realizar reuniones mensuales, en las cuales cada persona aporta ideas de posibles fallos en el sistema o qué mejoras o complementos se pueden realizar al software. Esta actividad es nueva y están definiendo formatos de reunión y seguimiento de este.

Las actividades finales del mantenimiento mejoraron porque se comenzó a registrar las métricas obtenidas en los repositorios que tienen destinados para ello. Al momento de las pruebas finales, comenzaron a hacer pruebas en paralelo con el sistema viejo y el nuevo, lo cual les ha servido para encontrar fallos en la integración del sistema. Y, por último, se realiza una entrega más detallada al cliente, con los componentes modificados, detalle de lo que se realizó con sus respectivos diseños si los hay y documentación de operación e instalación si aplica.

Los resultados fueron muy gratificantes para ambas partes, pues los puntajes obtenidos fueron mayores que los obtenidos en las evaluaciones iniciales. A continuación, se muestran los resultados obtenidos:

Tabla 7. Evaluación final mantenimiento correctivo urgente –Empresa 1

EVALUACIÓN FINAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE												
MANTENIMIENTO CORRECTIVO URGENTE												
FASE	No.	TAREA	RESPUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN	
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado con registros					Se cumple plenamente con registros
			0	1	2	3	4					5
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre solicitud de los servicios de mantenimiento						X	5,0	5,0		
	2	Preparar la propuesta de mantenimiento para el cliente						X	5,0			
	3	Elaborar el contrato de prestación de servicios de mantenimiento						X	5,0			
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades					X		4,0	4,6		
	2	Adquirir conocimiento de la aplicación: Verificar el estado del software - Estudiar la documentación, el código de los programas, referencias cruzadas y su funcionamiento						X	5,0			
	3	Desarrollar planes: Estimar recursos necesarios (tiempo y personal) - Inventariar el sistema de información - Definir las herramientas para registrar los defectos que se reportan, para la gestión de la configuración y versiones y para la gestión del conocimiento (Documentación, Lecciones aprendidas, entre otras)						X	5,0			
	4	Definir procedimientos de petición de modificación						X	5,0			
	5	Preparar los entornos de pruebas: Realizar copias del entorno software para el trabajo						X	4,0			
	6	Recibir la petición de modificación: Recibir petición y radicalarla						X	4,0			
	7	Decidir el tipo de mantenimiento requerido - Revisar la petición, justificando la razón o aceptar la petición - Decidir las actividades de mantenimiento a aplicar						X	5,0			
PROCESO DE MANTENIMIENTO	1	Investigar y analizar las causas del error: Analizar la petición - Verificar el problema - Estudiar alternativas para implementar modificación - Listar elementos software a corregir							X	5,0	4,6	
	2	Realizar las acciones correctivas: Identificar las raíces y bases de datos afectadas por la intervención							X	5,0		
	3	Diligenciar la documentación exigida: Documentar los cambios realizados						X	4,0			
	4	Ejecutar las pruebas unitarias: Comprobar la corrección de los cambios realizados						X	5,0			
	5	Passar a producción el software: Enviar al entorno de producción el software corregido para su utilización						X	4,0			
MÉTRICAS DEL MANTENIMIENTO	1	Desarrollar Planes	Tiempo dedicado a la tarea					X	4,0	4,3		
	2		Número de puntos-función					X	5,0			
	3		Número de módulos					X	5,0			
	4		Número de líneas de código					X	4,0			
	5	Decidir tipo de mantenimiento	Complejidad ciclométrica media de cada módulo					X	4,0			
	6		Número de variables globales (visibles externamente) de cada módulo					X	4,0			
	7		Número de rutinas públicas (visibles externamente) de cada módulo					X	4,0			
	8		Petición aceptada o rechazada						X			5,0
	9	Investigar y analizar causas	Causa de la petición: Cambios legales - Evolución del negocio - Mejora del proceso - Cambio en las políticas de negocio - Otra (null)						X			5,0
	10		Área o dominio funcional que ha presentado la petición						X			5,0
	11		Tiempo dedicado a la tarea						X			5,0
	12	Realizar acciones correctivas	Número de puntos-función que previsiblemente se verán afectados por el error					X	4,0			
	13		Origen del error: Cambio en el análisis de requisitos - Cambio en el análisis de localización - Cambio en el análisis de diseño - Codificación					X	4,0			
	14		Causa del error: Desconocimiento del dominio de la aplicación - Desconocimiento del diseño o la implementación del Sistema - Requisitos ambiguos o incompletos - Error sintáctico o semántico del lenguaje de programación - Tensión en la planificación - Fallo existente no detectado - Descuido						X			5,0
15	Realizar acciones correctivas	Tiempo dedicado a la tarea					X	4,0				
16		Número de puntos-función añadidos					X	4,0				
17		Número de puntos-función modificados					X	4,0				
18		Número de puntos-función borrados					X	4,0				
19		Número de módulos alterados					X	4,0				
20		Número de líneas de código añadidas					X	4,0				
21		Número de líneas de código modificadas					X	4,0				
22		Número de líneas de código borradas					X	4,0				
23		Número de defectos corregidos al producto por mes					X	4,0				
24		Complejidad ciclométrica media de cada módulo alterado					X	4,0				
25		Número de variables globales (visibles externamente) de cada módulo					X	4,0				
26		Número de rutinas públicas (visibles externamente) de cada módulo					X	4,0				
27		Medición de la calidad del código					X	4,0				
28		Índice de madurez del software					X	4,0				
29		Análisis de impacto de la corrección					X	4,0				
30		Número de retritos de software realizados					X	4,0				
31		Cada cuánto perfeccionan el producto software					X	4,0				
32	Diligenciar documentación	Tiempo dedicado a la tarea						X	5,0			
33		Número de páginas añadidas a la documentación, según el estándar de la Organización						X	5,0			
34	Ejecutar las pruebas unitarias	Tiempo dedicado a la tarea						X	5,0			
35	Passar a producción	Tiempo dedicado a la tarea						X	5,0			
ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	1	Estimar los costos del mantenimiento del software: Definir la técnica utilizada						X	4,0	4,5		
	2	Identificar la técnica para estimar los costos de mantenimiento: Paramétrico - Matemático - Juicio de Expertos - Otro (Null)						X	4,0			
	3	Definir el costo de operación del Hardware donde va a funcionar el producto de Software						X	5,0			
	4	Determinar el costo total del mantenimiento al producto						X	5,0			
PERSONAL DE MANTENIMIENTO	1	Verificar si el equipo de mantenimiento: Se encuentra dentro del desarrollo - Tiene conocimiento de la aplicación - Tiene conocimientos en el uso de técnicas y herramientas - Tiene conocimiento en lenguajes de programación - Está laboralmente estable en la empresa - Se encuentra motivado - Recibe retroalimentación sobre la calidad de su trabajo						X	4,0	4,0		
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Analizar: Se evoluciono o mejora el producto sin que el cliente reporte los defectos						X	5,0	5,0		
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Registrar intervenciones						X	5,0	4,8		
	2	Actualizar base de datos: Incorporar las métricas a la base de datos histórica						X	5,0			
	3	Ejecutar paralelos: Realizar operaciones reales sobre el software que se va a retirar y el software nuevo (si se sustituye)						X	5,0			
	4	Notificar retirada: Notificar al Cliente la retirada del software						X	5,0			
	5	Almacenar datos del entorno antiguo						X	4,0			
	6	Entrega del inventario y de la documentación: Entregar al cliente todos los productos software generados y modificados						X	5,0			
	7	Traspaso de experiencia y formación: Enseñar las nuevas características del sistema al cliente						X	4,0			
	8	Cesión definitiva del servicio: Finalizar la prestación de los servicios al Cliente						X	5,0			

EXCELENTE

4,6

Tabla 8. Evaluación final mantenimiento correctivo no urgente, perfecto y preventivo –Empresa 1

EVALUACIÓN FINAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE													
MANTENIMIENTO CORRECTIVO NO URGENTE - PERFECTIVO - PREVENTIVO													
FASE	No.	TAREA	RESPUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACION		
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado con registros					Se cumple plenamente con registros	
			0	1	2	3	4	5					
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre solicitud de los servicios de mantenimiento						X	5,0	5,0			
	2	Preparar la propuesta de mantenimiento para el cliente						X	5,0				
	3	Elaborar el contrato de prestación de servicios de mantenimiento						X	5,0				
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades					X		4,0	4,3			
	2	Adquirir conocimiento de la aplicación. Verificar el estado del software. Estudiar la documentación, el código de los programas, referencias cruzadas y su funcionamiento					X		4,0				
	3	Desarrollar planes. Estimar recursos necesarios (tiempo y personal). Inventariar el sistema de información. Definir los herramientas para registrar los defectos que se reportan, para la gestión de la configuración y versiones y para la gestión del conocimiento (Documentación, Lecciones aprendidas, entre otros)					X		4,0				
	4	Definir procedimientos de petición de modificación						X	5,0				
	5	Preparar los entornos de pruebas. Realizar copias del sistema software para el trabajo					X		4,0				
	6	Recibir la petición de modificación. Recibir petición y validarla					X		4,0				
	7	Decidir el tipo de mantenimiento requerido. Rechazar la petición, justificando la razón o aceptar la petición. – Decidir las actividades de mantenimiento a aplicar						X	5,0				
PROCESO DE MANTENIMIENTO	1	Valorar la petición. Analizar las características de la petición. Verificar el problema. Listar elementos software a corregir						X	5,0	4,3			
	2	Documentar los posibles soluciones. Documentar la causa del error. Estudiar alternativas para implementar la modificación						X	5,0				
	3	Elegir la alternativa adecuada (Esta tarea no se realiza para el Mantenimiento Preventivo)					X		4,0				
	4	Ejecutar la intervención. Ejecutar las acciones necesarias para servir la petición de modificación conforme a la alternativa seleccionada					X		4,0				
	5	Ejecutar las pruebas unitarias. Comprobar la que la petición de modificación quede servida					X		4,0				
	6	Ejecutar las pruebas de integración. Comprobar la que los elementos software funcionan correctamente de forma conjunta					X		4,0				
	7	Ejecutar paralelamente en software antiguo y nuevo. Ejecutar acciones reales en el producto software antiguo y en el modificado para detectar y presentar posibles errores de proceso					X		4,0				
	8	Verificar y validar corrección con el cliente					X		4,0				
	9	Redocumentar manual de usuario						X	5,0				
	10	Pasar a producción. Instalar el producto software modificado en el entorno de trabajo real					X		4,0				
	11	Realizar revisión (Esta tarea no se realiza para el Mantenimiento Preventivo). Comprobar que el producto modificado funciona correctamente en el entorno de trabajo real					X		4,0				
MÉTRICAS DEL MANTENIMIENTO	1	Desarrollar Planes	Tiempo dedicado a la tarea					X	5,0	4,4	BUENO		
	2		Número de errores detectados					X	4,0				
	3		Número de errores detectados					X	4,0				
	4		Número de errores detectados					X	4,0				
	5		Completitud de la última media de cada módulo					X	4,0				
	6		Número de errores detectados (cubiertos externamente) de cada módulo					X	4,0				
	7		Número de errores detectados (cubiertos externamente) de cada módulo					X	4,0				
	8		Petición aceptada o rechazada					X	5,0				
	9	Decidir tipo de mantenimiento	Causa de la petición. Cambios legales – Evaluación del impacto – Mejora del proceso. Cambio en las políticas de negocio – Otra (cuál?)					X	5,0				
	10		Área o dominio funcional que ha presentado la petición					X	5,0				
	11		Tiempo dedicado a la tarea					X	5,0				
	12	Validar la petición	Fecha estimada para el inicio de la intervención					X	4,0				
	13		Fecha estimada para la finalización de la intervención					X	4,0				
	14		Tiempo dedicado a la tarea					X	5,0				
	15	Documentar las posibles soluciones	Número de pruebas función que previamente se veían afectadas por el error					X	5,0				
	16		Origen del error. Cambio en el análisis de requisitos – Cambio en el análisis de localización. Cambio en el análisis de diseño. Codificación					X	5,0				
	17		Causa del error. Desconocimiento del dominio de la aplicación. Desconocimiento del diseño o la implementación del Sistema. Requisitos ambiguos o incompletos. Error sintáctico o semántico del lenguaje de programación. Tensión en las planificaciones. Fallo cuando es probable. Descuido					X	4,0				
18	Área o dominio funcional que ha presentado la petición						X	5,0					
19	Elegir la alternativa adecuada	Tiempo dedicado a la tarea (Esta tarea no se realiza para el Mantenimiento Preventivo)					X	4,0					
20		Tiempo dedicado a la tarea					X	5,0					
21		Número de pruebas función detectados					X	4,0					
22		Número de pruebas función modificados					X	4,0					
23		Número de errores detectados					X	4,0					
24		Número de errores detectados					X	4,0					
25		Número de errores detectados					X	4,0					
26		Número de errores detectados					X	4,0					
27		Completitud de la última media de cada módulo alterado					X	4,0					
28		Número de errores detectados (cubiertos externamente) de cada módulo					X	4,0					
29	Número de errores detectados (cubiertos externamente) de cada módulo					X	4,0						
30	Ejecutar la intervención	Tiempo dedicado a la tarea					X	5,0					
31		Número de errores detectados					X	4,0					
32	Ejecutar las pruebas de integración	Tiempo dedicado a la tarea					X	5,0					
33		Número de errores detectados					X	4,0					
34	Ejecutar las pruebas de aceptación (software antiguo y software nuevo)	Tiempo dedicado a la tarea					X	5,0					
35		Número de errores detectados					X	4,0					
36	Validar y validar corrección con el cliente	Tiempo dedicado a la tarea					X	5,0					
37		Tiempo dedicado a la tarea					X	5,0					
38	Redocumentar manual de usuario	Número de páginas añadidas/borradas/modificadas al manual					X	4,0					
39		Tiempo dedicado a la tarea					X	5,0					
40	Realizar revisión	Tiempo dedicado a la tarea (Esta tarea no se realiza para el Mantenimiento Preventivo)					X	5,0					
41		Estimar los costos del mantenimiento del software. Definir la técnica utilizada					X	4,0					
ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	2	Identificar la técnica para estimar los costos de mantenimiento. Paramétrico – Matemático – Juicio de Experto. – Otra (cuál?)					X	4,0	4,3				
	3	Definir el costo de operación del Hardware donde va a funcionar el producto de Software					X	5,0					
	4	Documentar el costo total del mantenimiento del producto					X	4,0					
	5	Verificar el presupuesto de mantenimiento. Se documenta dentro del presupuesto. Tiene conocimiento de la aplicación. Tiene conocimiento en el uso de técnicas y herramientas. Tiene conocimiento en lenguaje de programación. Está involucrado cuando es en la empresa. Se encuentra involucrado. Resalta recomendaciones sobre la calidad de su trabajo					X	4,0					
PERSONAL DE MANTENIMIENTO	1	Analizar. Se evalúa o mejora el producto sin que el cliente reporte los defectos					X	5,0	5,0				
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Registrar observaciones					X	4,0	4,4				
	2	Validar/Probar de datos. Incorporar los servicios de la base de datos heredada					X	5,0					
	3	Ejecutar paralelo. Realizar operaciones reales sobre el software que se va a retirar y el software nuevo (o se reutiliza)					X	4,0					
	4	Notificar estado. Notificar al Cliente la estado del software					X	5,0					
	5	Entregar datos del sistema antiguo					X	4,0					
	6	Entrega del inventario y de la documentación. Entregar al cliente todos los productos software producidos o modificados					X	5,0					
	7	Tiempo de experiencia y formación. Enseñar las nuevas características del sistema al cliente					X	4,0					
	8	Cerrar definitivo del servicio. Finalizar la prestación de los servicios al Cliente					X	5,0					

Tabla 9. Evaluación final mantenimiento adaptativo –Empresa 1

EVALUACIÓN FINAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE												
MANTENIMIENTO ADAPTATIVO												
FASE	No.	TAREA	RESPUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN	
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado con registros					Se cumple plenamente con registros
			0	1	2	3	4	5				
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre solicitud de los servicios de mantenimiento						X	5,0	5,0		
	2	Preparar la propuesta de mantenimiento para el cliente						X	5,0			
	3	Elaborar el contrato de prestación de servicios de mantenimiento						X	5,0			
PLANIFICACIÓN DEL MANTENIMIENTO	1	Definir el cronograma de actividades						X	4,0	4,3		
	2	Adquirir conocimiento de la aplicación: Verificar el estado del software - Establecer la documentación, el código de los programas, referencias cruzadas y su funcionamiento						X	5,0			
	3	Desarrollar planes: Estimar recursos necesarios (tiempo y personal) - Iniciar el sistema de información - Definir los horizontes para registrar los defectos que se reportan, para la gestión de la configuración y recursos y para la gestión del conocimiento (Documentación, Lecciones aprendidas, entre otros)						X	5,0			
	4	Definir procedimientos de petición de modificación					X	4,0	4,0			
	5	Preparar los entornos de pruebas: Realizar copias del entorno software para el trabajo					X	4,0	4,0			
	6	Recibir la petición de modificación: Recibir petición y validarla					X	4,0	4,0			
	7	Decidir el tipo de mantenimiento requerido: Rechazar la petición, justificando la razón o aceptar la petición - Decidir las actividades de mantenimiento a aplicar					X	4,0	4,0			
PROCESO DE MANTENIMIENTO	1	Planificar el catastro						X	4,0	4,2		
	2	Realizar copia del producto software: Realiza una copia del producto software, las bases de datos y los ficheros relacionados					X	4,0	4,0			
	3	Ejecutar la intervención: Ejecutar las acciones necesarias para servir la petición de modificación conforme a la alternativa seleccionada				X			3,0			
	4	Ejecutar las pruebas unitarias: Comprobar la que la petición de modificación quede servida					X	4,0	4,0			
	5	Ejecutar las pruebas de integración: Comprobar la que los elementos software funcionan correctamente de forma conjunta					X	5,0	5,0			
	6	Ejecutar paralelamente en software antiguo y nuevo: Ejecuta acciones reales en el producto software antiguo y en el modificado para detectar y prevenir posibles errores de proceso					X	5,0	5,0			
	7	Verificar y validar corrección con el cliente					X	5,0	5,0			
	8	Redocumentar manual de usuario					X	4,0	4,0			
	9	Pasar a producción: Betatear el producto software modificado en el entorno de trabajo real					X	4,0	4,0			
	10	Realizar revisión: Funciona correctamente en el entorno de trabajo real					X	4,0	4,0			
	11	Archivar datos del producto software inicial: Realizar una copia de seguridad de todo lo relacionado con el producto software que se ha adaptado (programas, datos y documentos)					X	4,0	4,0			
MÉTRICAS DEL MANTENIMIENTO	1	Desarrollar Planes de mantenimiento	1	Tiempo dedicado a la tarea					X	5,0	4,2	BUENO
	2		Número de puntos-función					X	4,0			
	3		Número de módulos					X	4,0			
	4		Número de líneas de código				X		3,0			
	5		Complejidad ciclomática a media de cada módulo				X	X	4,0			
	6		Número de variables globales (variables externamente) de cada módulo				X		3,0			
	7		Número de rutinas públicas (variables externamente) de cada módulo				X		3,0			
	8		Peticiones aceptadas o rechazadas						X	5,0		
	9		Decidir tipo de mantenimiento	1	Causa de la petición: Cambios legales - Evolución del negocio - Mejora del proceso - Cambios en las políticas de negocio - Otro (cual?)				X	4,0		
	10			1	Área o división funcional que ha presentado la petición				X	4,0		
	11		Ejecutar las intervenciones	1	Tiempo dedicado a la tarea				X	4,0		
	12			1	Número de puntos-función afectados				X	4,0		
	13			1	Número de puntos-función modificados				X	4,0		
	14			1	Número de puntos-función borrados				X	4,0		
	15			1	Número de módulos afectados				X	4,0		
	16			1	Número de líneas de código borrados				X	4,0		
	17			1	Número de líneas de código borradas				X	4,0		
	18		1	Complejidad ciclomática a media de cada módulo alterado				X	4,0			
	19		1	Número de variables globales (variables externamente) de cada módulo				X	4,0			
20	1	Número de rutinas públicas (variables externamente) de cada módulo				X	4,0					
21	Ejecutar las pruebas de integración	1	Tiempo dedicado a la tarea				X	4,0				
22		1	Número de errores detectados				X	4,0				
23		1	Tiempo dedicado a la tarea				X	4,0				
24		1	Número de errores detectados				X	4,0				
25		1	Tiempo dedicado a la tarea				X	4,0				
26		1	Número de errores detectados				X	4,0				
27		1	Verificar y validar corrección con el cliente				X	4,0				
28	Redocumentar manual de usuario	1	Tiempo dedicado a la tarea				X	5,0				
29		1	Número de páginas añadidas/borradas/modificadas al manual				X	4,0				
30	Pasar a producción	1	Tiempo dedicado a la tarea				X	5,0				
31		1	Tiempo dedicado a la tarea				X	5,0				
ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	1	Estimar los costos del mantenimiento del software: Definir la técnica utilizada						X	5,0	4,5		
	2	Identificar la técnica para estimar los costos de mantenimiento: Paramétrico - Matemático - Juicio de Expertos - Otro (cual?)						X	5,0			
	3	Definir el costo de operación del Hardware donde va a funcionar el producto de Software						X	4,0			
	4	Desarrollar el costo total del mantenimiento: el producto						X	4,0			
PERSONAL DE MANTENIMIENTO	1	Verificar si el equipo de mantenimiento: Se encuentra dentro del desarrollo - Tiene conocimiento de la aplicación - Tiene conocimientos en el uso de técnicas y herramientas - Tiene conocimiento en lenguajes de programación - Está laboralmente estable en la empresa - Se encuentra motivado - Recibe retroalimentación sobre la calidad de su trabajo				X	X		4,0	4,0		
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Analizar: Se evolucionó o mejoró el producto sin que el cliente reporte los defectos				X	X		3,0	3,0		
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Registrar intervenciones					X	4,0	4,3			
	2	Actualizar base de datos: Incorporar las métricas a la base de datos histórica					X	4,0				
	3	Ejecutar planes: Realizar operaciones reales sobre el software que se va a retirar y el software nuevo (o su migración)					X	4,0				
	4	Notificar retirada: Notificar al Cliente la retirada del software					X	4,0				
	5	Almacenar datos del sistema antiguo					X	5,0				
	6	Entrega del inventario y de la documentación: Entregar al cliente todos los productos software generados y modificados					X	4,0				
	7	Traspaso de experiencia y formación: Enseñar las nuevas características del sistema al cliente					X	4,0				
	8	Cierre definitivo del servicio: Finalizar la prestación de los servicios al Cliente					X	5,0				

Empresa 2

Evaluación inicial

De igual manera que en la empresa anterior, se realizó una reunión inicial con el equipo de mantenimiento de la empresa, conformado por tres ingenieros de sistemas. Los dos investigadores llevaron a cabo una explicación de la tarea investigativa desarrollada, a fin de socializar las etapas que componen la guía de mantenimiento de software propuesta; pasando a aplicar la evaluación inicial del cumplimiento de la guía, la cual fue contestada por el equipo.

“Empresa 2”, presta el servicio de mantenimiento al software propio, el cual es desarrollado según la necesidad del cliente y en todas las áreas posibles de la empresa del sector público y privado.

A continuación, se detalla el resultado de la evaluación inicial del cumplimiento de la guía de mantenimiento de software de la empresa, para los tipos de mantenimiento en estudio:

- Mantenimiento Correctivo Urgente
- Mantenimiento Correctivo No Urgente, Perfectivo y Preventivo
- Mantenimiento Adaptativo

Tabla 10. Evaluación inicial mantenimiento correctivo urgente –Empresa 2

EVALUACIÓN INICIAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE												
MANTENIMIENTO CORRECTIVO URGENTE												
FASE	No.	TAREA	RESPUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN	
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado con registros					Se cumple plenamente con registros
			0	1	2	3	4	5				
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre solicitud de los servicios de mantenimiento				X			3,0	3,7	DEFICIENTE	
	2	Preparar la propuesta de mantenimiento para el cliente					X		4,0			
	3	Elaborar el contrato de prestación de servicios de mantenimiento					X		4,0			
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades				X			3,0	2,6		
	2	Adquirir conocimiento de la aplicación. Verificar el estado del software - Estudiar la documentación, el código de los programas, referencias cruzadas y su funcionamiento					X		4,0			
	3	Desarrollar planes: Estimar recursos necesarios (tiempo y personal) - Inventariar el sistema de información - Definir las herramientas para registrar los defectos que se reportan, para la gestión de la configuración y versiones y para la gestión del conocimiento (Documentación, Lecciones aprendidas, entre otros)			X				2,0			
	4	Definir procedimientos de petición de modificación				X			3,0			
	5	Preparar los entornos de pruebas: Realizar copias del entorno software para el trabajo			X				2,0			
	6	Recibir la petición de modificación: Recibir petición y redactarla			X				2,0			
PROCESO DE MANTENIMIENTO	1	Investigar y analizar las causas del error: Analizar la petición - Verificar el problema - Estudiar alternativas para implementar modificación - Listar elementos software a corregir					X		4,0	2,6		
	2	Realizar las acciones correctivas: Identificar las rutinas y bases de datos afectadas por la intervención			X				2,0			
	3	Diligenciar la documentación exigida: Documentar los cambios realizados			X				2,0			
	4	Ejecutar las pruebas unitarias: Comprobar la corrección de los cambios realizados				X			3,0			
	5	Pasar a producción el software: Enviar al entorno de producción el software corregido para su utilización			X				2,0			
MÉTRICAS DEL MANTENIMIENTO	1	Desarrollar planes	Tiempo dedicado a la tarea			X			2,0	1,6		DEFICIENTE
	2		Número de puntos-función			X			2,0			
	3		Número de módulos			X			2,0			
	4		Número de líneas de código			X			2,0			
	5		Complejidad ciclomática (media de cada módulo)			X			2,0			
	6		Número de variables globales (visibles externamente) de cada módulo			X			2,0			
	7		Número de rutinas públicas (visibles externamente) de cada módulo			X			2,0			
	8	Decidir tipo de mantenimiento	Petición aceptada o rechazada			X			2,0			
	9		Causa de la petición: Cambios legales - Evolución del negocio - Mejora del proceso - Cambio en las políticas de negocio - Otra (causa)		X				1,0			
	10		Área o dominio funcional que ha presentado la petición			X			2,0			
	11	Realizar acciones correctivas	Tiempo dedicado a la tarea		X				1,0			
	12		Número de puntos-función que previsiblemente se verán afectados por el error				X		3,0			
	13		Origen del error: Cambio en el análisis de requisitos - Cambio en el análisis de localización - Cambio en el análisis de diseño - Codificación				X		3,0			
14	Investigar y analizar causas		Causa del error: Desconocimiento del dominio de la aplicación - Desconocimiento del diseño o la implementación del Sistema - Requisitos ambiguos o incompletos - Error sintáctico o semántico del lenguaje de programación - Tensión en la planificación - Falta existente (análisis) - Descuido.			X			2,0			
15			Tiempo dedicado a la tarea		X				2,0			
16	Realizar acciones correctivas		Número de puntos-función añadidos		X				1,0			
17		Número de puntos-función modificados		X				1,0				
18		Número de puntos-función borrados		X				1,0				
19		Número de módulos alterados		X				1,0				
20		Número de líneas de código añadidas		X				1,0				
21		Número de líneas de código modificadas		X				1,0				
22		Número de líneas de código borradas		X				1,0				
23		Número de defectos corregidos al producto por mes		X				1,0				
24		Complejidad ciclomática (media de cada módulo alterado)		X				1,0				
25		Número de variables globales (visibles externamente) de cada módulo		X				1,0				
26		Número de rutinas públicas (visibles externamente) de cada módulo		X				1,0				
27		Deficiencia de la calidad del código		X				1,0				
28		Índice de madurez del software		X				1,0				
29		Análisis de impacto de la corrección		X				2,0				
30		Número de errores de software encontrados		X				2,0				
31		Cada cuánto perfecciona el producto software		X				2,0				
32		Diligenciar documentación	Tiempo dedicado a la tarea		X				2,0			
33	Número de páginas añadidas a la documentación, según el estándar de la Organización				X			2,0				
34	Ejecutar las pruebas unitarias	Tiempo dedicado a la tarea			X			2,0				
35		Tiempo dedicado a la tarea			X			2,0				
ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	1	Estimar los costos del mantenimiento del software: Definir la técnica utilizada			X			2,0	1,8			
	2	Identificar la técnica para estimar los costos de mantenimiento: Paramétrico - Matemático - Juicio de Experto - Otro (Cuál?)		X				1,0				
	3	Definir el costo de operación del Hardware donde va a funcionar el producto de Software			X			2,0				
	4	Determinar el costo total del mantenimiento al producto			X			2,0				
PERSONAL DE MANTENIMIENTO	1	Verificar si el equipo de mantenimiento: Se encuentra dentro del desarrollo - Tiene conocimiento de la aplicación - Tiene conocimientos en el uso de técnicas y herramientas - Tiene conocimientos en lenguajes de programación - Está laboralmente estable en la empresa - Se encuentra motivado - Recibe retroalimentación sobre la calidad de su trabajo					X	4,0	4,0			
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Analizar: Se evalúa o mejora el producto sin que el cliente reporte los defectos				X		3,0	3,0			
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Registrar intervención			X				2,0	1,5		
	2	Actualizar base de datos: Incorporar las métricas a la base de datos histórica		X				1,0				
	3	Ejecutar pruebas: Realizar operaciones reales sobre el software que se va a retirar y el software nuevo (si se sustituye)		X				1,0				
	4	Notificar retirada: Notificar al Cliente la retirada del software		X	X			2,0				
	5	Actualizar datos del nuevo sistema		X				1,0				
	6	Entrega del inventario y de la documentación: Entregar al cliente todos los productos software generados y modificados		X				1,0				
	7	Traspaso de experiencia y formación: Enseñar las nuevas características del sistema al cliente			X			2,0				
	8	Creación definitiva del servicio: Finalizar la prestación de los servicios al Cliente			X			2,0				

Tabla 11. Evaluación inicial mantenimiento correctivo no urgente, perfectivo y preventivo –Empresa 2

EVALUACIÓN INICIAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE												
MANTENIMIENTO CORRECTIVO NO URGENTE - PERFECTIVO - PREVENTIVO												
FASE	No.	TAREA	RESPUESTA						PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN
			No sabe	No se cumple	Se cumple un registro	Se cumple con algunos registros	Se cumple en alto grado con registros	Se cumple plenamente con registros				
			0	1	2	3	4	5				
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre actividad de los servicios de mantenimiento						X	5,0	4,3		
	2	Preparar la propuesta de mantenimiento para el cliente						X	4,0			
	3	Elaborar el contrato de prestación de servicios de mantenimiento						X	4,0			
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades				X			3,0	2,9		
	2	Adaptar conocimiento de la aplicación. Verificar el estado del software - Estudiar la documentación, el código de los programas, referencias cruzadas y su funcionamiento						X	4,0			
	3	Desarrollar planes: Estimar recursos necesarios (tiempo y personal) - Inventariar el sistema de información - Definir los herramientas para registrar los defectos que se reportan, para la gestión de la configuración y versiones y para la gestión del conocimiento (Documentación, Lecciones aprendidas, entre otras)						X	4,0			
	4	Definir procedimientos de petición de modificación				X			3,0			
	5	Preparar los entornos de pruebas: Realizar copias del sistema software para el trabajo			X				2,0			
	6	Recibir la petición de modificación: Recibir petición y radicarla			X				2,0			
	7	Decidir el tipo de mantenimiento requerido: Revisar la petición, justificando la razón o aceptar la petición - Decidir las actividades de mantenimiento a realizar			X				2,0			
PROCESO DE MANTENIMIENTO	1	Validar la petición: Analizar las características de la petición - Verificar el problema - Listar elementos software involucrados			X				2,0	2,6		
	2	Documentar las posibles soluciones: Documentar la causa del error - Estudiar alternativas para implementar la modificación				X			3,0			
	3	Eligir la alternativa adecuada (Esta tarea no se realiza para el Mantenimiento Preventivo)			X				2,0			
	4	Ejecutar la intervención: Ejecutar las acciones necesarias para servir la petición de modificación conforme a la alternativa seleccionada				X			3,0			
	5	Ejecutar los pruebas unitarias: Comprobar que la petición de modificación quede servida				X			3,0			
	6	Ejecutar las pruebas de integración: Comprobar que los elementos software funcionan correctamente de forma conjunta				X			3,0			
	7	Ejecutar paralelamente en software antiguo y nuevo: Ejecutar acciones reales en el producto software antiguo y en el modificado para detectar y prevenir posibles errores de proceso				X			3,0			
	8	Verificar y validar corrección con el cliente				X			2,0			
	9	Recomendar manual de usuario				X			2,0			
	10	Pasar a producción: Transferir el producto software modificado en el entorno de trabajo real				X			3,0			
	11	Realizar revisión (Esta tarea no se realiza para el Mantenimiento Preventivo): Comprobar que el producto modificado funciona correctamente en el entorno de trabajo real				X			3,0			
MÉTRICAS DEL MANTENIMIENTO	1	Documentar planes				X			2,0	2,8	DEFICIENTE	
	2	Número de planes documentados				X			2,0			
	3	Número de errores detectados				X			2,0			
	4	Número de horas de código				X			2,0			
	5	Completitud ciclística media de cada módulo				X			2,0			
	6	Número de errores detectados por cada módulo				X			2,0			
	7	Número de errores por cada cambio experimental de cada módulo				X			2,0			
	8	Peticiones aceptadas o rechazadas				X			2,0			
	9	Causa de la petición: Cambios legales - Evolución del negocio - Mejora del proceso - Cambios en las políticas de negocio - Otros (cau?)		X					1,0			
	10	Área o dominio funcional que ha presentado la petición				X			2,0			
	11	Tiempo dedicado a la tarea				X			2,0			
	12	Fecha estimada para el inicio de la intervención				X			2,0			
	13	Fecha estimada para la finalización de la intervención				X			3,0			
	14	Tiempo dedicado a la tarea				X			2,0			
15	Número de puntos de función que previamente se verían afectados por el error				X			2,0				
16	Origen del error: Cambio en el análisis de requisitos - Cambio en el análisis de localización - Cambio en el análisis de diseño - Codificación					X		3,0				
MÉTRICAS DEL MANTENIMIENTO	17	Causa del error: Desconocimiento del dominio de la aplicación - Desconocimiento del diseño o la implementación del Sistema - Requisitos ambiguos o incompletos - Error sintáctico o semántico del lenguaje de programación - Tránsito en la planificación - Fallo cuando ensayando - Descuido				X			2,0	2,0		
	18	Eligir la alternativa adecuada				X			2,0			
	19	Tiempo dedicado a la tarea (Esta tarea no se realiza para el Mantenimiento Preventivo)				X			2,0			
	20	Tiempo dedicado a la tarea				X			2,0			
	21	Número de puntos función afectados				X			2,0			
	22	Número de puntos función modificados				X			2,0			
	23	Número de errores detectados				X			2,0			
	24	Número de errores por cada módulo				X			2,0			
	25	Número de errores por cada cambio experimental de cada módulo				X			2,0			
	26	Número de horas de código modificadas				X			2,0			
	27	Número de horas de código afectados				X			2,0			
	28	Completitud ciclística media de cada módulo alterado				X			2,0			
	29	Número de errores por cada cambio experimental de cada módulo				X			2,0			
	30	Número de errores por cada cambio experimental de cada módulo				X			2,0			
ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	31	Tiempo dedicado a la tarea				X			2,0	1,8		
	32	Número de errores detectados				X			2,0			
	33	Tiempo dedicado a la tarea				X			2,0			
	34	Número de errores detectados				X			2,0			
	35	Tiempo dedicado a la tarea				X			2,0			
	36	Número de errores detectados				X			2,0			
	37	Tiempo dedicado a la tarea				X			2,0			
	38	Número de páginas añadidas/borradas/modificadas al manual				X			2,0			
	39	Tiempo dedicado a la tarea				X			2,0			
	40	Tiempo dedicado a la tarea (Esta tarea no se realiza para el Mantenimiento Preventivo)				X			2,0			
PERSONAL DE MANTENIMIENTO	1	Estimar los costos del mantenimiento del software - Definir la técnica utilizada				X			2,0	4,0		
	2	Identificar la técnica para estimar los costos de mantenimiento: Paramétrico - Matemático - Juicio de Experto - Otro (Cau?)			X			1,0				
	3	Definir el costo de operación del hardware donde se va a funcionar el producto de Software				X			2,0			
	4	Determinar el costo total del mantenimiento al producto				X			2,0			
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Verificar el equipo de mantenimiento: Se encuentra dentro del desarrollo - Tiene conocimiento de la aplicación - Tiene conocimiento en el uso de técnicas y herramientas - Tiene conocimiento en lenguaje de programación - Está laboralmente estable en la empresa - Se encuentra motivado - Recibe retroalimentación sobre la calidad de su trabajo					X		4,0	3,0		
	2	Analizar: Se evoluciona o mejora el producto sin que el cliente reporte los defectos				X		3,0				
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Regular intervención				X			2,0	1,5		
	2	Actualizar base de datos: Recuperar los metadatos a la base de datos histórica			X			1,0				
	3	Ejecutar pruebas: Realizar operaciones reales sobre el software que se va a instalar y el software nuevo (o se sustituye)			X			1,0				
	4	Revisar evidencia: Notificar al Cliente la totalidad del software			X			2,0				
	5	Almacenar datos del sistema antiguo			X			1,0				
	6	Entrega del inventario y de la documentación: Entregar al cliente todos los productos software generables y modificados			X			1,0				
	7	Traspaso de experiencia y formación: Entregar las nuevas características del sistema al cliente				X		2,0				
	8	Cierre definitivo del servicio: Finalizar la prestación de los servicios al Cliente				X		2,0				

Tabla 12. Evaluación inicial mantenimiento adaptativo –Empresa 2

EVALUACIÓN INICIAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE											
MANTENIMIENTO ADAPTATIVO											
FASE	Nº	TAREA	RESPUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado con registros				
			0	1	2	3	4	5			
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre utilidad de los servicios de mantenimiento						X	5,0	4,3	DEFICIENTE
	2	Preparar la propuesta de mantenimiento para el cliente					X	4,0			
	3	Elaborar el contrato de prestación de servicios de mantenimiento					X	4,0			
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades				X		3,0	2,9		
	2	Adaptar conocimientos de la aplicación. Verificar el estado del software - Evaluar la documentación, el código de los programas, reformas creadas y su funcionamiento					X	4,0			
	3	Desarrollar planes. Estimar recursos necesarios (tiempo y personal) - Inventariar el sistema de información - Definir las herramientas para registrar los defectos que se reportan, para la gestión de la configuración y versiones y para la gestión del conocimiento (Documentación, Lecturas aprendidas, entre otras)					X	4,0			
	4	Definir procedimientos de petición de modificación			X			3,0			
	5	Preparar los entornos de pruebas. Realizar copia del entorno software para el trabajo		X				2,0			
	6	Recibir la petición de modificación. Recibir petición y redactarla		X				2,0			
	7	Decidir el tipo de mantenimiento requerido. Rechazar la petición, justificando la razón o aceptar la petición - Decidir las actividades de mantenimiento a aplicar		X				2,0			
PROCESO DE MANTENIMIENTO	1	Planificar el calendario			X			3,0	2,8		
	2	Realizar copia del producto software: Realiza una copia del producto software, las bases de datos y los ficheros relacionados				X		3,0			
	3	Ejecutar la intervención. Ejecutar las acciones necesarias para servir la petición de modificación conforme a la alternativa seleccionada		X				2,0			
	4	Ejecutar las pruebas unitarias. Comprobar que la petición de modificación quede servida		X				2,0			
	5	Ejecutar las pruebas de integración. Comprobar que los elementos software funcionan correctamente de forma conjunta			X			3,0			
	6	Ejecutar paralelamente en software antiguo y nuevo. Ejecuta acciones reales en el producto software antiguo y en el modificado para detectar y prevenir posibles errores de proceso			X			3,0			
	7	Verificar y validar corrección con el cliente			X			3,0			
	8	Rehebrar manual de usuario		X				2,0			
	9	Pasar a producción. Instalar el producto software modificado en el entorno de trabajo real			X			3,0			
	10	Realizar revisión. Funciona correctamente en el entorno de trabajo real			X			3,0			
	11	Archivar datos del producto software inicial. Realizar una copia de seguridad de todo lo relacionado con el producto software que se ha adaptado (programas, datos y documentos)					X	4,0			
MÉTRICAS DEL MANTENIMIENTO	1	Tiempo dedicado a la tarea			X			2,0	2,7		
	2	Número de errores detectados			X			2,0			
	3	Número de líneas de código			X			2,0			
	4	Complejidad ciclomática media de cada módulo		X				2,0			
	5	Número de variables globales/locales externamente de cada módulo		X				2,0			
	6	Número de rutinas públicas/privadas externamente de cada módulo		X				2,0			
	7	Petición aceptada o rechazada		X				2,0			
	8	Costo de la petición. Cambios logados - Evolución del negocio - Mejora del proceso - Cambios en las políticas de negocio - Otro (cost?)		X				2,0			
	9	Área o dominio funcional que ha presentado la petición		X				2,0			
	10	Tiempo dedicado a la tarea		X				2,0			
	11	Número de pruebas-función añadidas		X				2,0			
	12	Número de pruebas-función modificadas		X				2,0			
	13	Número de pruebas-función borradas		X				2,0			
	14	Número de errores detectados		X				2,0			
	15	Número de líneas de código añadidas		X				2,0			
	16	Número de líneas de código borradas		X				2,0			
	17	Número de líneas de código heredadas		X				2,0			
	18	Complejidad ciclomática media de cada módulo alterado		X				2,0			
	19	Número de variables globales/locales externamente de cada módulo		X				2,0			
	20	Número de rutinas públicas/privadas externamente de cada módulo		X				2,0			
	21	Tiempo dedicado a la tarea		X				2,0			
	22	Número de errores detectados		X				2,0			
	23	Tiempo dedicado a la tarea		X				2,0			
	24	Número de errores detectados		X				2,0			
	25	Tiempo dedicado a la tarea		X				2,0			
ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	1	Estimar los costos del mantenimiento del software. Definir la técnica utilizada		X				2,0	1,8		
	2	Identificar la técnica para estimar los costos de mantenimiento: Paramétrico - Matemático - Juicio de Experto - Otro (Cost?)		X				1,0			
	3	Definir el costo de operación del Hardware donde va a funcionar el producto de Software		X				2,0			
	4	Determinar el costo total del mantenimiento al producto		X				2,0			
PERSONAL DE MANTENIMIENTO	1	Verificar el equipo de mantenimiento. Se encuentra dentro del desarrollo - Tiene conocimientos de la aplicación - Tiene conocimientos en el uso de técnicas y herramientas - Tiene conocimientos en lenguajes de programación - Está laboralmente estable en la empresa - Se encuentra motivado - Recibe retroalimentación sobre la calidad de su trabajo				X		3,0	3,0		
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Analizar. Se evalúa o mejora el producto sin que el cliente reporte los defectos				X		3,0	3,0		
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Registrar intervención		X				2,0	1,5		
	2	Actualizar base de datos. Incorporar los servicios a la base de datos heredada		X				1,0			
	3	Ejecutar paralelo. Realizar operaciones reales sobre el software que se va a retirar y el software nuevo (o se sustituye)		X				1,0			
	4	Revisar resultado. Notificar al Cliente la entrega del software		X	X			2,0			
	5	Almacenar datos del entorno antiguo		X				1,0			
	6	Entrega del inventario y de la documentación. Entregar al cliente todos los productos software generados y modificados		X				1,0			
	7	Traspaso de experiencia y formación. Enseñar las nuevas características del sistema al cliente			X			2,0			
	8	Creado definitiva del servicio. Finalizar la prestación de los servicios al Cliente			X			2,0			

Con la evaluación inicial realizada, se pudo observar que la información que se obtiene para realizar el mantenimiento no es estándar para todas las peticiones que se reciben. Este proceso está sujeto a la persona que lo reciba para levantar la información y algunas veces la información queda incompleta, por lo que la persona debe volver a contactar al cliente para obtener la información completa. Al momento de guardar la información, solo se guarda el acta final y las conclusiones, mas no las evidencias y todo lo que se recolecta para levantar la información.

En la fase de planificación de mantenimiento, la empresa no tiene registros históricos sobre la estimación que realizan para una solicitud de mantenimiento, no realizan un registro de defectos reportados por el cliente o defectos que se encuentran al realizar las pruebas del software. No cuentan con un sistema para gestión de la configuración y tampoco del conocimiento.

Por otro lado, no se tienen entornos de pruebas. las pruebas las realiza cada desarrollador en su pc; las peticiones que se reciben, no todas quedan radicadas y guardadas en el mismo lugar, algunas quedan en el pc de la persona que la recibió y otras se pierden.

En la fase del proceso de mantenimiento correctivo, la empresa realiza un buen proceso de análisis para saber las causas del error reportado y así poder identificar la mejor solución al problema. Tiene personal con amplio conocimiento en todos los desarrollos que realizan a los diferentes clientes. Pero este conocimiento se queda en ellos, por lo cual, si hay algo que ellos no sepan, es muy difícil que alguien más lo sepa. Por esta razón, en muchas ocasiones se hace complicado identificar los componentes de software y/o base de datos que impactan la solicitud de mantenimiento, pues no se tiene registro de lo que se realizó anteriormente. Además, no siempre se realiza la documentación de recepción y entrega de la solicitud. Con respecto a las métricas, la empresa manifiesta que los tiempos incurridos en las tareas si se tienen en cuenta para estimaciones y productividad del personal, sin embargo, no se pudo comprobar los registros guardados. Además, no cuentan con un sistema que les permita medir la calidad del código generado.

Para la fase del proceso de mantenimiento para el mantenimiento no urgente, preventivo, perfectivo y adaptativo, para verificar el problema y saber que elementos se ven afectados es difícil porque no se tiene una trazabilidad completa de los componentes y de las posteriores modificaciones que han tenido. Este estudio se hace mediante la experiencia del personal y analizando el código desarrollado para poder identificar las acciones que se deben realizar. Cuando se realizan las pruebas de la petición, no quedan evidencias de éstas y, según nos cuentan, han tenido problemas posteriores con el cliente porque no tienen cómo garantizar que la petición se entregó funcionando. Muy pocas veces se actualiza la documentación del sistema modificado, lo cual conlleva a que si en alguna ocasión se revisan los documentos que hallan, es muy probable que estén desactualizados.

Para la estimación de costos, al no tener registros de lo que se ha realizado anteriormente, la estimación la realizan los ingenieros que más experiencia tienen. Para el costo de la operación, no saben cuánto es el costo de la energía que consumen los computadores, los servidores y solo se tienen guardadas las facturas de algunas licencias compradas, pero no las tienen en cuenta para la estimación.

La empresa cuenta con más analistas y desarrolladores con experiencia que sin experiencia. Es un grupo pequeño de ingenieros pero que llevan mucho tiempo trabajando juntos y se conocen su trabajo. El trabajo que requiere mayor esfuerzo lo realizan los de más experiencia, de allí que, al llegar muchas solicitudes de mantenimiento, en ocasiones no tienen quien más ayude en esta labor y les toca apoyarse en las personas nuevas con supervisión de alguien. Al recibir solicitudes de mantenimiento, no tienen registros históricos de quién realiza qué funcionalidad y no saben a quién acudir para preguntar algo, por lo tanto, la asignación de la solicitud es a quién esté disponible.

Para la evolución del sistema, no se tienen estrategias o mecanismos para que el software tenga nuevas funcionalidades o mejorar las que ya se tienen. Algunas que han surgido ha sido producto de las pruebas realizadas o ideas que han surgido del mismo personal, pero no quedan documentadas y con el tiempo se pierden.

En la fase de actividades finales del mantenimiento, la empresa genera un documento con la funcionalidad nueva o modificada, las acciones que se deben realizar para instalar lo nuevo si aplica y las nuevas características que se adicionaron. Sin embargo, este documento es un acta de entrega mas no un manual de usuario o un manual que sirva como evidencia del detalle de la intervención realizada. Por lo tanto, para una próxima modificación, este no serviría como base para saber al detalle que fue lo que se realizó previamente y del traspaso del conocimiento hacia el cliente. La empresa no actualiza los históricos de las métricas obtenidas del proceso, no realiza pruebas en paralelo con la versión anterior del software, no guarda la configuración o el entorno de cómo funcionaba antes el sistema por si se debe regresar a un punto de éstos.

Proceso de aplicación y Evaluación Final

Para realizar la evaluación final, se realizaron 3 jornadas de capacitación con el equipo de mantenimiento de la empresa, durante las cuales se explicaron cada una de las actividades que hacen parte de las ocho fases de la guía de mantenimiento de software propuesta por los dos investigadores.

Una vez capacitados los tres integrantes del equipo de mantenimiento en el cumplimiento de la guía, se llevaron a cabo dos visitas para aplicar la evaluación final para los mantenimientos correctivo urgente, correctivo no urgente – perfectivo – preventivo y adaptativo. Para esta tarea, los investigadores interrogaron sobre cada una de las actividades verificando la respuesta obtenida, es decir, que se cumplieran los registros respectivos.

Teniendo en cuenta la evaluación inicial y la capacitación dada durante las jornadas, el equipo decidió realizar algunas mejoras a su proceso de mantenimiento de software.

Los hallazgos fueron los siguientes:

En la fase del proceso de mantenimiento, comenzaron a utilizar una herramienta para el cronograma de actividades, paralela a como lo hacen ahora que es a través de un archivo en Excel. La estimación de los recursos va a comenzar a utilizar las métricas que se están comenzando a recopilar y compararlas con el juicio de expertos que ahora tienen. Tienen pendiente crear un blog o algún recurso para la gestión del conocimiento, pues vieron la necesidad de tener documentación técnica y de negocio para las personas nuevas que se incorporen al equipo y también para las que ya estén y no sepan de algún tema en particular. Ajustaron el procedimiento de petición de modificación donde son más detallistas al momento de recibir la petición, pues ya se toman un poco más de tiempo para analizarla, documentarla, radicarla en un repositorio en la empresa y así definir el tipo de mantenimiento y decidir las actividades que ésta requiere.

Al comenzar a documentar más cada petición de mantenimiento, se les está haciendo más fácil identificar los componentes que intervienen en el ajuste a realizar. Para ello se les recomendó llevar la trazabilidad en un archivo Excel para que luego fuera más fácil identificarlos.

El paso a producción lo mejoraron, entregando más detalle al cliente con manuales de operación, instalación y un acta donde se describe los componentes que se modificaron y las nuevas versiones de los componentes que se les entregan a ellos.

A pesar de que ya contaban con una herramienta para registrar los defectos que se iban reportando, no tenían en cuenta las métricas que éste arrojaba. Por lo tanto, se recomendó ver la forma de generar reportes por medio de la herramienta y obtener información histórica que pueda servir para posteriores estimaciones y aspectos por mejorar de todo el equipo. Además, se recomendó utilizar una herramienta para medir la calidad de código llamada Sonar.

Para la estimación de los costes del mantenimiento comenzaron para tener en cuenta los costes de las licencias que necesitaban para el desarrollo, para las bases de datos, y la energía que consumen todos los equipos y los servidores involucrados. Al momento de

realizar la evaluación final, estaban apenas estaban obteniendo la información por lo cual no la habían incluido en los presupuestos. Además, gracias a las métricas que se están revisando, han ido ajustando los costes del mantenimiento.

Dadas las mejoras que se han ido implementando, el personal del equipo ha adquirido mayor conocimiento del negocio, de las modificaciones que todos realizan lo cual conlleva a que no necesariamente una persona deba responder por lo que hace, sino que cualquiera pueda dar soporte a las diferentes aplicaciones

Con respecto a la evolución y mejora de las aplicaciones, implementaron una nueva estrategia de dar incentivo a las personas que hagan aportes para mejorarlas. Sin embargo, hasta el momento de la evaluación final, no se había presentado ninguna propuesta por parte del personal y los líderes tampoco habían vuelto a recordar ni hacer énfasis en la nueva estrategia.

Y, por último, al ver la necesidad de tener que documentar todo lo que realizan, se mejoró el registro de la intervención por medio de los documentos mencionados anteriormente, las métricas obtenidas de los reportes se están almacenando en el repositorio mensualmente, se comenzaron a realizar pruebas en paralelo con el software nuevo y viejo si aplica para una sustitución de componentes. Almacenan la configuración usada para las correcciones y pruebas, pues es posible que otro miembro del equipo deba continuar y así se garantiza que tiene la misma configuración. El nuevo conocimiento adquirido, por el momento, se transmite al resto del equipo de forma verbal, mientras se define el mecanismo para esta gestión; y para el cliente se realiza a través del documento de operación y de instalación que se comenzó a realizar.

Al igual que en la otra empresa, los resultados fueron muy gratificantes para ambas partes, pues los puntajes obtenidos fueron mayores que los obtenidos en las evaluaciones iniciales. A continuación, se muestran los resultados obtenidos:

Tabla 13. Evaluación final mantenimiento correctivo urgente –Empresa 2

EVALUACIÓN FINAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE																
MANTENIMIENTO CORRECTIVO URGENTE																
FASE	No.	TAREA	RESPUESTA						PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN				
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado con registros	Se cumple plenamente con registros								
			0	1	2	3	4	5								
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre solicitud de los servicios de mantenimiento				X			3,0	3,7						
	2	Preparar la propuesta de mantenimiento para el cliente					X		4,0							
	3	Elaborar el contrato de prestación de servicios de mantenimiento					X		4,0							
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades					X		4,0	4,0						
	2	Adquirir conocimiento de la aplicación. Verificar el estado del software - Estudiar la documentación, el código de los programas, referencias cruzadas y su funcionamiento					X		4,0							
	3	Desarrollar planes. Estimar recursos necesarios (tiempo y personal) - Inventariar el sistema de información - Definir las herramientas para registrar los defectos que se reportan, para la gestión de la configuración y versiones y para la gestión del conocimiento (Documentación, Lecciones aprendidas, entre otras)				X			3,0							
	4	Definir procedimientos de petición de modificación					X		4,0							
	5	Preparar los entornos de pruebas. Realizar copia del entorno software para el trabajo					X		4,0							
	6	Recibir la petición de modificación. Recibir petición y validarla					X		4,0							
	7	Decidir el tipo de mantenimiento requerido. Rechazar la petición, justificando la razón o aceptar la petición - Decidir las actividades de mantenimiento a aplicar						X	5,0							
PROCESO DE MANTENIMIENTO	1	Investigar y analizar las causas del error. Analizar la petición - Verificar el problema - Estudiar alternativas para implementar modificación - Listar elementos software a coverage				X			3,0	3,8						
	2	Realizar las acciones correctivas. Identificar las rutinas y bases de datos afectadas por la intervención					X		4,0							
	3	Diligenciar la documentación exigida. Documentar los cambios realizados				X			3,0							
	4	Ejecutar las pruebas unitarias. Comprobar la corrección de los cambios realizados						X	5,0							
	5	Pasar a producción el software. Evitar al entorno de producción el software corregido para su validación					X		4,0							
MÉTRICAS DEL MANTENIMIENTO	15	Desarrollar Planes de mantenimiento	Tiempo dedicado a la tarea				X		4,0	3,9						
	16		Número de puntos-función				X		4,0							
	17		Número de módulos				X		4,0							
	18		Número de puntos-función borrados				X		4,0							
	19		Número de módulos alterados				X		4,0							
	20		Número de líneas de código borradas				X		4,0							
	21		Número de líneas de código modificadas				X		4,0							
	22	Número de líneas de código borradas				X		4,0								
	23	Número de defectos corregidos al producto por mes				X		4,0								
	24	Complejidad ciclomática media de cada módulo alterado				X		4,0								
	25	Número de variables globales (visibles externamente) de cada módulo				X		4,0								
	26	Número de rutinas públicas (visibles externamente) de cada módulo				X		4,0								
	27	Definición de la calidad del código				X		4,0								
	28	Índice de madurez del software				X		4,0								
Realizar acciones correctivas	29	Análisis de impacto de la corrección				X		4,0	4,0							
	30	Número de retiros de software realizados				X		4,0								
	31	Cada cuánto perfeccionan el producto software				X		4,0								
	32	Tiempo dedicado a la tarea				X		4,0								
	33	Diligenciar documentación				X		3,0								
	34	Ejecutar las pruebas unitarias					X	4,0								
	35	Pasar a producción					X	4,0								
	ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	1	Estimar los costos del mantenimiento del software. Definir la técnica utilizada					X					4,0	3,5		
		2	Identificar la técnica para estimar los costos de mantenimiento Paramétrico - Matemático - Juicio de Expertos - Otra (Qual?)				X					3,0				
		3	Definir el costo de operación del Hardware donde va a funcionar el producto de Software					X				4,0				
4		Determinar el costo total del mantenimiento al producto				X		3,0								
PERSONAL DE MANTENIMIENTO	1	Verificar el equipo de mantenimiento. Se encuentra dentro del desarrollo - Tiene conocimiento de la aplicación - Tiene conocimientos en el uso de técnicas y herramientas - Tiene conocimiento en lenguajes de programación - Está laboralmente estable en la empresa - Se encuentra motivado - Recibe retroalimentación sobre la calidad de su trabajo						X	5,0	5,0						
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Analizar: Se evoluciona o mejora el producto sin que el cliente reporte los defectos					X		4,0	4,0						
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Registrar intervenciones				X			4,0	4,4						
	2	Actualizar base de datos. Incorporar las métricas a la base de datos histórica				X		4,0								
	3	Ejecutar paralelo: Realizar operaciones reales sobre el software que se va a retirar y el software nuevo (si se sustituye)					X	5,0								
	4	Notificar retirada. Notificar al Cliente la retirada del software				X		4,0								
	5	Almacenar datos del entorno antiguo				X		4,0								
	6	Entrega del inventario y de la documentación. Entregar al cliente todos los productos software generados y modificados					X	5,0								
	7	Traspaso de experiencia y formación. Enseñar las nuevas características del sistema al cliente				X		4,0								
	8	Cesión definitiva del servicio. Finalizar la prestación de los servicios al Cliente					X	5,0								

4,0

BUENO

Tabla 14. Evaluación final mantenimiento correctivo no urgente, perfecto y preventivo –Empresa 2

EVALUACIÓN FINAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE														
MANTENIMIENTO CORRECTIVO NO URGENTE - PERFECTIVO - PREVENTIVO														
FASE	No.	TAREA	RESPUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN			
			No sabe	No se cumple	Se cumple en registros	Se cumple con algunos registros	Se cumple en alto grado con registros					Se cumple plenamente con registros		
			0	1	2	3	4	5						
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre solicitud de los servicios de mantenimiento				X			3,0	4,3				
	2	Preparar la propuesta de mantenimiento para el cliente						X	5,0					
	3	Elaborar el contrato de prestación de servicios de mantenimiento							X			5,0		
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades					X		4,0	4,0				
	2	Adquirir conocimiento de la aplicación. Verificar el estado del software - Estudiar la documentación, el código de los programas, referencias cruzadas y su funcionamiento					X		4,0					
	3	Desarrollar planes. Estimar recursos necesarios (tiempo y personal) - Levantar el sistema de información - Definir las herramientas para reparar los defectos que se reportan, para la gestión de la configuración y versiones y para la gestión del conocimiento (Documentación, Lecciones aprendidas, entre otras)				X			3,0					
	4	Definir procedimientos de petición de modificación					X		4,0					
	5	Preparar los entornos de pruebas: Realizar copias del entorno software para el trabajo					X		4,0					
	6	Recibir la petición de modificación. Realizar petición y validarla					X		4,0					
	7	Decidir el tipo de mantenimiento requerido: Rechazar la petición, justificando la razón o aceptar la petición - Decidir las actividades de mantenimiento a aplicar						X	5,0					
PROCESO DE MANTENIMIENTO	1	Validar la petición. Analizar las características de la petición - Verificar el problema - Listar elementos software a corregir						X	5,0	4,4				
	2	Documentar las posibles soluciones: Documentar la causa del error - Estudiar alternativas para implementar la modificación						X	5,0					
	3	Elegir la alternativa adecuada (Esta tarea no se realiza para el Mantenimiento Preventivo)					X		4,0					
	4	Ejecutar la intervención: Ejecutar las acciones necesarias para servir la petición de modificación conforme a la alternativa seleccionada					X		4,0					
	5	Ejecutar las pruebas unitarias: Comprobar la que la petición de modificación quede servida					X		4,0					
	6	Ejecutar las pruebas de integración: Comprobar la que los elementos software funcionen correctamente de forma conjunta					X		4,0					
	7	Ejecutar paralelamente en software antiguo y nuevo: Ejecutar acciones reales en el producto software antiguo y en el modificado para detectar y prevenir posibles errores de proceso					X		4,0					
	8	Verificar y validar corrección con el cliente					X		4,0					
	9	Redimensionar manual de usuario						X	5,0					
	10	Pasar a producción: Instalar el producto software modificado en el entorno de trabajo real						X	5,0					
	11	Realizar revisión (Esta tarea no se realiza para el Mantenimiento Preventivo): Comprobar que el producto modificado funciona correctamente en el entorno de trabajo real					X		4,0					
MÉTRICAS DEL MANTENIMIENTO	1	Documentar Peticiones						X	5,0	4,1	BUENO			
	2	Tiempo dedicado a la tarea						X	4,0					
	3	Número de peticiones atendidas					X		4,0					
	4	Número de módulos					X		4,0					
	5	Número de líneas de código					X		4,0					
	6	Complejidad ciclomática media de cada módulo					X		4,0					
	7	Número de variables globales (variables externamente de cada módulo)					X		4,0					
	8	Número de rutinas públicas (variables externamente de cada módulo)					X		4,0					
	9	Peticiones aceptadas o rechazadas						X	5,0					
	10	Causa de la petición: Cambios legales - Evolución del negocio - Mejora del proceso: Cambio en las políticas de respecto - Otra (cuál?)					X		4,0					
	11	Área o dominio funcional que ha presentado la petición				X			3,0					
	12	Tiempo estimado para el inicio de la intervención					X	X	5,0					
	13	Fecha estimada para la finalización de la intervención				X			4,0					
	14	Tiempo dedicado a la tarea					X		3,0					
	15	Número de peticiones/funciones que presuntamente se venían afectadas por el error					X		4,0					
	16	Origen del error: Cambio en el análisis de requisitos - Cambio en el análisis de localización - Cambio en el análisis de diseño - Codificación					X		4,0					
	17	Causa del error: Desconocimiento del dominio de la aplicación - Desconocimiento del hecho o la implementación del Sistema - Requisitos ambiguos o incompatibles - Error sintáctico o semántico del lenguaje de programación - Términos en la planificación - Fallo existente descubierta - Descuido					X		4,0					
MÉTRICAS DEL MANTENIMIENTO	18	Documentar los posibles subproductos						X	4,0	4,1	BUENO			
	19	Tiempo dedicado a la tarea (Esta tarea no se realiza para el Mantenimiento Preventivo)					X		4,0					
	20	Tiempo dedicado a la tarea						X	5,0					
	21	Número de peticiones/funciones atendidas					X		5,0					
	22	Número de peticiones/funciones rechazadas					X		4,0					
	23	Número de peticiones/funciones rechazadas					X		4,0					
	24	Número de peticiones/funciones rechazadas					X		4,0					
	25	Número de líneas de código atendidas					X		4,0					
	26	Número de líneas de código rechazadas					X		4,0					
	27	Complejidad ciclomática media de cada módulo atendido					X		4,0					
	28	Complejidad ciclomática media de cada módulo rechazado					X		4,0					
	29	Número de variables globales (variables externamente de cada módulo)					X		4,0					
	30	Número de rutinas públicas (variables externamente de cada módulo)					X		4,0					
	31	Tiempo dedicado a la tarea					X		4,0					
	32	Número de errores detectados					X		4,0					
	ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	33	Documentar los posibles subproductos						X			4,0	3,5	
		34	Tiempo dedicado a la tarea					X				4,0		
35		Número de errores detectados					X		4,0					
36		Tiempo dedicado a la tarea					X		4,0					
37		Número de errores detectados					X		4,0					
38		Tiempo dedicado a la tarea					X		4,0					
39		Número de páginas añadidas/eliminadas/modificadas al manual					X		4,0					
40		Tiempo dedicado a la tarea					X		4,0					
41		Tiempo dedicado a la tarea (Esta tarea no se realiza para el Mantenimiento Preventivo)					X		4,0					
42		Tiempo dedicado a la tarea					X		4,0					
ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	1	Estimar los costos del mantenimiento del software: Definir la técnica utilizada					X		4,0	3,5				
	2	Identificar la técnica para estimar los costos de mantenimiento: Paramétrico - Matemático - Juicio de Experto - Otro (cuál?)				X		3,0						
	3	Definir el costo de operación del Hardware donde va a funcionar el producto de Software						X	4,0					
	4	Documentar el costo total del mantenimiento al producto				X		3,0						
PERSONAL DE MANTENIMIENTO	1	Verificar si el equipo de mantenimiento: Se encuentra dentro del desarrollo - Tiene conocimiento de la aplicación - Tiene conocimiento en el uso de técnicas y herramientas - Tiene conocimiento en lenguajes de programación - Está laboralmente estable en la empresa - Se encuentra motivado - Recibe retroalimentación sobre la calidad de su trabajo					X		4,0	4,0				
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Analizar: Se evolucionan o mejora el producto sin que el cliente reporte los defectos					X		4,0	4,0				
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Registrar intervenciones					X		4,0	4,3				
	2	Actualizar base de datos: Incorporar las métricas a la base de datos histórica				X		3,0						
	3	Ejecutar paralelamente: Realizar operaciones reales sobre el software que se va a retirar y el software nuevo (si se usará)					X		5,0					
	4	Notificar entrada: Notificar al Cliente la entrada del software					X		4,0					
	5	Actualizar datos: Actualizar el cronograma de trabajo					X		3,0					
	6	Entrega del inventario y de la documentación: Entregar al cliente todos los productos software generados y modificados					X		5,0					
	7	Tiempo de experiencia y formación: Enseñar las nuevas características del sistema al cliente					X		4,0					
	8	Cesión definitiva del servicio: Finalizar la prestación de los servicios al Cliente						X	5,0					

Tabla 15. Evaluación final mantenimiento adaptativo –Empresa 2

EVALUACIÓN FINAL DEL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE											
MANTENIMIENTO ADAPTATIVO											
FASE	Nº	TAREA	RESPUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE	INTERPRETACIÓN
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado con registros				
			0	1	2	3	4	5			
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre actividad de los servicios de mantenimiento					X	5,0	5,0		
	2	Preparar la propuesta de mantenimiento para el cliente					X	5,0			
	3	Elaborar el contrato de prestación de servicios de mantenimiento					X	5,0			
PLANIFICACIÓN DEL MANTENIMIENTO	4	Elaborar el cronograma de actividades					X	5,0	5,0		
	5	Adquirir conocimiento de la aplicación: Verificar el estado del software - Estudiar la documentación, el código de los programas, referencias cruzadas y su funcionamiento					X	5,0			
	6	Desarrollar planes: Estimar recursos necesarios (tiempo y personal) - Inventariar el sistema de información - Definir los herramientas para registrar los defectos que se reportan, para la gestión de la configuración y versiones y para la gestión del conocimiento (Documentación, Lecturas, agendas, entre otras)					X	5,0			
	7	Definir procedimientos de petición de modificación					X	5,0			
	8	Preparar los entornos de prueba: Realizar copias del entorno software para el trabajo					X	5,0			
	9	Recibir la petición de modificación: Recibir petición y radicarla					X	5,0			
	10	Decidir el tipo de mantenimiento requerido: Rechazar la petición, justificando la razón o aceptar la petición - Decidir las actividades de mantenimiento a aplicar					X	5,0			
PROCESO DE MANTENIMIENTO	1	Planificar el calendario					X	4,0	4,1		
	2	Realizar copia del producto software: Realiza una copia del producto software, las bases de datos y los ficheros relacionados					X	4,0			
	3	Ejecutar la intervención: Ejecutar las acciones necesarias para servir la petición de modificación conforme a la alternativa seleccionada					X	5,0			
	4	Ejecutar las pruebas unitarias: Comprobar la que la petición de modificación quede servida					X	4,0			
	5	Ejecutar las pruebas de integración: Comprobar la que los elementos software funcionan correctamente de forma conjunta					X	4,0			
	6	Ejecutar paralelamente en software antiguo y nuevo: Ejecutar acciones reales en el producto software antiguo y en el modificado para detectar y prevenir posibles errores de proceso					X	4,0			
	7	Verificar y validar corrección con el cliente					X	4,0			
	8	Reubicar manual de usuario					X	4,0			
	9	Pasar a producción: Instala el producto software modificado en el entorno de trabajo real					X	4,0			
	10	Realizar revisiones: Ejecuta correctamente en el entorno de trabajo real					X	4,0			
	11	Archivar datos del producto software inicial: Realizar una copia de seguridad de todo lo relacionado con el producto software que se ha adaptado (programas, datos y documentos)					X	4,0			
MÉTRICAS DEL MANTENIMIENTO	1	Desarrollar planes					X	5,0	4,3	BUENO	
	2	Número de pautas firmadas					X	5,0			
	3	Número de líneas de código					X	5,0			
	4	Completitud del cambio a nivel de cada módulo			X		X	5,0			
	5	Número de variables definidas (variables externamente de cada módulo)				X	X	4,0			
	6	Número de variables definidas (variables externamente de cada módulo)				X	X	4,0			
	7	Número de rutinas públicas (variables externamente de cada módulo)				X	X	4,0			
	8	Petición aceptada o rechazada					X	5,0			
	9	Causa de la petición: Cambios legales - Evolución del negocio - Mejora del proceso - Cambio en las políticas de negocio - Otra (cau?)					X	4,0			
	10	Área o dominio funcional que ha presentado la petición					X	4,0			
	11	Espero dedicado a la tarea					X	4,0			
	12	Número de pautas firmadas					X	4,0			
	13	Número de pautas-función modificadas					X	4,0			
	14	Número de pautas-función borradas					X	4,0			
	15	Número de módulos alterados					X	4,0			
	16	Ejecutar la intervención					X	4,0			
	17	Número de líneas de código alteradas					X	4,0			
	18	Número de líneas de código modificadas					X	4,0			
	19	Número de líneas de código borradas					X	4,0			
20	Completitud del cambio a nivel de cada módulo alterado					X	4,0				
21	Número de variables definidas (variables externamente de cada módulo)					X	4,0				
22	Número de variables definidas (variables externamente de cada módulo)					X	4,0				
23	Número de rutinas públicas (variables externamente de cada módulo)					X	4,0				
24	Espero dedicado a la tarea					X	5,0				
25	Número de errores detectados					X	5,0				
26	Espero dedicado a la tarea					X	5,0				
27	Número de errores detectados					X	5,0				
28	Espero dedicado a la tarea					X	5,0				
29	Número de errores detectados					X	5,0				
30	Espero dedicado a la tarea					X	5,0				
31	Número de errores detectados					X	5,0				
32	Espero dedicado a la tarea					X	5,0				
33	Número de errores detectados					X	5,0				
34	Espero dedicado a la tarea					X	5,0				
35	Número de errores detectados					X	5,0				
36	Espero dedicado a la tarea					X	5,0				
37	Número de errores detectados					X	5,0				
38	Espero dedicado a la tarea					X	5,0				
39	Número de errores detectados					X	5,0				
40	Espero dedicado a la tarea					X	5,0				
41	Número de errores detectados					X	5,0				
42	Espero dedicado a la tarea					X	5,0				
43	Número de errores detectados					X	5,0				
44	Espero dedicado a la tarea					X	5,0				
45	Número de errores detectados					X	5,0				
46	Espero dedicado a la tarea					X	5,0				
47	Número de errores detectados					X	5,0				
48	Espero dedicado a la tarea					X	5,0				
49	Número de errores detectados					X	5,0				
50	Espero dedicado a la tarea					X	5,0				
51	Número de errores detectados					X	5,0				
52	Espero dedicado a la tarea					X	5,0				
53	Número de errores detectados					X	5,0				
54	Espero dedicado a la tarea					X	5,0				
55	Número de errores detectados					X	5,0				
56	Espero dedicado a la tarea					X	5,0				
57	Número de errores detectados					X	5,0				
58	Espero dedicado a la tarea					X	5,0				
59	Número de errores detectados					X	5,0				
60	Espero dedicado a la tarea					X	5,0				
61	Número de errores detectados					X	5,0				
62	Espero dedicado a la tarea					X	5,0				
63	Número de errores detectados					X	5,0				
64	Espero dedicado a la tarea					X	5,0				
65	Número de errores detectados					X	5,0				
66	Espero dedicado a la tarea					X	5,0				
67	Número de errores detectados					X	5,0				
68	Espero dedicado a la tarea					X	5,0				
69	Número de errores detectados					X	5,0				
70	Espero dedicado a la tarea					X	5,0				
71	Número de errores detectados					X	5,0				
72	Espero dedicado a la tarea					X	5,0				
73	Número de errores detectados					X	5,0				
74	Espero dedicado a la tarea					X	5,0				
75	Número de errores detectados					X	5,0				
76	Espero dedicado a la tarea					X	5,0				
77	Número de errores detectados					X	5,0				
78	Espero dedicado a la tarea					X	5,0				
79	Número de errores detectados					X	5,0				
80	Espero dedicado a la tarea					X	5,0				
81	Número de errores detectados					X	5,0				
82	Espero dedicado a la tarea					X	5,0				
83	Número de errores detectados					X	5,0				
84	Espero dedicado a la tarea					X	5,0				
85	Número de errores detectados					X	5,0				
86	Espero dedicado a la tarea					X	5,0				
87	Número de errores detectados					X	5,0				
88	Espero dedicado a la tarea					X	5,0				
89	Número de errores detectados					X	5,0				
90	Espero dedicado a la tarea					X	5,0				
91	Número de errores detectados					X	5,0				
92	Espero dedicado a la tarea					X	5,0				
93	Número de errores detectados					X	5,0				
94	Espero dedicado a la tarea					X	5,0				
95	Número de errores detectados					X	5,0				
96	Espero dedicado a la tarea					X	5,0				
97	Número de errores detectados					X	5,0				
98	Espero dedicado a la tarea					X	5,0				
99	Número de errores detectados					X	5,0				
100	Espero dedicado a la tarea					X	5,0				

9 RESULTADOS

9.1 CARACTERIZACIÓN DE LAS PRÁCTICAS DEL MANTENIMIENTO DE SOFTWARE EN DEPARTAMENTOS DE SISTEMAS Y MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA.

Con el objetivo de obtener y analizar la información requerida para caracterizar las prácticas del mantenimiento de software, se utilizó el método de la encuesta. Para el estudio, los dos autores del trabajo se encargaron de realizar las 13 encuestas en los departamentos de sistemas y Mipymes desarrolladoras de software de la ciudad de Pereira.

- Parámetros de la encuesta
 - La encuesta fue diseñada de tal manera que fuera sencilla, concreta y concisa para recolectar y tabular más fácilmente la información requerida en el estudio.
 - La encuesta cuenta con objetivo y notas que permiten dar a conocer a los miembros de los departamentos de sistemas y Mipymes desarrolladoras de software encuestados, su finalidad.

- Temas de la encuesta

Las preguntas hechas en la encuesta se enfocan específicamente a los siguientes temas:

- Generalidades del mantenimiento.
- Planificación del mantenimiento.
- Estimación de costos.
- Métricas del mantenimiento.
- Proceso de mantenimiento.
- Personal de mantenimiento.
- Técnicas y herramientas de mantenimiento.

- Evolución o mejora del producto.

9.1.1 Tabulación de la información de la caracterización de las empresas en mantenimiento de software

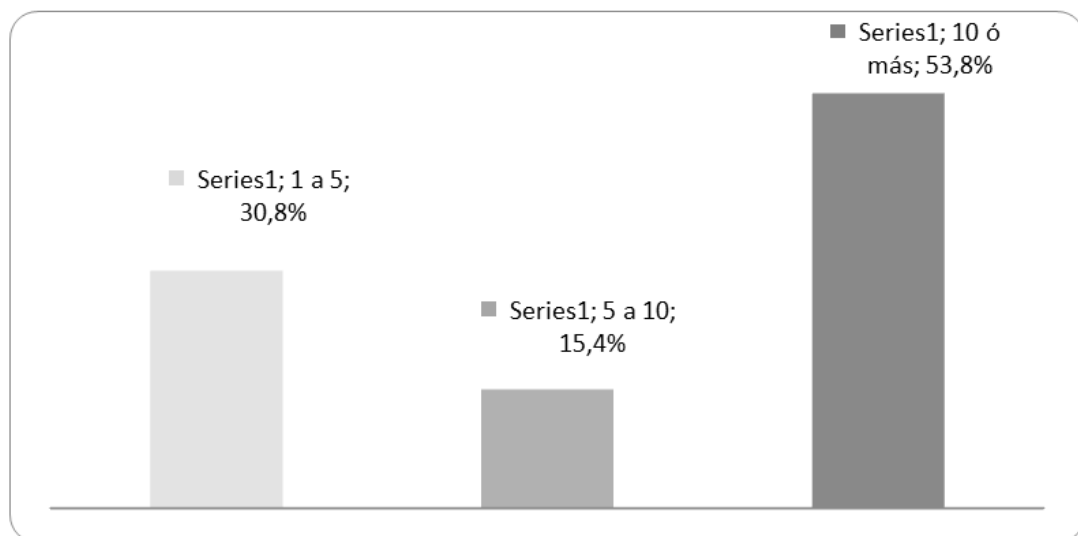
Con base en las encuestas sobre las prácticas del mantenimiento de software aplicado en las empresas, se obtuvo la información requerida para ser procesada, analizada y determinar así las conclusiones respectivas del estudio.

A. GENERALIDADES DEL MANTENIMIENTO

Esta primera fase busca tener un conocimiento general de las empresas en cuanto a su creación, años su actividad principal, procesos de certificación y así lograr caracterizarlas

1. ¿Años de creación de la empresa?

Figura 14. Años de creación de la empresa



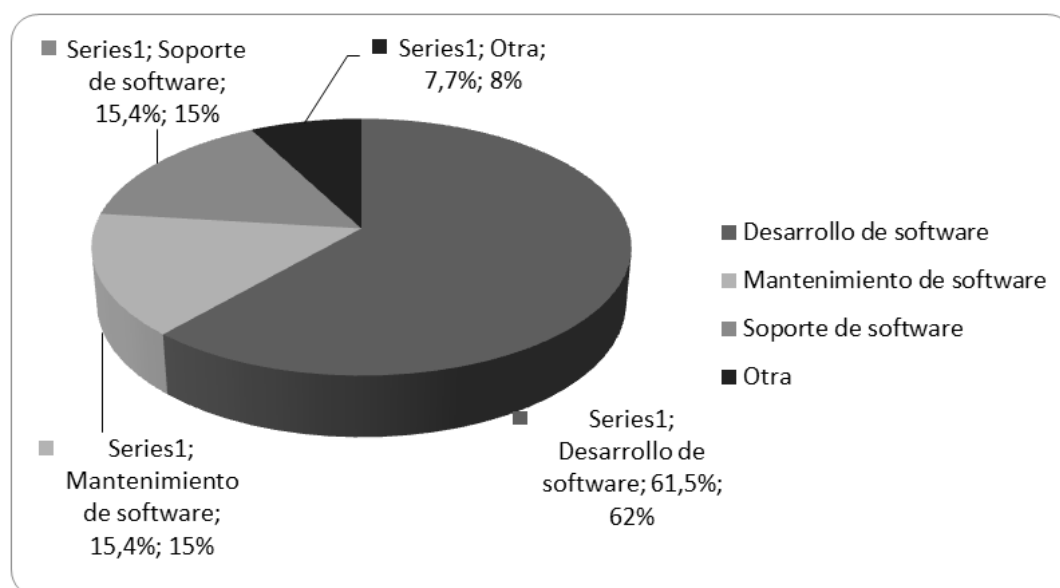
Fuente: Los Autores

En el momento de levantar la información y querer conocer y caracterizar las empresas del sector desarrollo se obtienen los siguientes resultados:

Del total de empresas que manifestaron interés en participar y que formaron parte de la muestra, el 53,8% lleva más de 10 años de haber sido creada, lo cual nos brinda un parte de confianza por su experiencia en el desarrollo, el 30,8% lleva entre 1 y 5 años y el restante 15,4% lleva en el mercado entre 5 y 10 años. A pesar de la relevancia en los años de creación y sostenimiento de dichas empresas se evidencia que a lo largo del tiempo se sostienen y van adquiriendo experiencia y posicionamiento en el mercado.

2. ¿Principal actividad de la empresa?

Figura 15. Actividad de la empresa



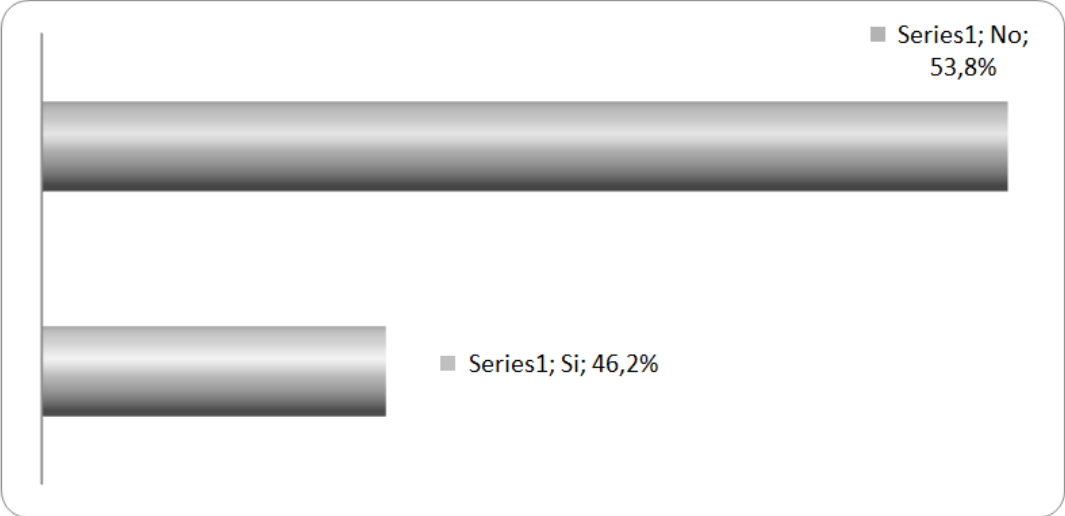
Fuente: Los Autores

El 61,5% de las empresas encuestadas tienen como principal actividad económica el desarrollo de software, el 15,4% se dedica al mantenimiento de software, el 15,4% ofrece soporte de software y el 7,7% a otra actividad diferente. En la línea de los procesos que manejan estas empresas se puede notar que el desarrollo de sistemas de información predomina

como actividad económica y a través de los años como lo muestra la Figura anterior no diversifican su actividad comercial y se mantienen en la línea del desarrollo.

3. ¿Tiene la empresa certificación de calidad?

Figura 16. Certificación de calidad

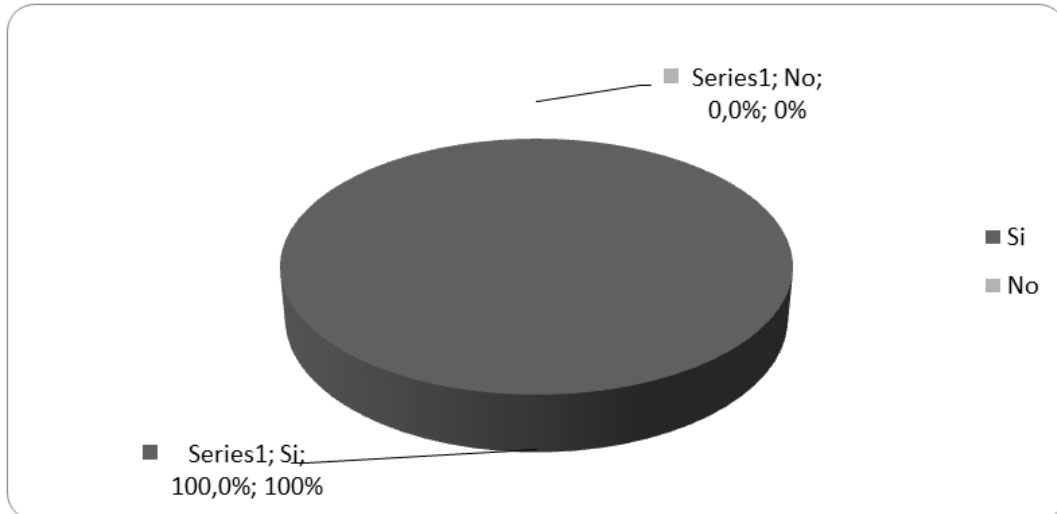


Fuente: Los Autores

Al momento de indagar sobre la certificación de calidad, el 46,2% está certificada en una de las siguientes Normas: ISO, TMMI, NTC 6001, CMMI Dev/3, IT Mark, ISO 1701 (No desarrollo de software), e ISO 9001, y el 53,8% no cuenta con ninguna certificación. Certificaciones como ISO, CMMI, IT Mark son certificaciones que han sido abordadas por las empresas para sus procesos de desarrollo y en esta investigación se ha indagado sobre algunas de ellas como son ISO, IT Mark y CMMI, por lo cual es importante resaltar que la implementación de algunas de ellas en estas empresas de la región es compleja, algunas elevan el costo en su implementación o requieren de mucho tiempo de preparación.

4. ¿La empresa realiza mantenimiento al software que desarrolla?

Figura 17. Realización de mantenimiento de software

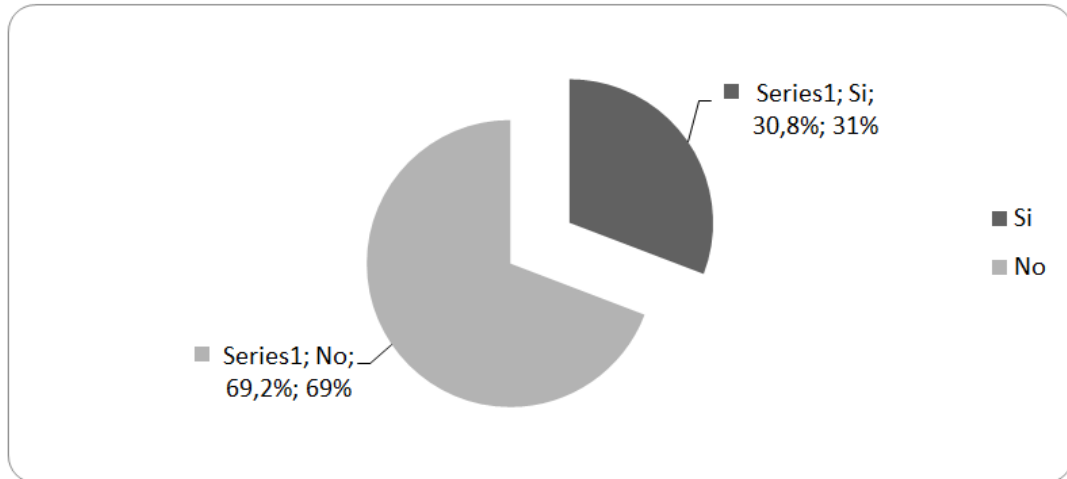


Fuente: Los Autores

Es importante anotar que el 100% de las empresas en estudio realiza el mantenimiento al software que desarrolla, pero únicamente el 30,8% (siguiente Figura) de ellas tiene como base o modelo alguna guía o norma para llevarlo a cabo, como son la CMMI – TSP – PSP, la MoProSoft y las buenas prácticas (empíricos); el restante 69,2% (siguiente Figura) no cuenta con ninguna guía o norma para tal fin.

4.1 Si respondió Si (punto 4): ¿Tiene como base o modelo alguna guía o norma para el mantenimiento del software?

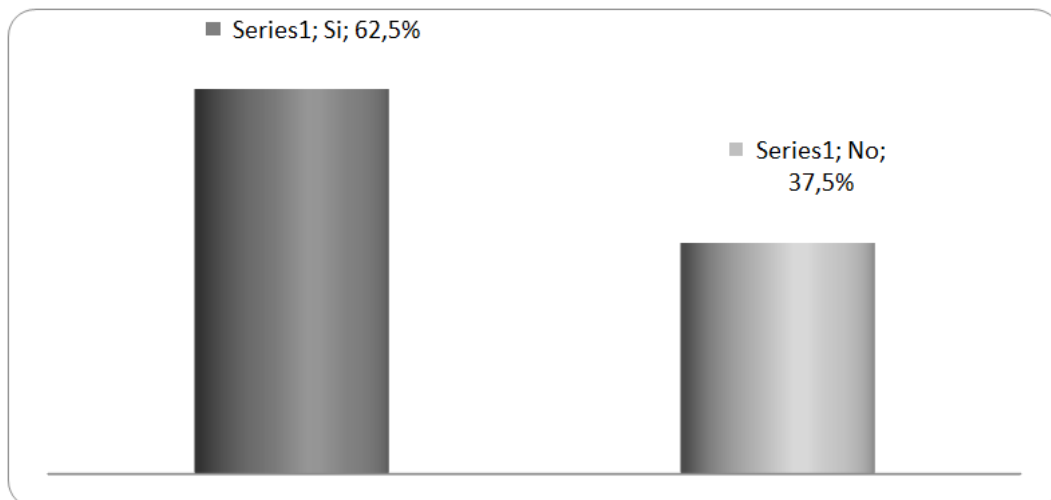
Figura 18. Modelo de guía para el mantenimiento



Fuente: Los Autores

4.2 Si respondió Si (punto 4): ¿Conoce para qué sirve una guía de mantenimiento y la forma en que puede ayudar a la empresa?

Figura 19. Conocimiento sobre guías de mantenimiento

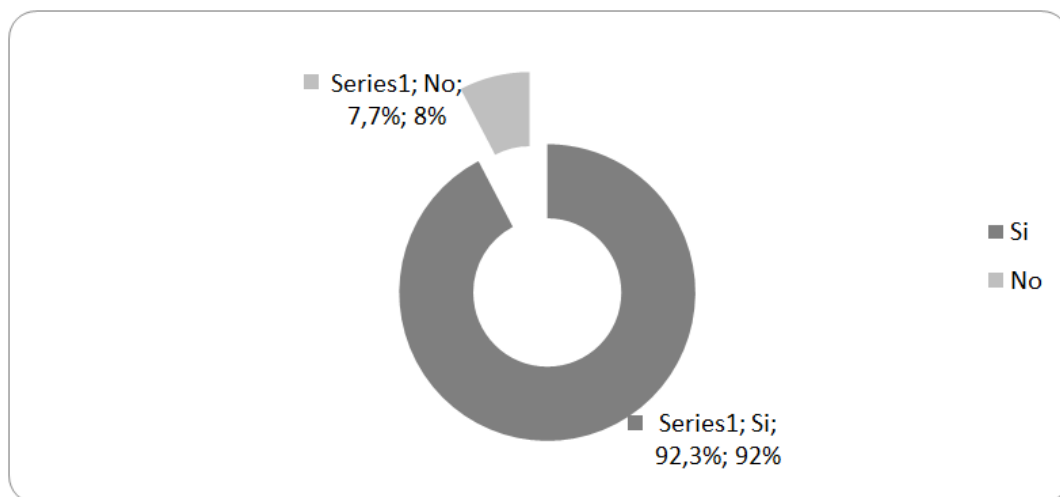


Fuente: Los Autores

Sin embargo, de estas últimas, el 62,5% conoce para qué sirven estas guías o normas y la forma en que pueden ayudar a la empresa al momento de aplicarlas; y el 37,5% no conoce ni su uso ni su ayuda.

5. ¿Tiene conocimientos sobre Mantenimiento de Software?

Figura 20. Conocimiento sobre mantenimiento de software

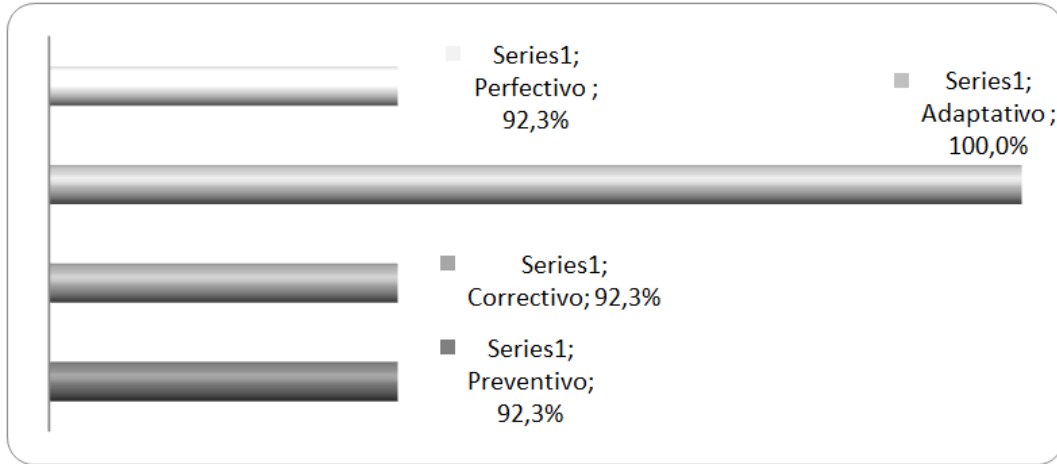


Fuente: Los Autores

Seguido en la encuesta, se preguntó si la empresa tiene conocimientos sobre mantenimiento de software, a lo cual el 92,3% afirmó que si y el 7,7% expresó no tenerlos. Con respecto al tipo de mantenimiento de software realizado por las empresas, éstas respondieron que llevan a cabo el mantenimiento preventivo, correctivo y perfectivo con un 92,3% respectivamente y mantenimiento adaptativo para el 100,0% de ellas (siguiente Figura)

6. ¿Qué tipo de mantenimiento de software realiza la empresa?

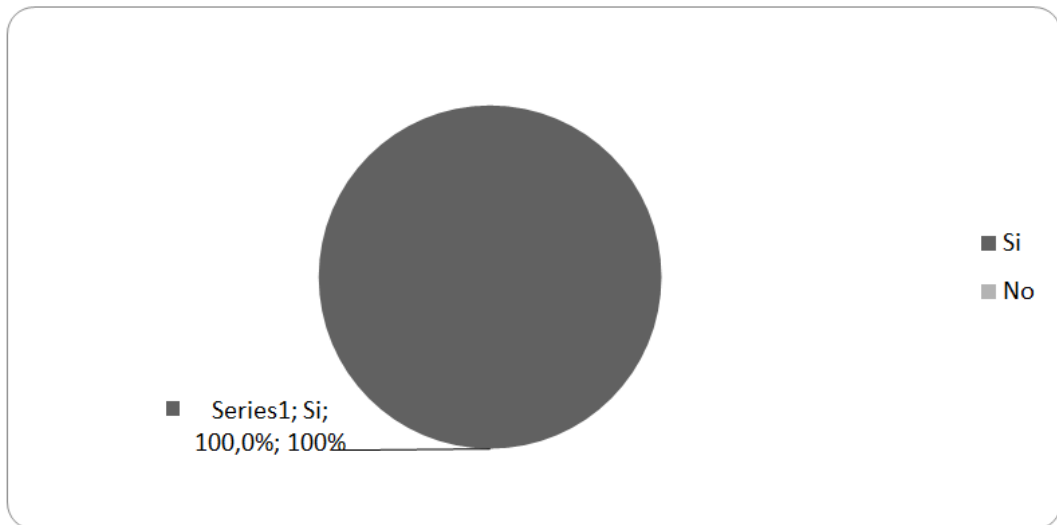
Figura 21. Tipos de mantenimiento de software que realiza



Fuente: Los Autores

7. ¿Haría uso de la aplicación de un proceso de mantenimiento en su empresa?

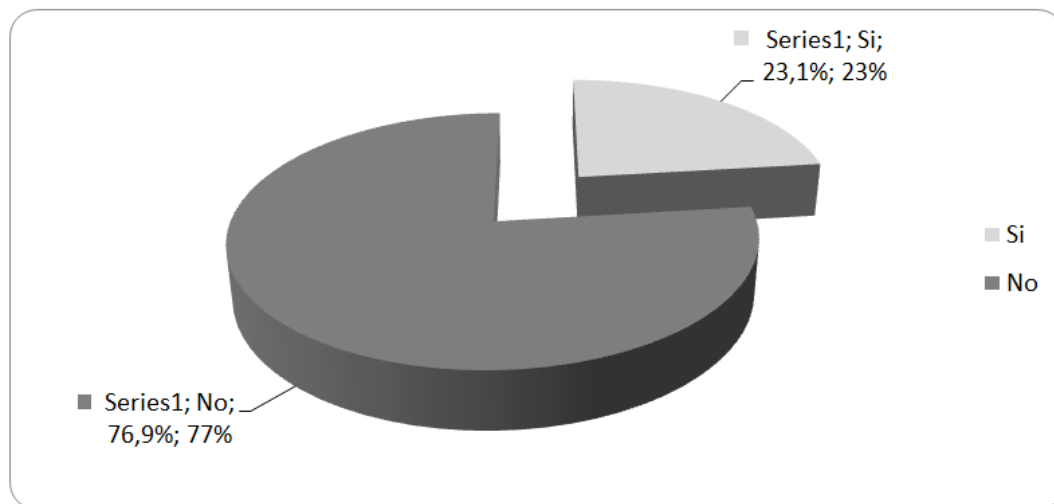
Figura 22. Uso de la aplicación de un proceso de mantenimiento



Fuente: Los Autores

8. ¿Realizan tercerización del mantenimiento del Software de sus productos?

Figura 23. Tercerización del mantenimiento de software



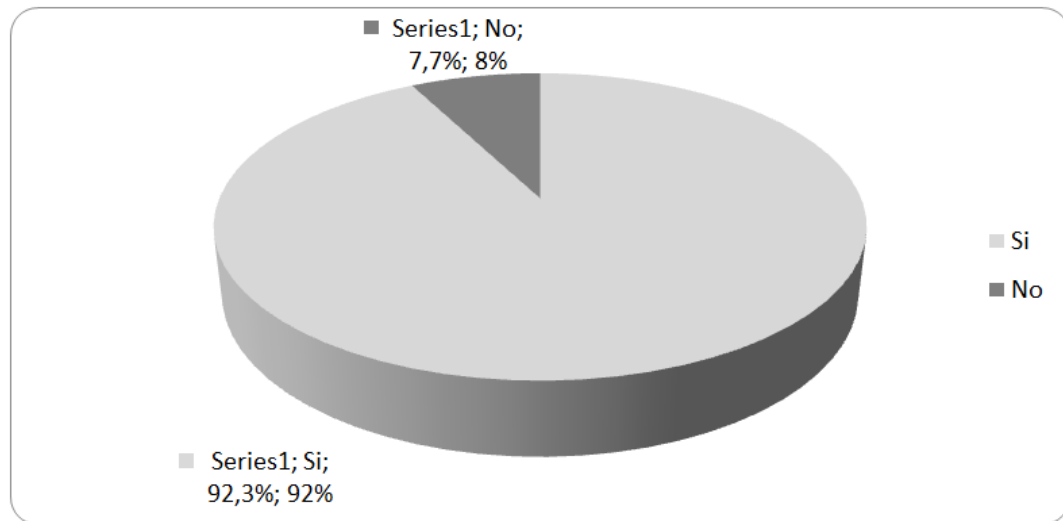
Fuente: Los Autores

El 76,9% de las empresas no realizan la tercerización del mantenimiento de software de sus productos, pero el 23,1% si lo hacen.

La creación de esta guía socializada de una forma ágil y didáctica a estas empresas y a las demás empresas de la región puede lograr que este porcentaje identificado rompa un paradigma en el tema de mantenimiento para un uso masivo y creación de versiones ajustadas.

9. ¿Dan garantía del software?

Figura 24. Garantía del software

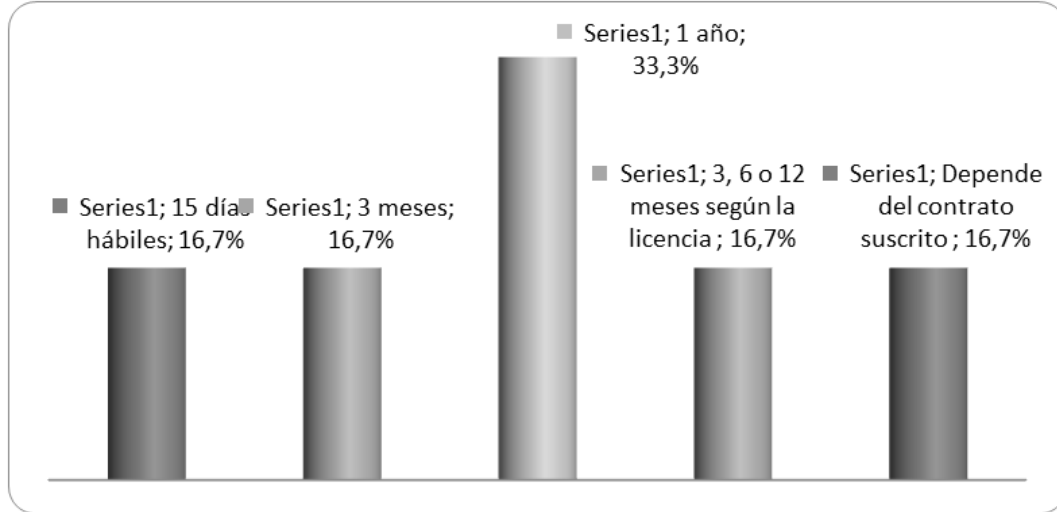


Fuente: Los Autores

En este aparte se evidencia que el 7,7% de los encuestados no brinda garantía del software que desarrolla y el 92,3% si brinda la garantía, de éstas el 33,3% ofrece 1 año de garantía, el 16,7% ofrece 15 días hábiles, el 16,7% ofrece 3 meses, el 16,7% ofrece 3, 6 o 12 meses según la licencia y el 16,7% (Figura 16) expresa que la garantía depende del contrato suscrito con el cliente.

9.1 Si respondió Si (punto 9): ¿Cuánto tiempo de garantía del SW ofrece?

Figura 25. Tiempo de garantía del software

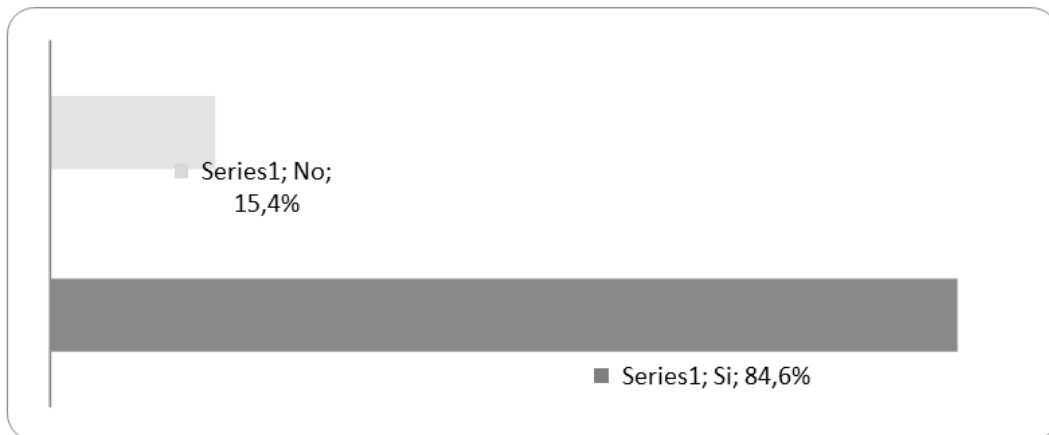


Fuente: Los Autores

B. PLANIFICACIÓN DEL MANTENIMIENTO

10. ¿La empresa planifica las operaciones y recursos que se van a realizar en el mantenimiento del software?

Figura 26. Planificación de operaciones y recursos

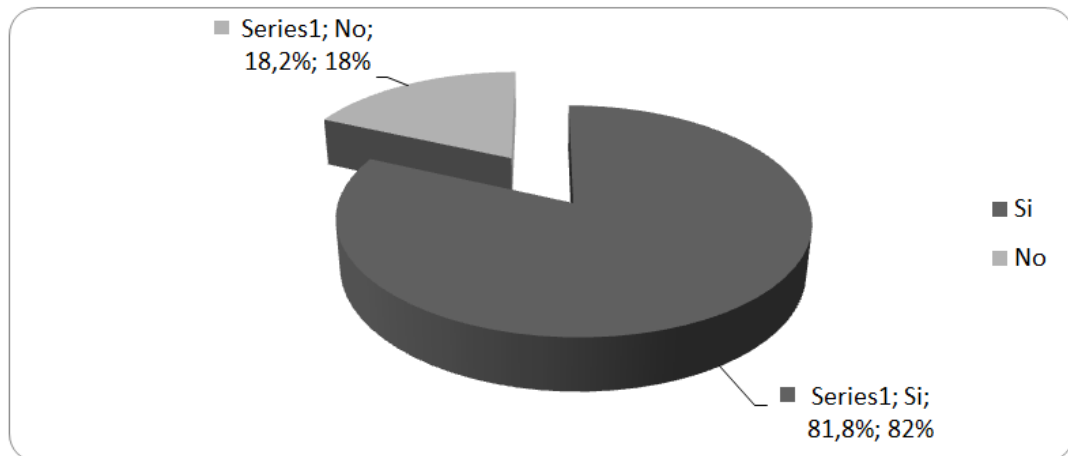


Fuente: Los Autores

En cuanto a la planificación de operaciones y recursos que se van a realizar en el mantenimiento de software, donde el 84,6% de los encuestados tiene presente esta planificación y el 15,4% no la lleva a cabo.

10.1 Si respondió Si (punto 10): ¿Se realiza un plan de mantenimiento durante el desarrollo del SW?

Figura 27. Realización del plan de mantenimiento

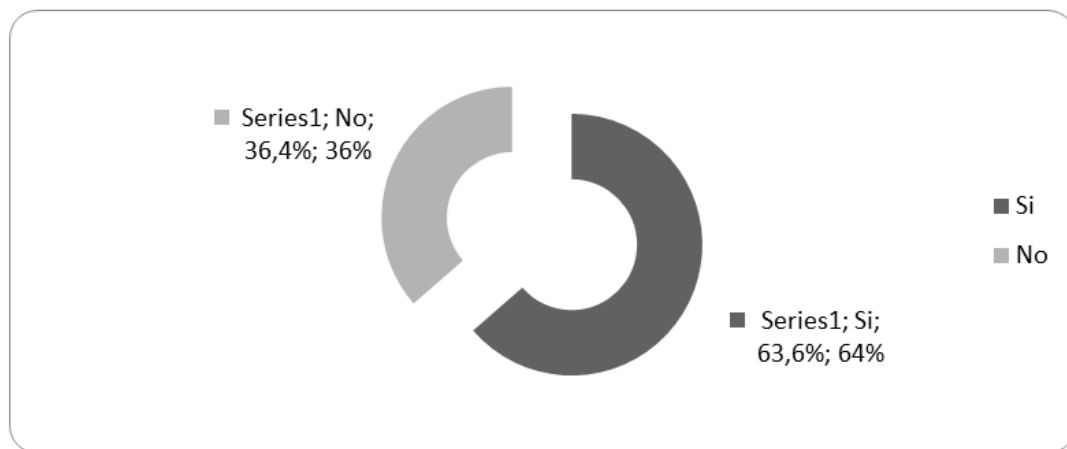


Fuente: Los Autores

De aquellas empresas que, si planifican las operaciones y recursos requeridos, el 81,8% realizan un plan de mantenimiento durante el desarrollo del software y el 18,2% no lo realiza; de igual manera, el 63,6% de las empresas planificadoras saben cuánto va a durar el mantenimiento de software y el 36,4% de estas mismas no estima su duración (siguiente Figura).

10.2 Si respondió Si (punto 10): ¿En la planificación sabe cuánto va a durar el mantenimiento del Software?

Figura 28. Tiempo de duración del mantenimiento



Fuente: Los Autores

C. ESTIMACIÓN DE COSTOS

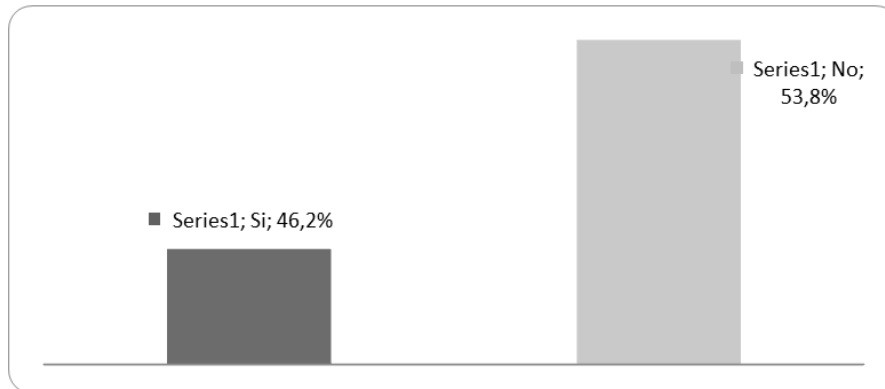
El 46,2% de las empresas afirma contar con un presupuesto aparte sólo para el mantenimiento de software, mientras que el 53,8% no cuenta con presupuesto únicamente destinado a esta labor. Sin embargo, el 76,9% de ellas estima los costos de este mantenimiento y el 23,1% no lleva a cabo esta importante actividad.

Para estimar los costos del mantenimiento de software, el 50,0% de las empresas utiliza la técnica Juicio de Expertos, el 30% hace uso de una estimación paramétrica, el 10% utiliza una estimación paramétrica y el 10% utiliza otra técnica diferente.

Continuando con el tema, del total de las empresas que estiman los costos de mantenimiento, únicamente el 70,0% tiene conocimiento de cuánto cuesta el mantenimiento al producto y también, el 70,0% tiene en cuenta los costos de operación del hardware donde va a funcionar el producto de software.

11. ¿Tiene un presupuesto aparte solo para el mantenimiento de software?

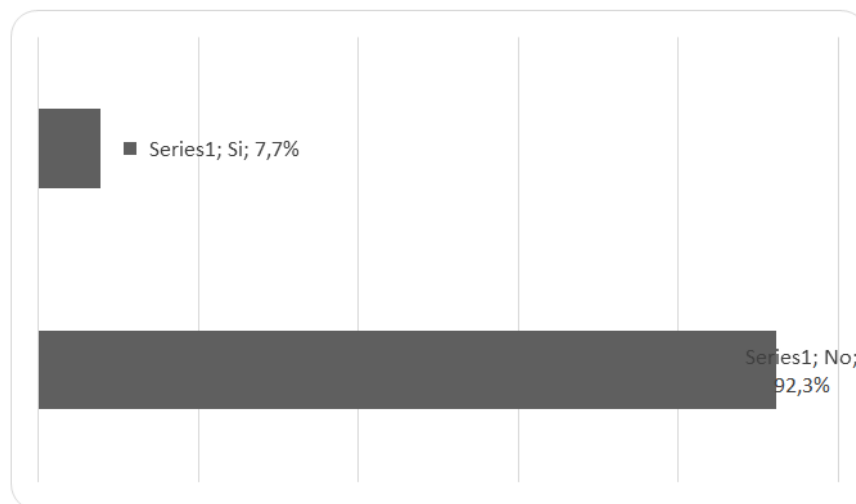
Figura 29. Presupuesto de mantenimiento de software



Fuente: Los Autores

12. ¿La empresa estima los costos del mantenimiento del software?

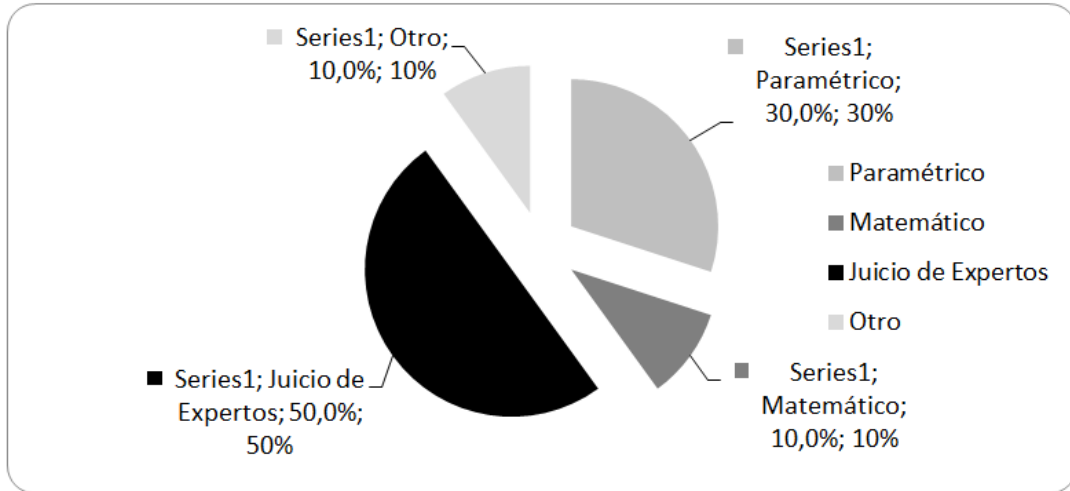
Figura 30. Estimación de los costos de mantenimiento de software



Fuente: Los Autores

12.1 Si respondió Si (punto 12): ¿Para estimar el costo que técnica utiliza?

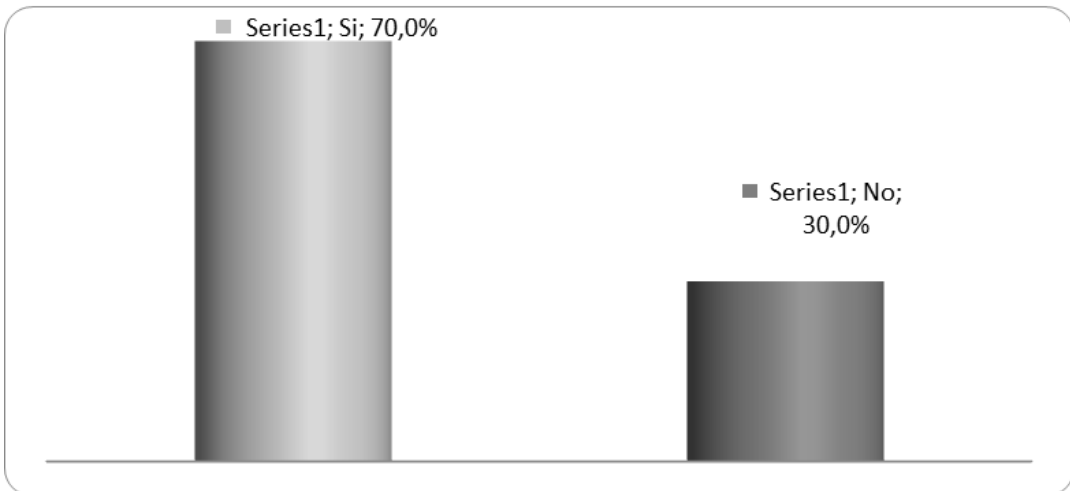
Figura 31. Técnica utilizada para estimar los costos



Fuente: Los Autores

12.2 Si respondió Si (punto 12): ¿Tiene conocimiento de cuánto cuesta el mantenimiento al producto?

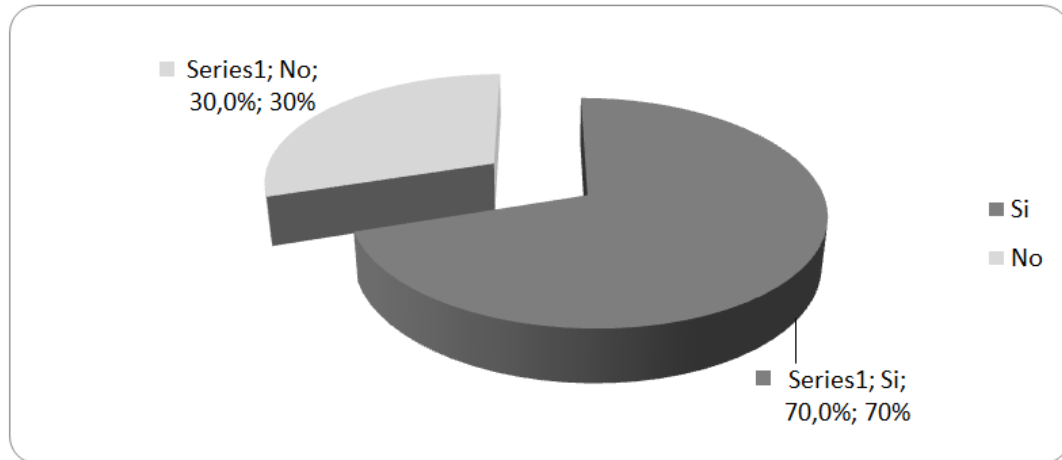
Figura 32. Costo del mantenimiento de software



Fuente: Los Autores

12.3 Si respondió Si (punto 12): ¿Tiene en cuenta los costos de operación del Hardware donde va a funcionar el producto de Software?

Figura 33. Estimación de costos de operación del hardware

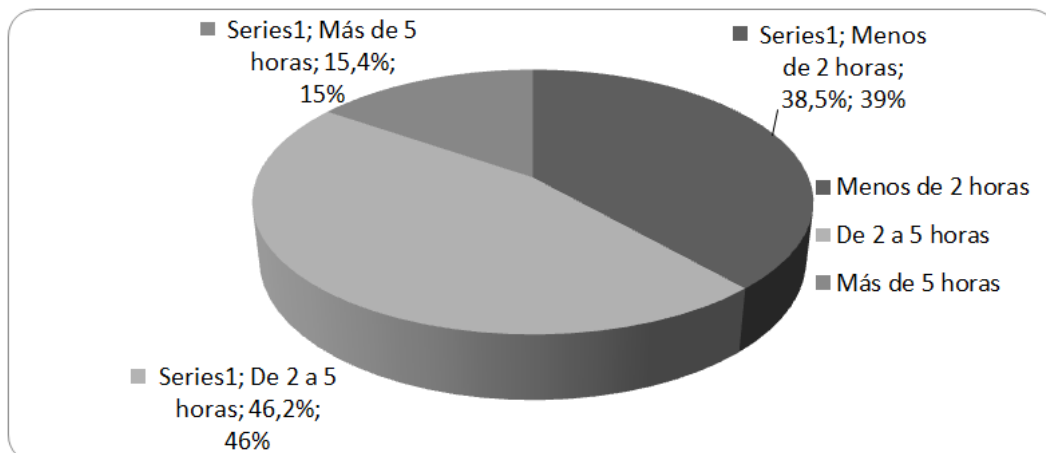


Fuente: Los Autores

D.MÉTRICAS DEL MANTENIMIENTO

13. ¿Cuánto tiempo promedio dedica el desarrollador para entender el cambio, mejora o corrección a realizar?

Figura 34. Tiempo para entender cambio, mejora o corrección



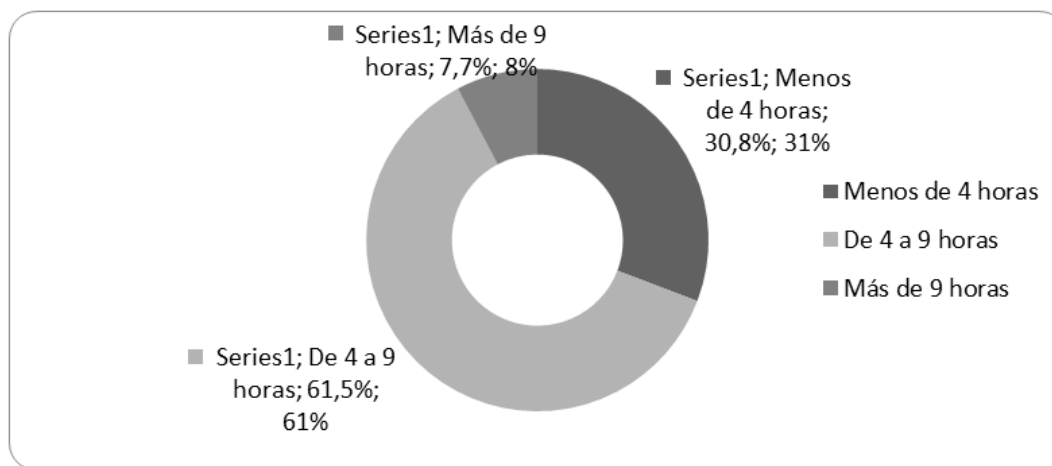
Fuente: Los Autores

Sobre el tiempo promedio dedicado por el desarrollador para entender el cambio, mejora o corrección a realizar, el 46,2% de las empresas encuestadas respondió que, de 2 a 5 horas, el 38,5% invierte menos de 2 horas y el 15,4% dedica más de 5 horas.

En estos procesos el tiempo utilizado de más en mejoras y reprocesos encarece el valor del producto, no para el cliente sino para la empresa, quien debe invertir en el personal más tiempo del presupuestado dejando de lado otros proyectos y retrasando una línea de desarrollo como tal.

14. ¿Cuánto tiempo promedio se demora en solucionar un defecto?

Figura 35. Tiempo de solución de defecto



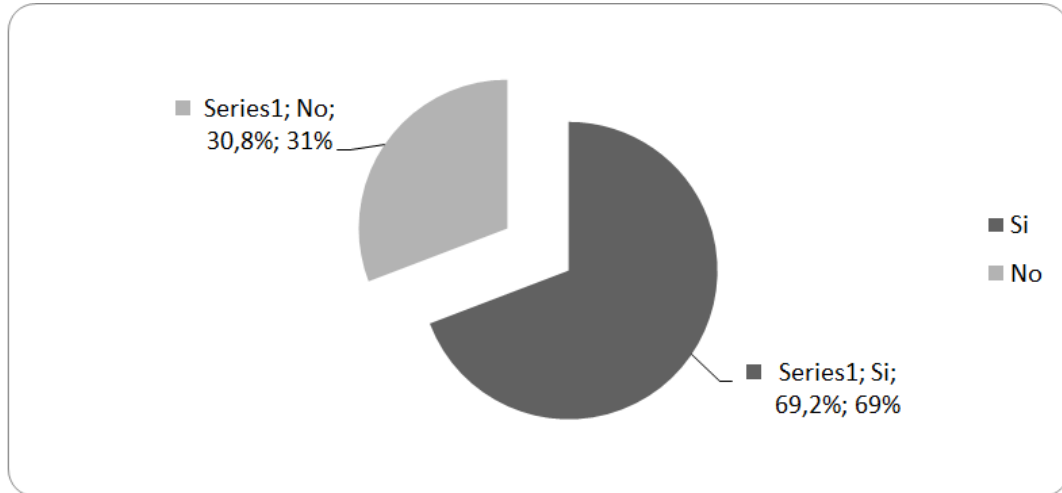
Fuente: Los Autores

En cuanto al tiempo promedio invertido en la solución del defecto, el 61,5% de las empresas invierte entre 4 y 9 horas, el 30,8% lo realiza en menos de 4 horas y el 7,7% considera que requiere más de 9 horas para llevar a cabo este procedimiento, mejora que la guía

pretende implementar.

15. ¿Registran cada uno de los defectos encontrados?

Figura 36. Registro de los defectos



Fuente: Los Autores

Como es importante registrar cada uno de los defectos encontrados al momento de realizar el mantenimiento de software, el 69,2% de las empresas encuestadas realizan esta función y el faltante 30,8% no registran los defectos encontrados.

15.1 Si respondió Si (punto 15): ¿Cuántos defectos se corrigen al producto por mes?

Figura 37. Número de defectos corregidos por mes

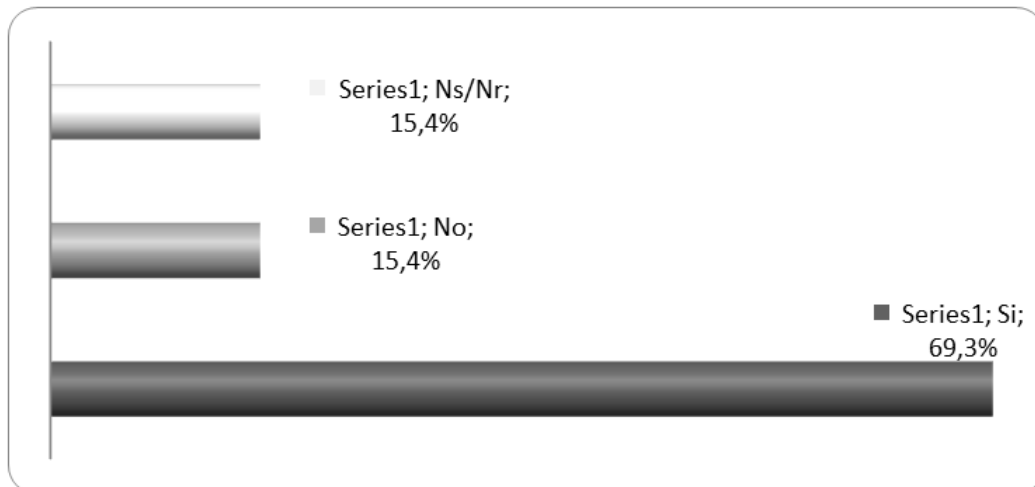


Fuente: Los Autores

Del 69,2% de las empresas que registran sus defectos, el 55,6% corrigen entre 1 y 10 defectos al producto por mes y el 44,4% entre 11 y 30 de estos defectos.

16. ¿Miden la calidad del código, para tener una buena mantenibilidad sobre el mismo?

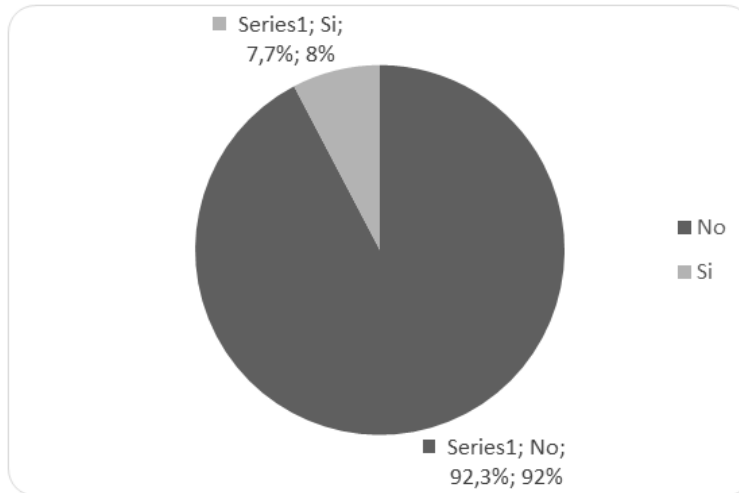
Figura 38. Medición de la calidad del código



Fuente: Los Autores

17. ¿Sabe cuál es el Índice de Madurez de su Software (IMS)?

Figura 39. Conocimiento del IMS



Fuente: Los Autores

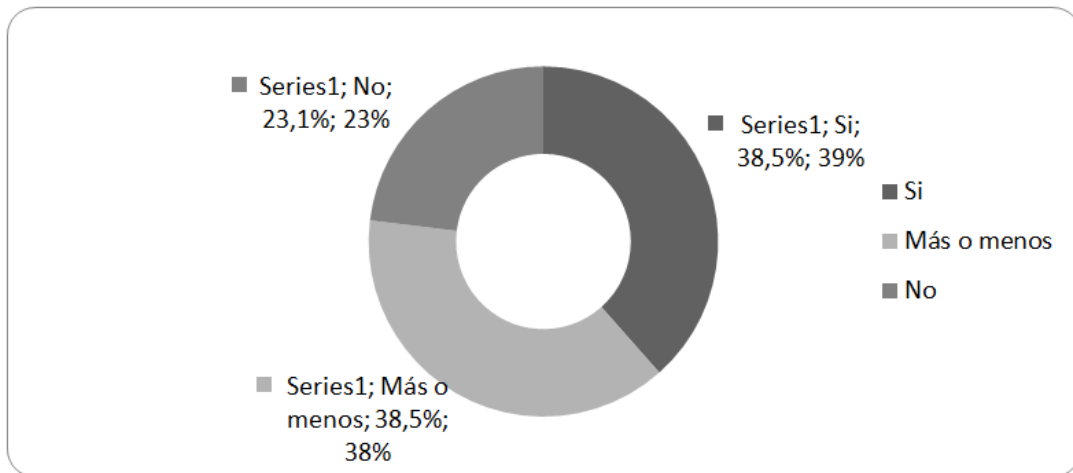
A la pregunta ¿Cómo miden la calidad del código para tener una buena mantenibilidad sobre el mismo?, el 7,7% contestó que con Indicadores de calidad interna y de calidad cliente, el 7,7% lo hace con una Revisión par – testing – banco de pruebas – casos de uso, el 7,7% con Funcionalidad, normalización, líneas de código empleadas, el 7,7% con Documentación, reutilización del código, control de errores y excepciones, el 7,7% con Métrica llamada producto no conforme, el 7,7% con TDD pruebas de unidad, el 7,7% con Funcionalidad, confiabilidad, portabilidad y usabilidad, el 7,7% con Pruebas automatizadas e integración continua, el 7,7% lo hace con herramientas de generación, el 7,7% indica que esta actividad no se realiza debido a la dinámica del día a día, se omite, pero que está en definición un proceso, el 7,7% respondió que en la empresa no se mide la calidad del código, y el 15,4% No sabe o no responde a esta pregunta.

Finalmente, el 92,3% de los encuestados no sabe cuál es el Índice de Madurez de su software (IMS) y el 7,7% de las empresas si sabe cuál es su IMS.

E. PROCESO DE MANTENIMIENTO

18. ¿Tiene suficientes recursos (tiempo y personal) para llevar a cabo las tareas de mantenimiento?

Figura 40. Recursos para el mantenimiento

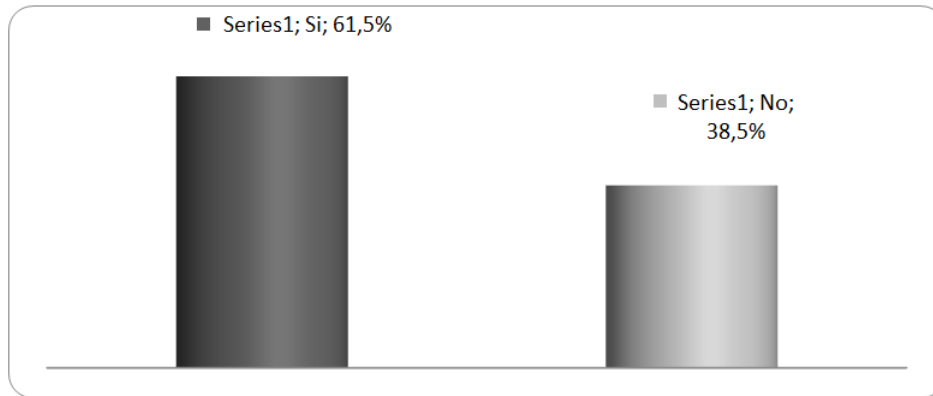


Fuente: Los Autores

El 38,5% de los encuestados mencionaron que si tenían los recursos necesarios para hacer el mantenimiento a sus productos de software. El 38,5% indicaron que los recursos con los que contaban eran más o menos suficientes para abarcar el mantenimiento de software. Mientras que el 23,1% no tienen los recursos suficientes.

19. ¿Realiza un análisis de impacto de la corrección o mejora a realizar en el producto?

Figura 41. Realización del impacto de la corrección o mejora del producto

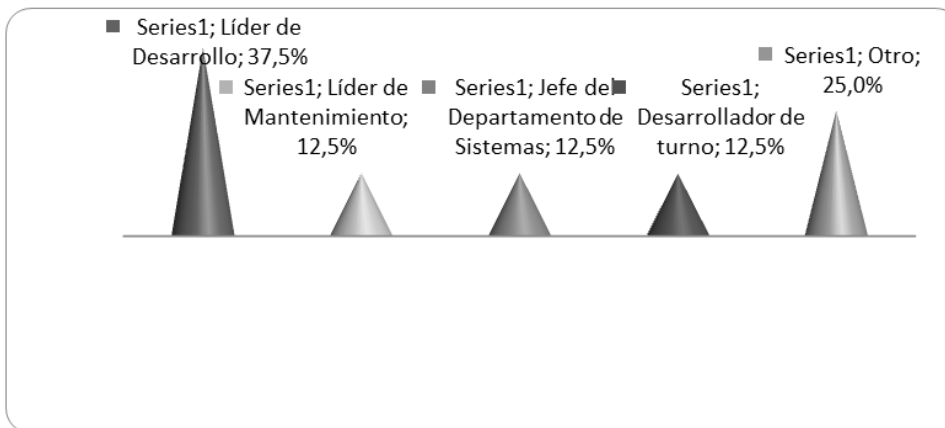


Fuente: Los Autores

En cuanto al impacto de la corrección o mejora realizada en el producto, el 61,5% realiza un análisis de dicho impacto, pero el 38,5% no lo hace.

19.1 Si respondió Si (punto 19): ¿Quién realiza el análisis?

Figura 42. Responsable de realizar el análisis

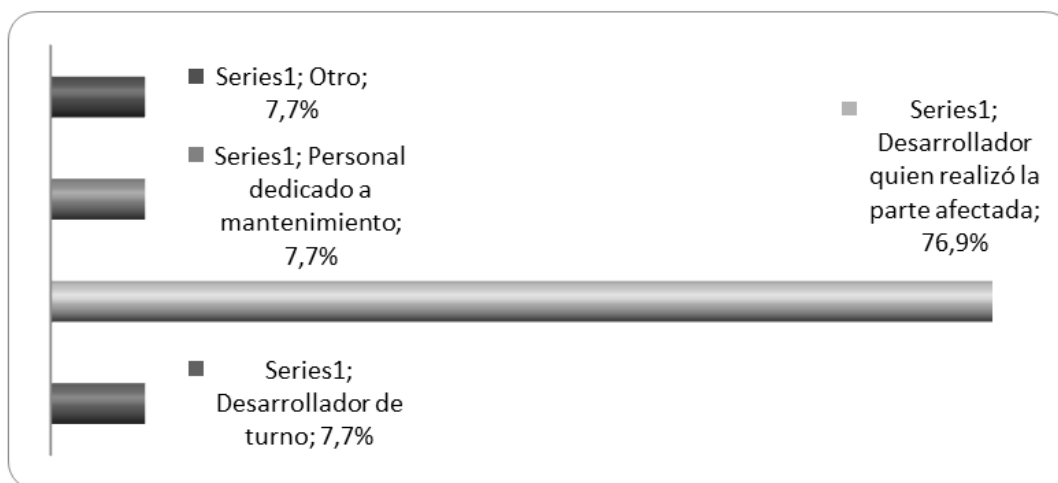


Fuente: Los Autores

Se evidencia que de las empresas que realizan este análisis el 37,5% expresó que es responsabilidad del Líder de Desarrollo, el 12,5% dijo que el Líder de Mantenimiento, el 12,5% que el jefe del Departamento de Sistemas, el 12,5% que el Desarrollador de turno y el 25,0% que otro.

20. ¿Quién realiza las correcciones a los defectos reportados?

Figura 43. Responsable de realizar correcciones

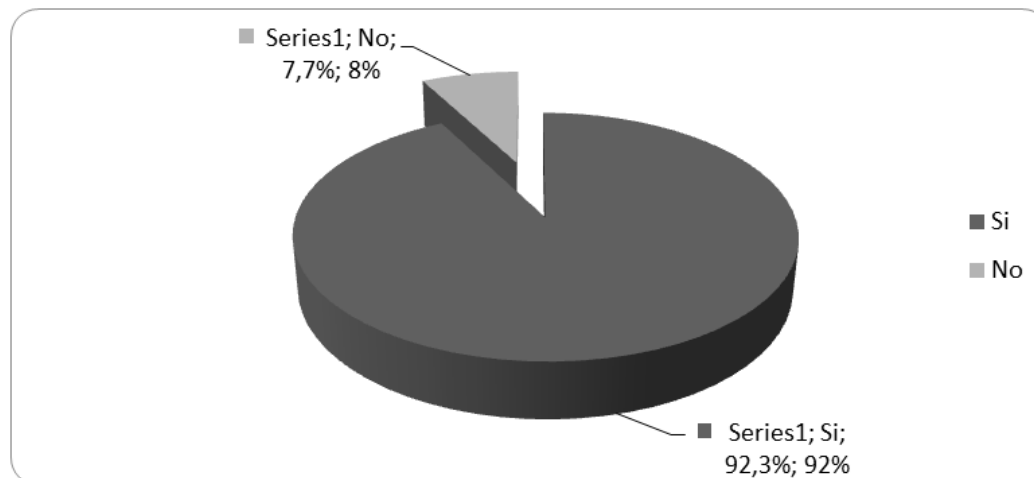


Fuente: Los Autores

Para el 76,9% de las empresas, el Desarrollador quien realizó la parte afectada es quien realiza las correcciones a los defectos reportados, para el 7,7% es el Desarrollador de turno, para el 7,7% es el personal dedicado a mantenimiento y para el 7,7% de las empresas es otro cargo.

21. ¿Le ha pasado que corrija algún defecto en el software y resulta dañando otra parte del software?

Figura 44. Ocurrencia de daños al realizar correcciones

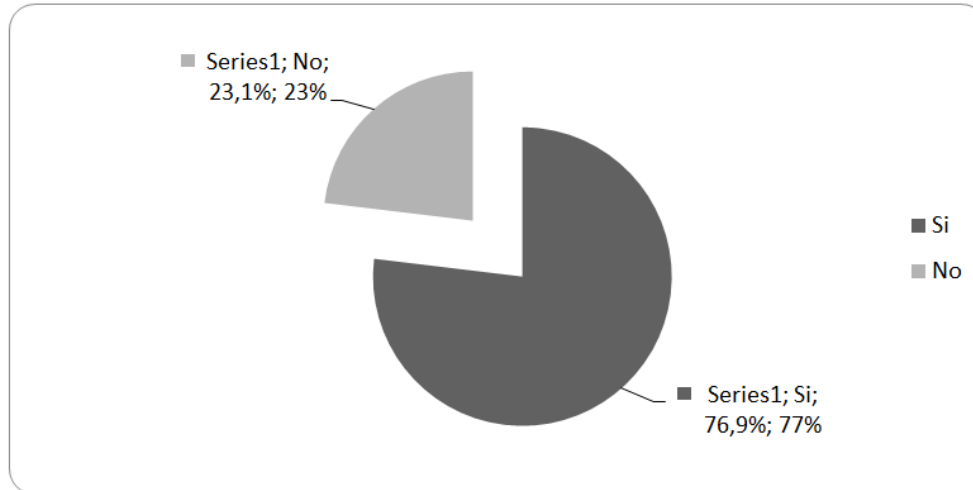


Fuente: Los Autores

Las respuestas a la pregunta número 21 exponen, que para el 92,3% de los encuestados le ha pasado que corrija algún defecto en el software y resulta dañando otra parte del software, mientras que el restante 7,7% nunca le ha pasado este problema.

22. ¿Realizan un plan de retiro del producto?

Figura 45. Realización del plan de retiro del producto

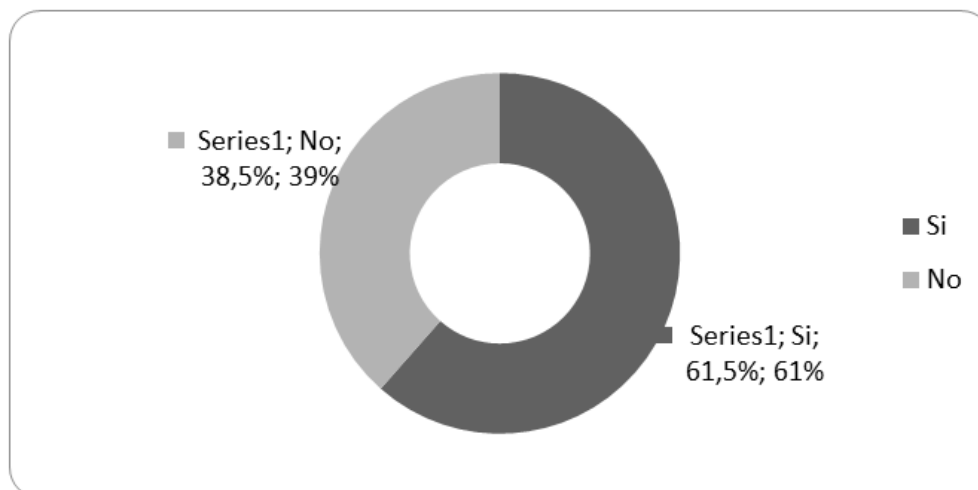


Fuente: Los Autores

Así mismo el 76,9% de las empresas realizan un plan de retiro del producto y el 23,1% no realiza este plan,

23. ¿Han hecho el retiro de algún software?

Figura 46. Realización de retiro de software

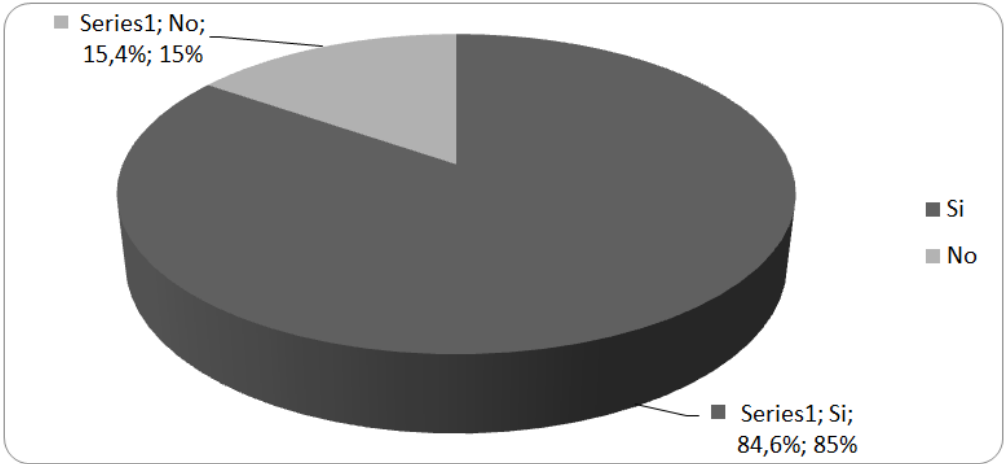


Fuente: Los Autores

Al indagar sobre la ejecución del retiro de algún software, el 38,5% no lo ha hecho y 61,5% de las empresas si lo ha hecho; de estas últimas, el 61,5% respondió sobre el número de productos que ha retirado siendo 2 productos para el 40,0%, 4 productos para el 20,0%, 20 productos para el 20,0% y 100 productos para el restante 20,0% de ellas.

24. ¿Documentan el cambio, mejora o defecto a realizar, sus pruebas, solución y las opciones de implementación?

Figura 47. Realización de documentación de acciones de mantenimiento

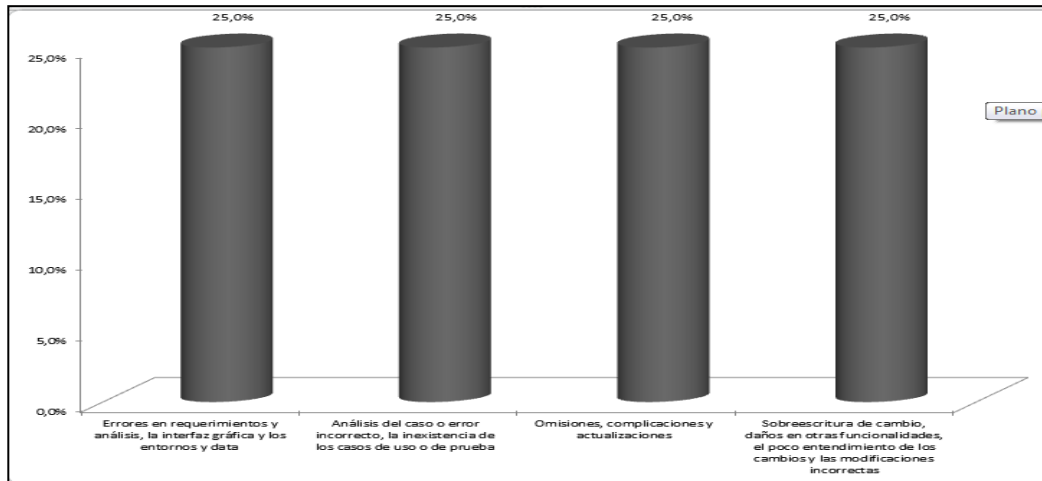


Fuente: Los Autores

El 84,6% de las empresas encuestadas documentan el cambio, mejora o defecto a realizar, sus pruebas, solución y las opciones de implementación, y el faltante 15,4% no realiza este tipo de documentación.

25. Por favor indique tres de los errores más comunes que se presentan durante el mantenimiento de software

Figura 48. Principales errores en el mantenimiento de software

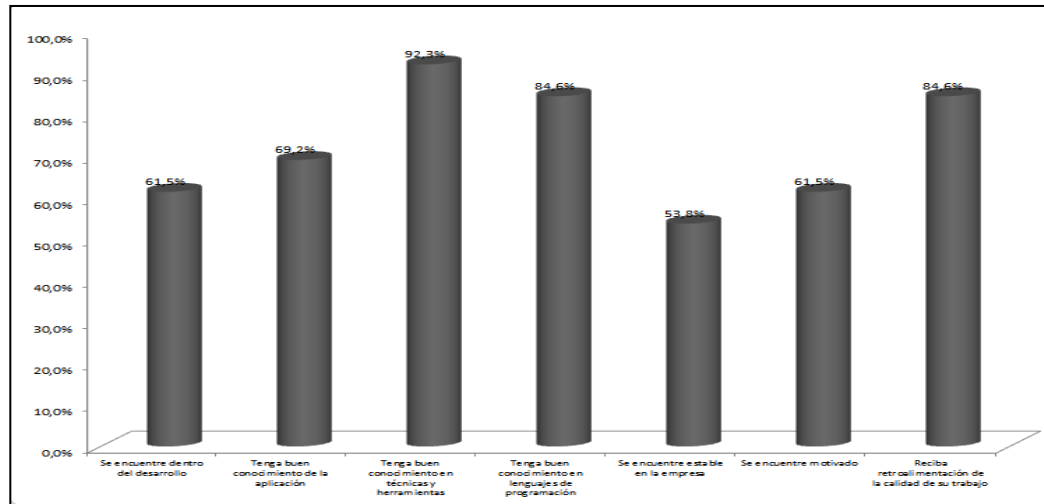


Fuente: Los Autores

Entre los tres errores más comunes que se presentan durante el mantenimiento de software están, los errores en requerimientos y análisis, la interfaz gráfica y los entornos y data para el 7,7%, el análisis del caso o error incorrecto, la inexistencia de los casos de uso o de prueba para el 7,7%, las omisiones, complicaciones y actualizaciones para el 7,7%, la omisión de los pasos estandarizados para el mantenimiento, la improvisación por creer conocer el proceso por parte del equipo de mantenimiento para el 7,7%, la sobre escritura de cambio, daños en otras funcionalidades, el poco entendimiento de los cambios y las modificaciones incorrectas para el 7,7%, los resultados no esperados para el 7,7% de las empresas, el daño de otros módulos, mayor tiempo al estimado y el poco control de cambios para el 7,7%, el juzgar mal el tiempo que se lleva implementar la solución para el 7,7%, la afectación a otros componentes, salir del alcance y consumir más tiempo para el 7,7%, el software no comentado, las funcionalidades no claras y no uso del manual para el 7,7% y el 23,1% no sabe o no responde a esta pregunta.

F. PERSONAL DE MANTENIMIENTO

Figura 49. Estado del personal de mantenimiento



Fuente: Los Autores

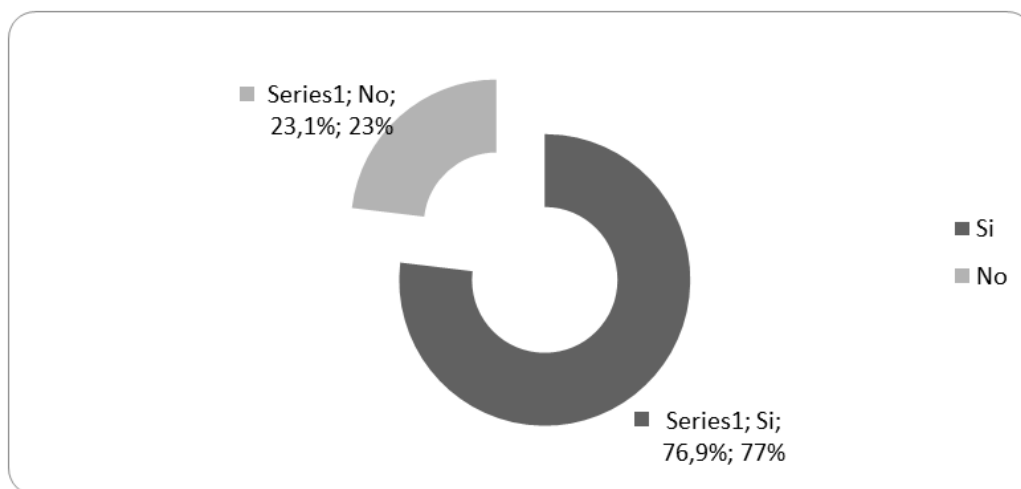
Para las preguntas de este numeral se solicitó a las empresas encuestadas que indicaran si estaban de acuerdo o en desacuerdo con diferentes afirmaciones (siendo 1 = “está en total desacuerdo” y 5 = “está totalmente de acuerdo”). En cuanto a la importancia de que el equipo de mantenimiento se encuentre dentro del desarrollo, el 61,5% de las empresas en estudio calificaron esta afirmación con 5, el 15,4% calificó con 4, el 15,4% calificó con 2 y el 7,7% le otorgó un 1. Acerca de la importancia de que el equipo de mantenimiento tenga buen conocimiento de la aplicación, el 69,2% otorgó un 5, el 23,1% un 4 y el 7,7% calificó con 1. Sobre la importancia de que el equipo de mantenimiento tenga buen conocimiento en técnicas y herramientas, el 92,3% calificó 5 y el 7,7% calificó un 4. A la afirmación sobre si es importante que el equipo de mantenimiento tenga buen conocimiento en lenguajes de programación, el 84,6% expresó como calificación un 5, el 7,7% un 4 y el 7,7% un 3. Sobre la importancia de que el equipo de mantenimiento se encuentre estable en la empresa, el 53,8% adjudicó un 5, el 23,1% lo hizo con un 4, el 7,7% con un 3 y el 15,4% con un 1. Con la afirmación de que es importante que el equipo de mantenimiento se encuentre motivado, el 61,5% indicó su posición con un número 5, el

23,1% indicó un 4 y el 15,4% lo hizo con un 3. Y, por último, se preguntó si es importante que el equipo de mantenimiento reciba retroalimentación de la calidad de su trabajo, para el 84,6% de las empresas esta afirmación se calificó con 5 y para el 15,4% con un 4.

G. TÉCNICAS Y HERRAMIENTAS DE MANTENIMIENTO

26. ¿Cuenta con herramientas para registrar los defectos que se reportan?

Figura 50. Existencia de herramientas para registro de defectos

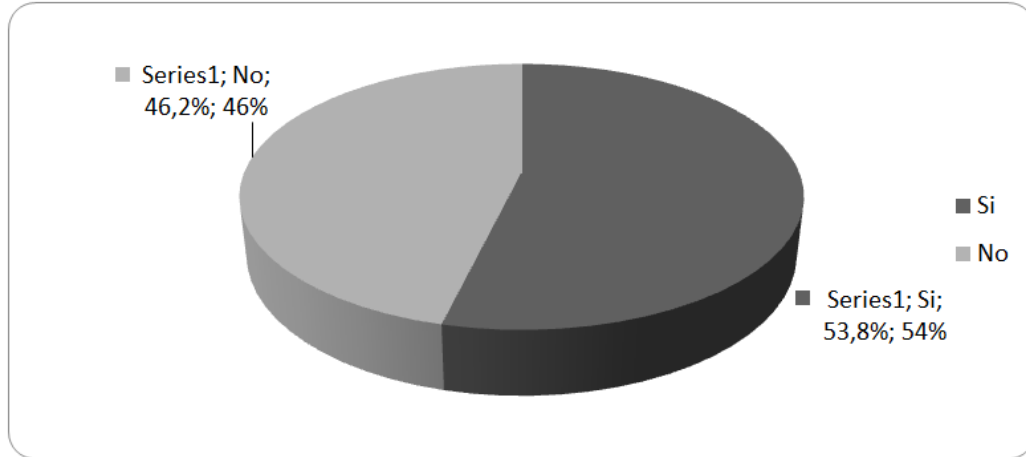


Fuente: Los Autores

Al indagar sobre la existencia de herramientas para registrar los defectos que se reportan, el 23,1% expresó que no tenerlas y el 76,9% de los encuestados afirmó contar con dichas herramientas, siendo las más utilizadas las hojas de cálculo, la integración continua (Gitlab CI y pipelines), la Mantis – Desarrollo interno, la Mantis – Taiga – Otros, la Pivotal tracker, changelog y el Software específico para el 10,0% respectivamente y la Jira y una herramienta propia para el 20,0% respectivamente.

27. ¿Cuenta con herramientas para la gestión de la configuración y versiones?

Figura 51. Existencia de herramientas para la gestión de la configuración y versiones

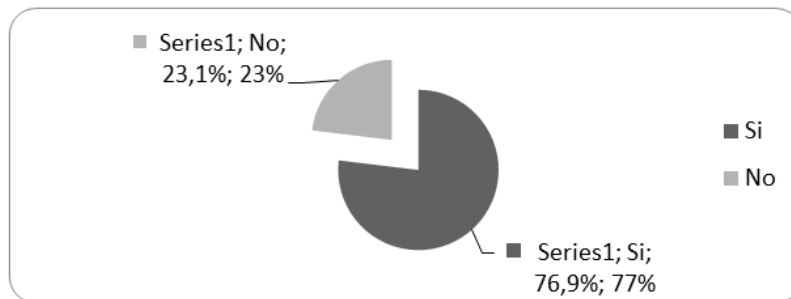


Fuente: Los Autores

El 53,8% de las empresas cuenta con herramientas para la gestión de la configuración y versiones, y el 46,2% no cuenta con estas importantes herramientas; siendo las más utilizadas por las empresas la Subversión con un 57,1%, la Documentación permanente, la Git y la SVN – Git – Vss con un 14,3% de las empresas respectivamente.

28. ¿Cuenta con herramientas para la gestión del conocimiento (Documentación, lecciones aprendidas, etc.)?

Figura 52. Existencia de herramientas para la gestión del conocimiento



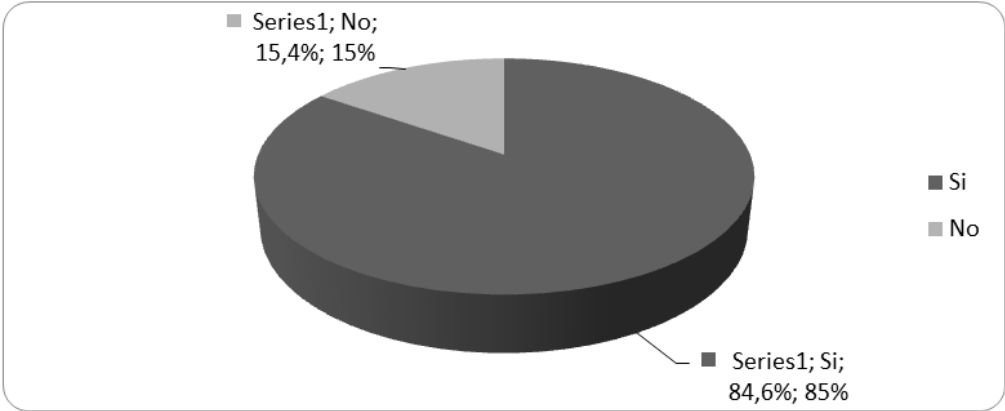
Fuente: Los Autores

Cuando se preguntó que si la empresa cuenta con herramientas para la gestión del conocimiento (Documentación, Lecciones aprendidas, entre otras), el 23,1% no cuenta con estas herramientas y el 76,9% de las empresas si cuentan con dichas herramientas; siendo las que son utilizadas la Mind – confluence – Desarrollo propio, la Información compartida en LAN, la Alifresco – openem – propio, la Propietaria, la Taiga, la Mediawiki, la de Hojas de cálculo y la GitLab wiki con un 10,0% respectivamente y la herramienta Software específico con un 20,0%.

H. EVOLUCIÓN O MEJORA DEL PRODUCTO

29. ¿Mejoran/evolucionan el producto sin necesidad que el cliente reporte los defectos?

Figura 53. Realización de mejoras y evolución por cuenta propia



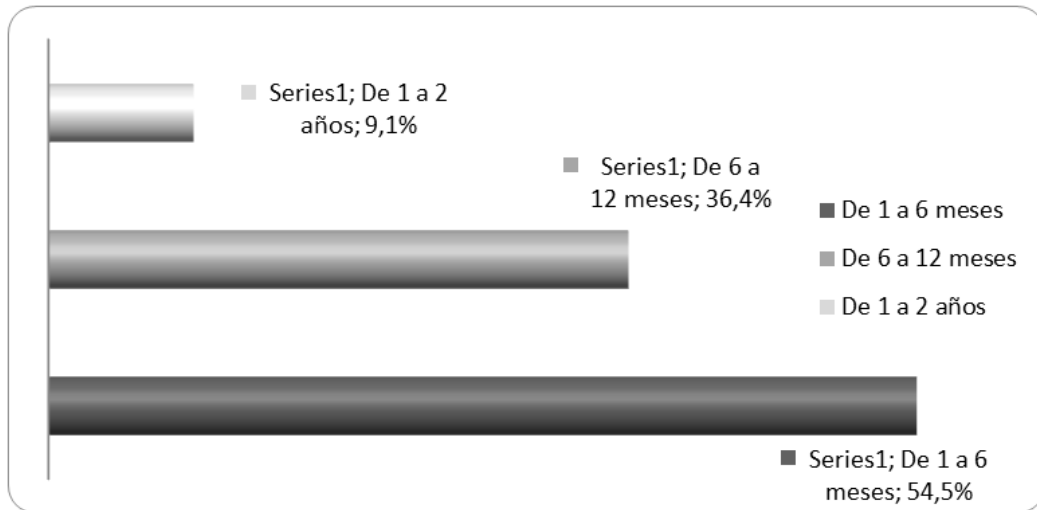
Fuente: Los Autores

El 84,6% de los encuestados mejoran/evolucionan el producto sin necesidad de que el cliente reporte los defectos, mientras que el restante 15,4% no llevan a cabo este proceso. De las empresas que mejoran/evolucionan este producto, el 54,5% lo perfeccionan (Figura

45) con una frecuencia comprendida entre 1 y 6 meses, el 36,4% lo hacen entre 6 y 12 meses y el 9,1% de 1 a 2 años.

29.1 Si respondió Si (punto 29): ¿Cada cuánto lo perfeccionan?

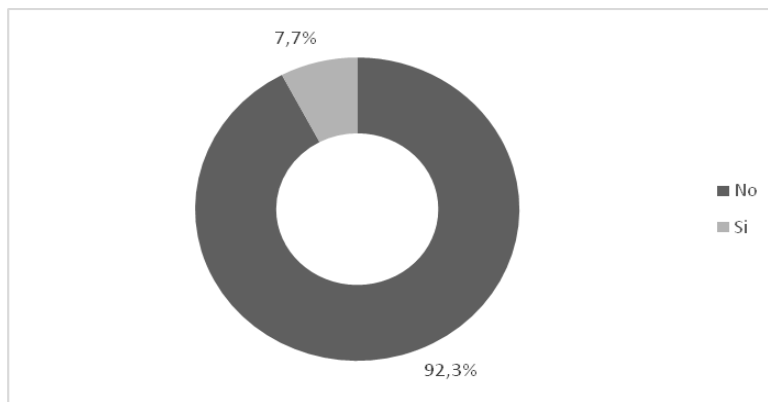
Figura 54. Tiempo de perfeccionamiento



Fuente: Los Autores

30. ¿Conoce las Leyes de Lehman, las cuáles hablan sobre la "Evolución del Software"?

Figura 55. Conocimiento de las Leyes de Lehman



Fuente: Los Autores

Por último, se les preguntó a las empresas si conocían las Leyes de Lehman, las cuales hablan sobre la “Evolución del software”, y el 92,3% expresó que no las conoce y el 7,7% dijo si conocer estas leyes.

9.1.2 Análisis de resultados de la encuesta de caracterización

Todas las empresas afirman realizar mantenimiento de software a sus productos, pero más de la mitad de ellas no siguen una guía o metodología para ello. En conversación con ellos, algunos afirman que es por desconocimiento de que exista alguna que les pueda ayudar, otros porque no les alcanza el tiempo para investigar e implementar y otros porque las que conocen son muy complejas de utilizar.

La gran mayoría de los encuestados afirman tener conocimientos sobre mantenimiento de software y, además, más de la tercera parte de ellos no tercerizan el mantenimiento, es decir que ellos mismos realizan su mantenimiento, pero no saben para qué les pueda servir una guía de mantenimiento y la forma en cómo los puede guiar y ayudar en este proceso.

La planificación del mantenimiento marca la importancia de conocer el tiempo para estimar los costos. La mayoría de las empresas realizan la planificación del mantenimiento de sus productos, sin embargo, no siempre saben cuánto puede durar esta fase de mantenimiento debido a la complejidad de este.

Un poco menos de la mitad de las empresas tienen un presupuesto solo para el mantenimiento del software, pero la gran mayoría de las empresas consultadas no estiman los costos del mantenimiento del software. Lo cual puede indicar que realmente las empresas no saben si el costo del mantenimiento fue como se tenía previsto o simplemente es una fase que no se está gestionando adecuadamente.

Las empresas saben del tiempo que les puede tomar analizar una incidencia y solucionarla. Esto indica que, ya sea por juicio de expertos o porque los tiempos los tengan registrados en alguna plataforma, tienen métricas que les pueden servir para sus

estimaciones. Sin embargo, algunas empresas no registran los defectos encontrados en alguna herramienta. Por lo tanto, se están perdiendo algunas métricas que serían de importancia para ellos.

Más de la mitad de las empresas manifiestan que los recursos, de personas y tiempo, con los que cuenta, no le son del todo suficientes para el mantenimiento del desarrollo que deben realizar.

Por otra parte, casi todas las empresas afirman que cuando corrigen un defecto resultan afectando otra parte del software. Además, no todas las empresas hacen un análisis de impacto de las correcciones o mejoras que van a realizar. Adicional, entre los errores más comunes que se presentan, según las empresas, son errores en el análisis y toma de requerimientos, improvisación por parte del equipo de mantenimiento, poco control de cambios, entre otros. Estas pueden ser razones por las cuales se reportan los defectos y que resulten afectando otras funcionalidades.

La documentación en todas las fases del desarrollo de software es muy importante, y por ende la mayoría de las empresas afirman que cada cambio o mejora que se realizan es documentada para dejar trazabilidad y evidencia de lo realizado para futuras consultas que se necesiten.

Y, por último, no todas las empresas cuentan con herramientas para registrar los defectos que se reportan, la gestión de configuración y versiones y la gestión del conocimiento. Es importante apuntarle a que estas empresas utilicen las herramientas que se tienen en el mercado, tanto libres como pagas, para mejorar su proceso de mantenimiento.

9.1.3 Conclusiones de la encuesta de caracterización

Se concluye que esta caracterización arroja como resultado que todas las empresas están dispuestas a utilizar una guía de mantenimiento, lo cual es beneficioso para este proyecto, pues estarían dispuestos a implementar la guía que sea producto de esta investigación.

Se evidencia además el desconocimiento o falta de implementación de guías o metodologías de mantenimiento por parte de las empresas, nos refuerza la idea de realizar una guía de mantenimiento sencilla de implementar y con lo realmente importante del mantenimiento de software, que pueda ser utilizada por estas. Además, marcar una ruta para establecer una guía de fácil comprensión y manejo para tener un visión del proceso de mantenimiento en el desarrollo de Sistemas de Información para las empresas de la región.

Como se menciona en el análisis, la gran mayoría de las empresas consultadas no estiman los costos del mantenimiento del software. Este resultado refuerza la idea de que MANTELASOFT, debe incluir un capítulo sobre este tema, para que las empresas tengan una base para comenzar a estimar estos costos y puedan tener un valor más ajustado de su software.

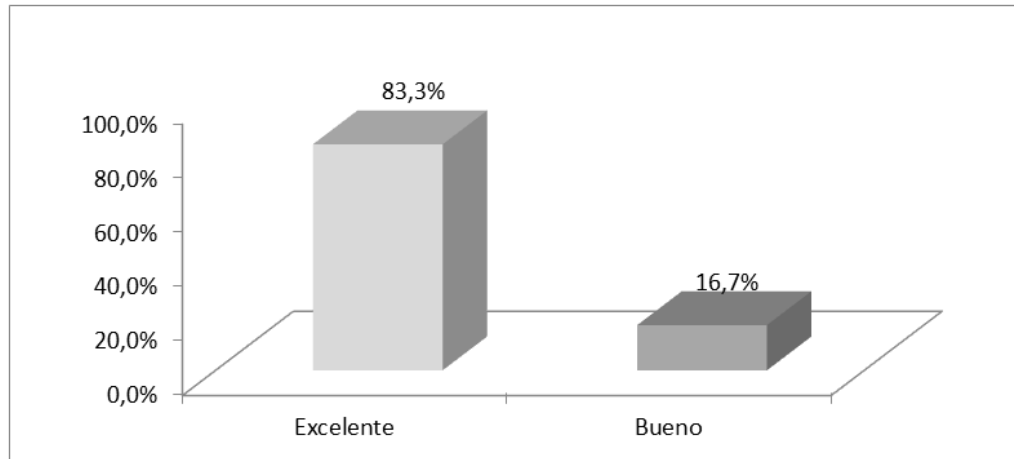
Lo anterior refuerza que las métricas de software son muy importantes para poder conocer que tan productivo y eficiente es el equipo de desarrollo de software. Como no todas las empresas tienen estos registros, se ve la necesidad de tener un capítulo dedicado a métricas que se pueden tener en cuenta en el mantenimiento de software, y así las empresas puedan hacer comparaciones de históricos para implementar acciones de mejora en sus procesos.

9.2 Tabulación y gráfico de la información Encuesta Post-Aplicación de la guía

Con base en las encuestas sobre la validación de la pertinencia del proceso para la evolución y mantenimiento de software, aplicada a los tres encargados de la empresa 1 y a los otros tres encargados de la empresa 2, se obtuvo la información requerida para ser procesada, analizada y determinar así el análisis y las conclusiones respectivas del estudio.

1. Conocimiento y dominio de los temas en la guía

Figura 56. Conocimiento y dominio de temas de la guía

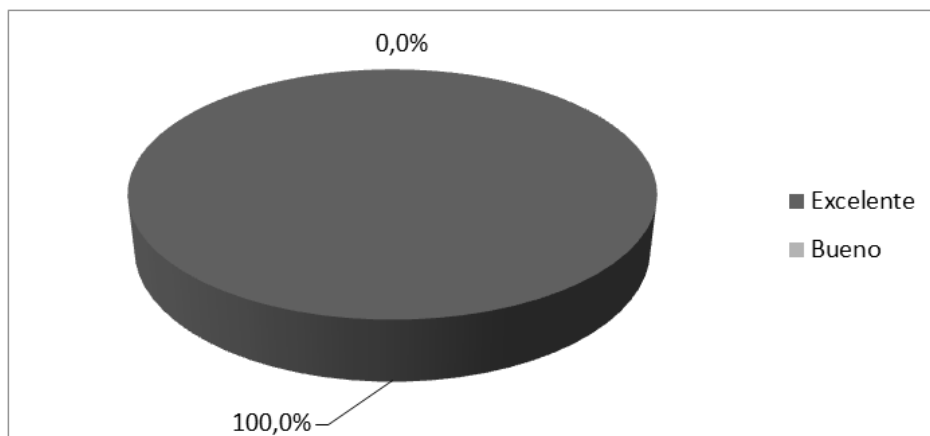


Fuente: Los Autores

El 83,3% de los encuestados califican de excelente el dominio de los temas de la guía por parte de los encargados de su aplicación en las empresas, mientras que el 16,7% califican este aspecto como bueno.

2. Claridad y orden en los puntos contenidos en la guía

Figura 57. Claridad y orden de la guía

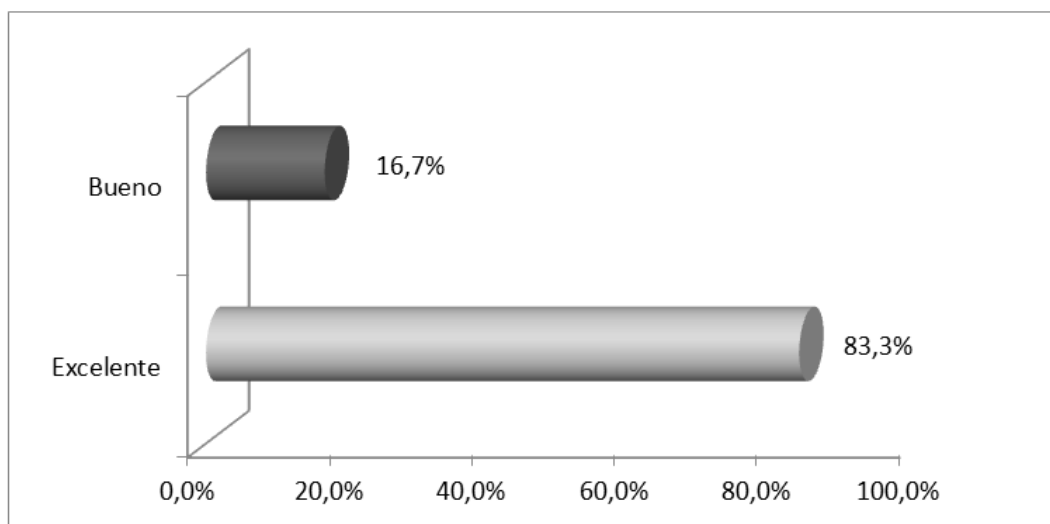


Fuente: Los Autores

El 100% de los encuestados califican de excelente la claridad y el orden propuesto en la guía MANTELASOFT.

3. Es suficiente el contenido de la guía. Conceptos y técnicas, etc.

Figura 58. Suficiencia de contenido de la guía

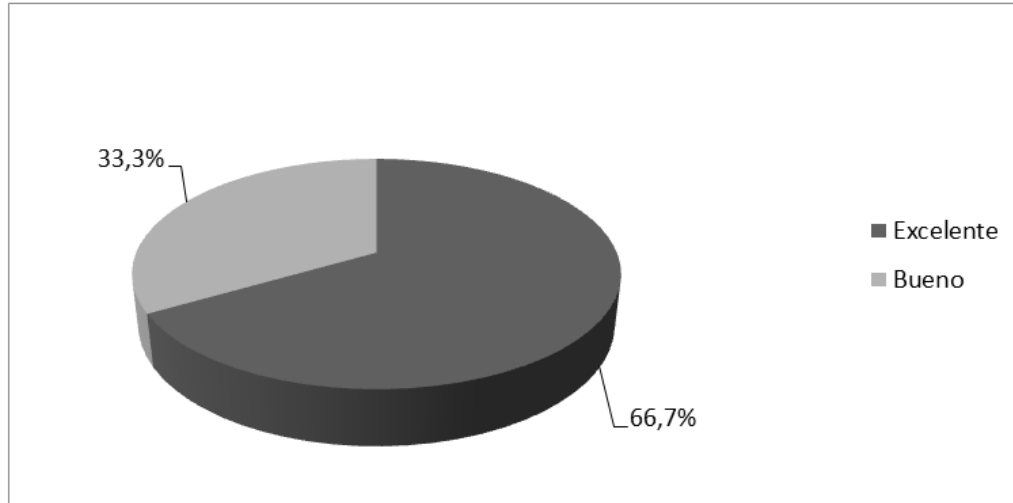


Fuente: Los Autores

Para el 83,3% de los encuestados el contenido de la guía, en cuanto a conceptos y técnicas, es suficiente para su aplicación en el mantenimiento de los desarrollos, por lo cual lo califican de excelente, mientras que el 16,7% lo califican de bueno.

4. Utilidad de toda la información de la guía

Figura 59. Utilidad de la información de la guía

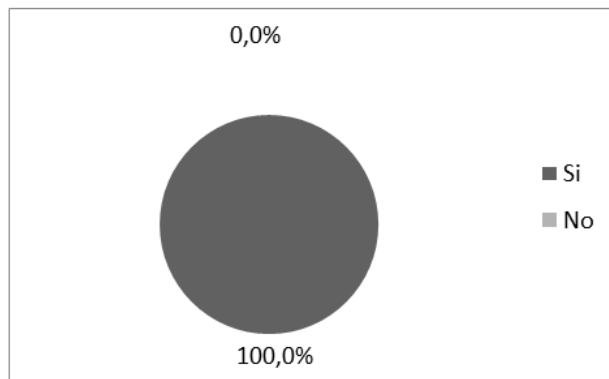


Fuente: Los Autores

Para el 66,7% de los encuestados la utilidad que les ofrece MANTELASOFT a sus productos es Excelente, mientras que para el 33,3% la calificación es bueno.

5. ¿Ejecutó la guía en su totalidad?

Figura 60. Ejecución de la guía

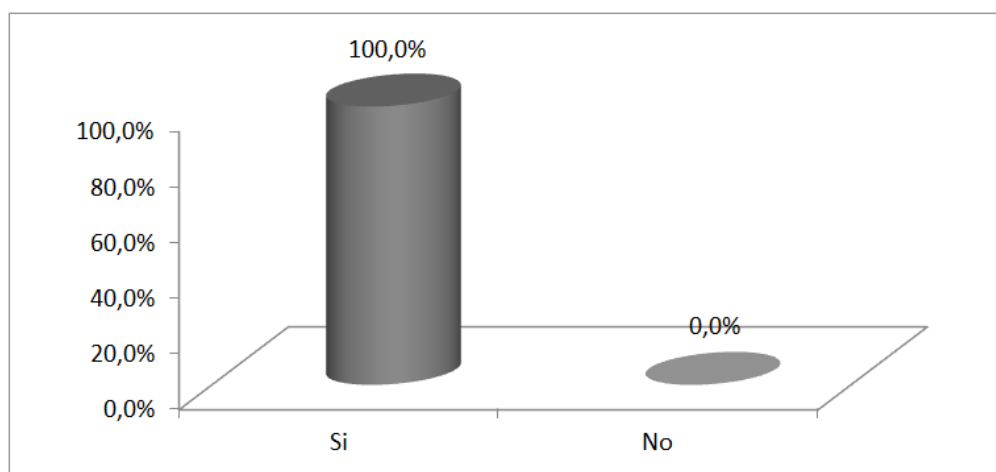


Fuente: Los Autores

El 100% de los encuestados ejecutaron todos los puntos planteados en MANTELASOFT

6. ¿El uso de la guía le ha generado aportes en la resolución de problemas?

Figura 61. Generación de aportes por la guía

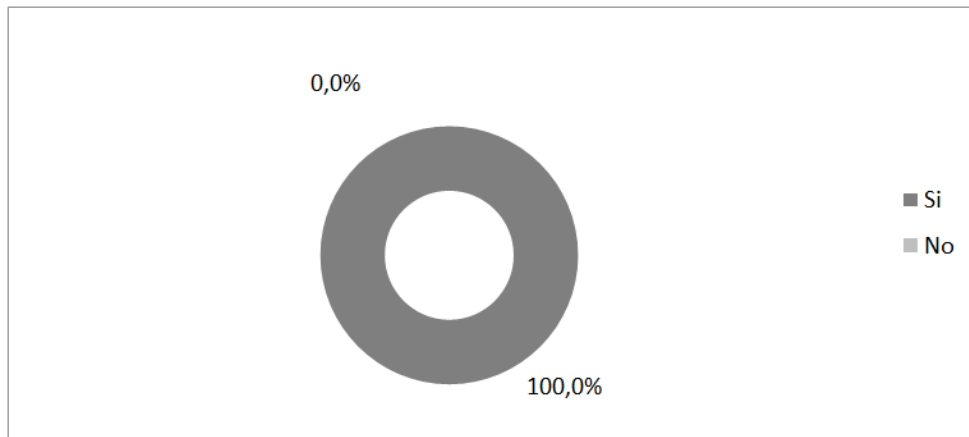


Fuente: Los Autores

El 100 % de los encuestados afirman que el uso de la guía les ha generado aportes para resolución de problemas y otros aspectos en cuanto a documentación, planificación del mantenimiento, métricas, entre otros. Como se puede evidenciar en el capítulo [Aplicación de MANTELASOFT en dos MIPYMES desarrolladoras de software](#).

7. ¿Considera que el uso de la guía de mantenimiento debe realizarse constantemente en futuras actividades del desarrollo y entrega del producto?

Figura 62. Uso de la guía en actividades futuras

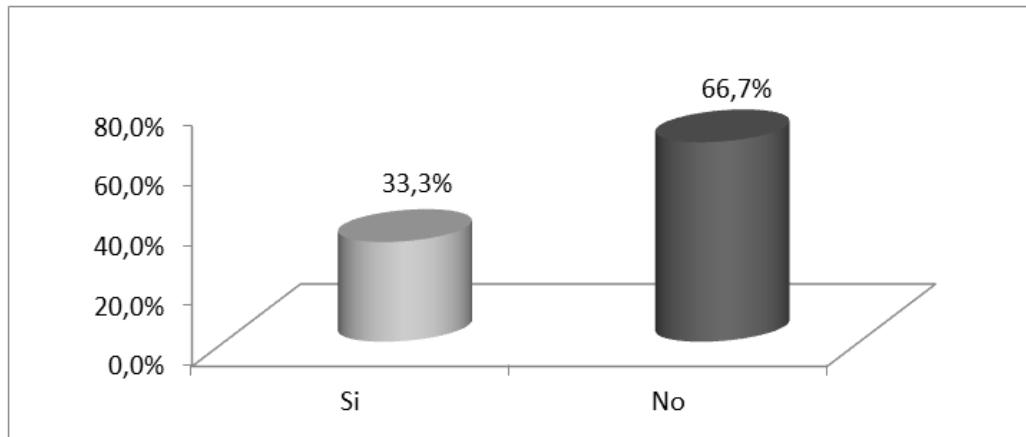


Fuente: Los Autores

El 100% coinciden que el uso de la guía se debe realizarse constantemente en actividades futuras para el mantenimiento de software.

8. ¿Considera que se deben añadir o eliminar pasos dentro del contenido de la guía?

Figura 63. Necesidad de añadir o eliminar pasos de la guía

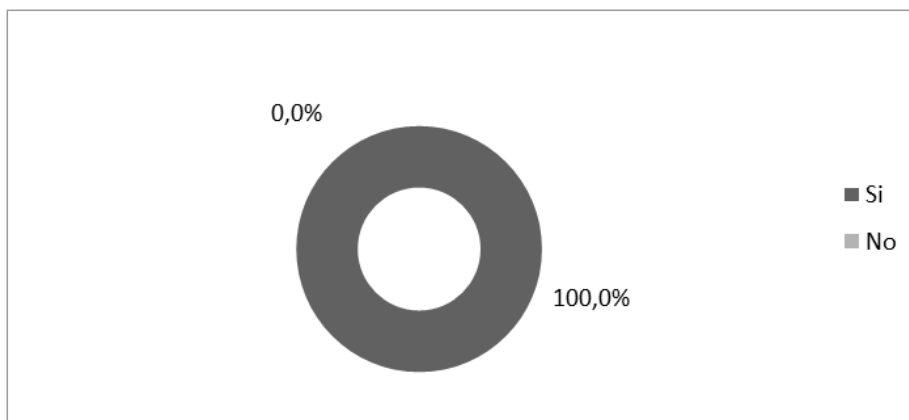


Fuente: Los Autores

El 66,7% de los encuestados mencionan que no hay necesidad de agregar o quitar pasos o elementos de MANTELASOFT, mientras que el 33,3% afirman que hay elementos que, si se deben eliminar o modificar, mas no especifican cuales.

9. ¿Cree usted que la ejecución de la guía aporta a la calidad del software?

Figura 64. Ejecución de la guía para calidad del software

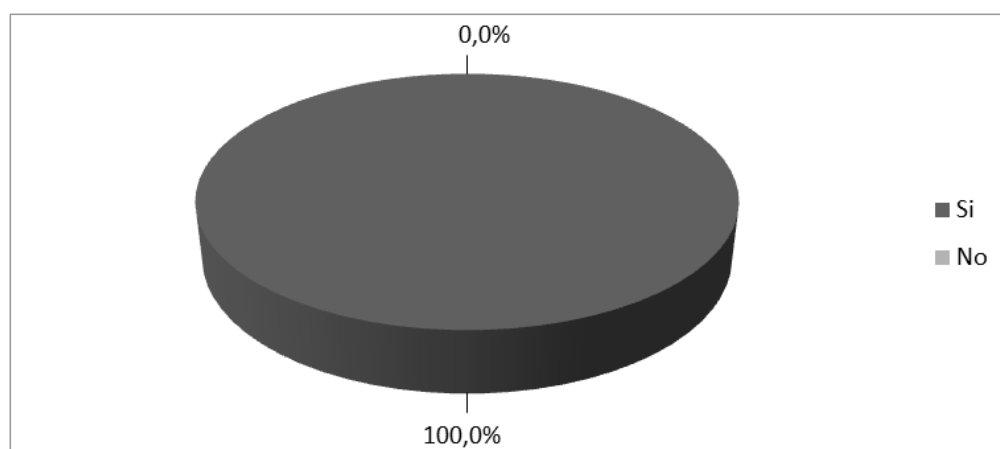


Fuente: Los Autores

El 100% de los encuestados afirman que la aplicación de MANTELASOFT aportan a la calidad de software en sus desarrollos y entregas de producto.

10. ¿La guía ayudó a encontrar problemas o situaciones que antes no se habían detectado?

Figura 65. Ayuda de la guía para encontrar problemas

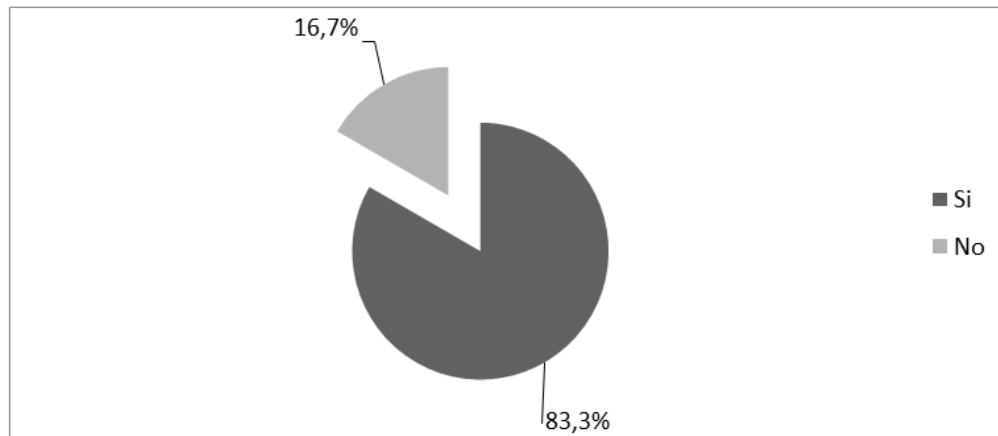


Fuente: Los Autores

El 100% de los encuestados afirman que la aplicación de la guía ayudó a encontrar situaciones que anteriormente no se habían detectado en alguna de las fases anteriores en el ciclo de desarrollo de software.

11. ¿La guía ayudó a tener mejor conocimiento de los costes del mantenimiento del producto?

Figura 66. Ayuda de la guía para el conocimiento de los costos de mantenimiento

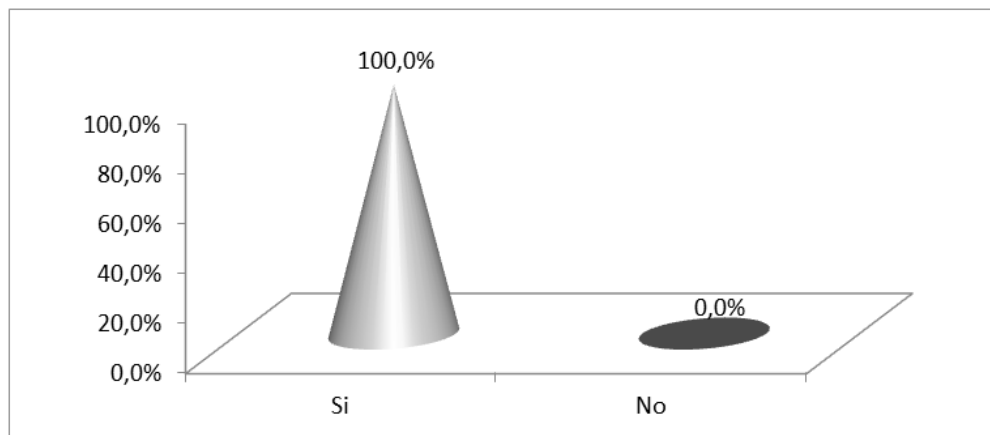


Fuente: Los Autores

El 83,3% de los encuestados afirman que MANTELASOFT les ayudó a mejorar el conocimiento de sus costes del mantenimiento de software y así tener una mejor estimación del precio; mientras que el 16,7% mencionan que no les aportó para este tema.

12. ¿Con base en el documento de mantenimiento realizado, la empresa planea realizar mejoras a sus procesos?

Figura 67. Planeamiento de la guía para realizar mejoras a los productos



Fuente: Los Autores

El 100% de los encuestados afirman que gracias a la aplicación de MANTELASOFT, la empresa planea realizar mejoras a sus productos de software.

9.2.1 Análisis de resultados de la encuesta post-aplicación de MANTELASOFT

El siguiente análisis se basa en la aplicación de MANTELASOFT a las empresas 1 y 2 y teniendo los resultados de la encuesta post, la cual fue contestada por los miembros de los equipos de desarrollo de ambas empresas.

Las dos empresas consideran que la guía es de fácil comprensión, la mayoría afirman que los temas abordados, en cuanto a conceptos y técnicas, son excelentes, la cual tiene claridad en sus atributos y un orden que permiten su entendimiento. Fue de gran utilidad para el mantenimiento de sus productos, y que seguirán utilizándola en sus productos de software para mejorar la calidad desde la fase de desarrollo. Cabe resaltar que las empresas ejecutaron todo el contenido de la guía, lo cual puede ser un indicador de lo mencionado anteriormente

Posteriormente, manifiestan que el documento aportó en un 100% a la resolución de problemas como la gestión de incidencias, decidir el tipo de mantenimiento a realizar, gestionar de mejor manera la documentación. Ambos equipos de la empresa 1 y 2 concuerdan en que seguirán utilizando la guía en sus desarrollos y que es un aporte para mejorar la calidad de sus productos.

Adicionalmente, casi todos afirman que MANTELASOFT les ayudó a tener mejor conocimiento sobre los costos del mantenimiento y, por ende, tener una mejor estimación global del producto

Y, por último, todos afirman que gracias a la aplicación de MANTELASOFT, las empresas planean realizar mejoras a sus productos.

9.2.2 Conclusiones de la encuesta post-aplicación de MANTELASOFT

Para los investigadores el resultado de la encuesta final es satisfactoria teniendo en cuenta los resultados y los comentarios generados por ambos equipos, lo cual proporciona un aporte al grupo de investigación para que socialice y difunda esta guía en las demás empresas, para que se trabaje sobre una futura versión teniendo en cuenta los comentarios y evaluaciones.

También es relevante para los investigadores saber que el trabajo realizado, para que MANTELASOFT fuera sencillo y comprensible para las empresas, pues como se mencionó en el análisis anterior, los integrantes de las empresas coincidieron que la guía tenía orden, claridad y era fácil de comprender.

Adicionalmente, es meritorio conocer por parte de las empresas que, gracias a la aplicación de la guía, planean mejorar los procesos de mantenimiento y desarrollo de software, para así poder mejorar la calidad de sus productos.

Por otra parte, algunas personas de los equipos sugieren que se realicen algunos ajustes en los ítems de personal de mantenimiento y mejoras del producto en los 4 tipos de mantenimiento, lo cual es de plasmarse en recomendaciones para que la próxima versión compare las evaluaciones finales y trabaje sobre dichos ítems, adicionando o eliminando algunos como involucrar desde el principio al cliente en el proceso. Es de anotar que no se ahondó en las observaciones sobre los elementos de la guía que se deberían añadir o eliminar, por falta de tiempo, para así mejorar el contenido de esta, pues los encuentros acordados con los equipos de desarrollo ya se habían pactado y no se logró otro encuentro adicional.

A continuación, se muestra una tabla con los resultados y productos que se esperan obtener de la realización del proyecto y quienes serían los beneficiados de este.

Tabla 16. Resultados y Productos Logrados

Aspecto	Resultado / Producto	Indicador	Beneficiarios del Resultado
Generación de nuevo conocimiento	Diagnóstico del Estado de Mantenimiento del software en la ciudad de Pereira	Informe del Diagnóstico del Estado del Mantenimiento de software en la ciudad de Pereira	Grupo de Investigación Entre Ciencia e Ingeniería de la UCP. 2 empresas de la región dedicadas al desarrollo de software
	Artículo de Investigación	Artículo de Investigación elaborado	Artículo publicado en la "Revista Facultad de Ciencias Económicas, Administrativas y Contables" de la Universidad Libre de Pereira
	Proceso de Mantenimiento de Software	Documento del proceso de Mantenimiento de Software elaborado	Grupo de Investigación Entre Ciencia e Ingeniería de la UCP. 2 empresas.

Fortalecimiento de la comunidad científica	Estado del Mantenimiento de Software en la ciudad de Pereira	Informe del Diagnóstico del Estado del Mantenimiento de software en la ciudad de Pereira	Grupo de Investigación Entre Ciencia e Ingeniería de la UCP. 2 empresas.
Apropiación social del conocimiento	Adquisición del proceso de Mantenimiento de Software en las Empresas Desarrolladoras de Software	Aplicación del proceso en (2) dos empresas de la ciudad de Pereira	2 (dos) empresas de desarrollo de Software impactadas
	Adquisición del estado del Mantenimiento de Software en las Empresas Desarrolladoras de Software	Trece (13) empresas que colaboraron con la elaboración del estudio	Departamentos de desarrollo de software de dos empresas de la región.

Fuente: Los Autores

10 DISCUSIÓN DE RESULTADOS

Esta investigación permitió a los investigadores obtener los siguientes resultados a la luz de los objetivos, la metodología y los antecedentes encontrados. En primer lugar, se hizo el levantamiento de la información y se caracterizó la forma en que las MiPymes desarrolladoras de software de la ciudad de Pereira, llevan a cabo el mantenimiento de software, dando alcance al primer objetivo propuesto. Esto permitió formular el proceso para la evolución y mantenimiento de software llamado “MANTELASOFT”, propuesta que se formuló haciendo una síntesis del proceso Ágil Mantema y las necesidades de las empresas, encontradas en la primera etapa de la presente investigación, aportando al segundo y tercer objetivo del proyecto ([ver página 7.2. Desarrollo del Proyecto](#))

Etapas posteriores se realizó la aplicación de la guía a dos de las empresas desarrolladoras de software de la ciudad de Pereira, en la cual se obtuvo una evaluación inicial y final para conocer en qué medida la guía aportaba a la mejora en el proceso de mantenimiento. Y, por último, con la encuesta-post aplicación se llevó a cabo la retroalimentación de los encargados de aplicar este documento, donde se llevaron a cabo ajustes a la guía y la valoración respectiva.

Es de anotar que la formalidad con la que se lleva a cabo los estudios encontrados en temas de mantenimiento de software a nivel internacional, nacional y regional, como se puede ver en los antecedentes, revela que los trabajos son pocos y estos no son tenidos en cuenta por la industria local del software, como lo ilustra la gráfica No. 17 y 18, donde todos los encuestados realizan mantenimiento al software, pero más de la mitad de ellos no parten de una guía o metodología.

El capítulo 8 de este trabajo resume todo el análisis y caracterización del mantenimiento en las empresas, del cual se destaca que el 100% de las empresas afirman realizar mantenimiento de software a sus productos, y están dispuestas a utilizar una guía de mantenimiento. Menos de la mitad de las empresas tienen un presupuesto para el

mantenimiento y la gran mayoría manifiesta no saber estimar los costos de este. Además, la mayoría no saben cuánto puede durar esta fase de mantenimiento debido a la complejidad de este. Las empresas conocen cuánto tiempo les puede tomar el análisis y solución de los defectos reportados, pero no todos registran esta información a través de herramientas que ayuda a su gestión.

Para la realización de MANTELASOFT fue importante haber encontrado los antecedentes correctos y las técnicas utilizadas para la caracterización del mantenimiento en las empresas, que también fue un insumo muy importante. Con estos insumos, se identificaron sus ventajas y desventajas para poder tener los argumentos suficientes para plantear la necesidad de crear la guía a la luz de las necesidades de las empresas. En esta etapa, se tuvo como resultado la creación de la guía de mantenimiento, donde se encuentra un conjunto de fases, actividades y tareas puntuales a tener en cuenta durante el mantenimiento, la cual ayuda al equipo en temas de estimación, planificación, costos, métricas, documentación y evolución o mejora.

Como producto se crea la guía de mantenimiento MANTELASOFT y un artículo publicado en revista reconocimiento nacional, Revista Facultad de Ciencias Económicas y Administrativas, No. 7 de diciembre de 2017, titulado "Las prácticas del mantenimiento de software para las MiPymes y los departamentos de desarrollo de software en la ciudad de Pereira, página 148, [ver Anexo I](#).

Se realiza la aplicación de la guía en dos empresas desarrolladoras de software, desde la tabla 4 a la 15 se evidencian los resultados obtenidos en las evaluaciones inicial y final en cada una de las empresas. Allí se puede ver un aumento en la calificación, pero más allá de la calificación, es la mejora que obtuvieron las empresas en el proceso de mantenimiento. Así las cosas, luego de la aplicación y evaluación de MANTELASOFT a las empresas, se realizó una encuesta post-aplicación para conocer qué tan pertinente fue la guía y qué mejoras resultan del proceso. Los resultados se pueden ver desde la Figura 56 a la 67. De esta retroalimentación, surgió una modificación a la misma y fue tener un capítulo

de generalidades del mantenimiento y los tres capítulos de mantenimiento que ya se tenían con actividades particulares para cada uno.

Por otro lado, la metodología utilizada por fases del Ciclo de Mejora PHVA, se ajustan a los objetivos propuestos por los investigadores; según la Figura 8, donde se muestra el diagrama que modela el proceso que se realizó y cada una de sus actividades, enmarcado en el Ciclo PHVA.

Los resultados obtenidos dan cumplimiento a satisfacción de los objetivos propuestos en el proyecto, ya que logran evidenciar la aplicación y validación de la guía “Mantelasoft” aplicada en dos MiPymes desarrolladoras de software de la ciudad de Pereira, a un grupo de desarrolladores que no tenían una técnica o guía soporte en sus procesos, el proceso de socialización o capacitación requirió de poco tiempo para su comprensión y el resultado en la curva de aprendizaje fue satisfactoria, la retroalimentación obtenida arrojó como resultado una guía fácil de utilizar y utilidad en sus productos, las dificultades y recomendaciones se tuvieron en cuenta y se dejan como propuestas para futuras versiones.

Haciendo una relación de MANTELASOFT con el principal antecedente que fue Agil_Mantema (Pino, Ruiz, & Salas, 2008), se puede denotar que la guía propuesta es en sí, un subconjunto de las principales actividades y tareas del antecedente, entre las que se encuentran “Asignar Responsables”, “Recibir petición de modificación”, “Decidir el tipo de mantenimiento”, “Analizar y elegir la solución”, “Migración y retirada del Software”; adicionando nuevos procesos, actividades y tareas que fueron “Métricas del mantenimiento”, “Estimación de Costos de Mantenimiento”, “Personal de Mantenimiento” y “Evolución y mejora del producto”, tomados de las necesidades reales de algunas empresas de desarrollo de software de la ciudad de Pereira, y que posteriormente fue validada su aplicación y conveniencia en dos de estas empresas. En contramedida al antecedente que no muestra dónde ha sido utilizada la misma.

11 CONCLUSIONES

La caracterización de las empresas y departamentos de desarrollo de software de la ciudad de Pereira muestran que el 69% de las empresas que realizan mantenimiento de software no siguen una guía o metodología para ello, debido al desconocimiento de su existencia o a lo tedioso de implementar una de ellas. Lo cual ratificó la necesidad de elaborar una guía que sea de fácil aplicación en las empresas.

El proceso de caracterización y recolección de la información en las empresas y departamentos de desarrollo de software, permitieron tener un estado del arte sobre el uso, procesos y metodologías del mantenimiento de software en las empresas de Pereira y servir como insumo para plantear MANTELASOFT. Además, también podrá servir como insumo para otros estudios.

En este ejercicio investigativo se demostró que existe un número representativo de micro, pequeñas y medianas empresas dedicadas al desarrollo, mantenimiento y soporte de software en la ciudad de Pereira que estarían dispuestas a hacer uso de la aplicación de un proceso de mantenimiento.

Se validó MANTELASOFT aplicándola en las empresas 1 y 2 de la ciudad de Pereira, las cuales presentaron una mejora en la calificación final después de realizar la evaluación final de cumplimiento de la guía MANTELASOFT.

Con la mejora en los resultados obtenidos en las dos empresas, se puede evidenciar que la guía contribuyó a la mejora del proceso de Mantenimiento de Software en cada una de ellas, lo cual también se ratifica con la encuesta post-aplicación realizada en cada una de ellas.

Gracias a la retroalimentación de las empresas, se ajustó el documento de MANTELASOFT en la fase “Actuar” del ciclo de mejora utilizado. En el cual se adicionó

un nuevo capítulo de generalidades, antes se repetían estas actividades en cada tipo de mantenimiento; y tres capítulos con actividades específicas para cada mantenimiento.

Esta investigación queda como insumo para el grupo de investigación “Entre Ciencia en Ingeniería” de la Universidad Católica de Pereira en la línea de Ingeniería de Software sobre Mantenimiento de Software para futuras investigaciones

Siguiendo cada etapa de la metodología fue factible el logro de los objetivos trazados para el proyecto, dejando como insumo la guía ajustada, el estado del arte, la información recopilada y tabulada a través de las encuestas.

Es notoria la contribución a la mejora de los procesos propios de las empresas donde se aplicó la guía.

12 RECOMENDACIONES

Se recomienda para futuros trabajos de investigación en la línea de Ingeniería del Software al grupo de investigación de la Universidad Católica de Pereira, *Entre Ciencia e Ingeniería*, seguir proponiendo proyectos de investigación que arrojen propuestas de mejoramiento para las empresas en temas de aseguramiento de la calidad en esta línea de mantenimiento.

Es importante el trabajo en las MiPymes ya que son el sector que más apoyo requiere por parte de los diferentes actores, entre ellos la academia y el Estado, para esto se propone al grupo de investigación de la Universidad Católica de Pereira, *Entre Ciencia e Ingeniería*, en conjunto con los investigadores y proyección social de la Universidad organizar un encuentro con las ocho empresas que participaron del diagnóstico, tanto en el foro como en la encuesta, para socializar los resultados de investigación y seguir recogiendo resultados que lleven al grupo a futuras investigaciones, ajustes y otras versiones de la guía.

Entregar a las empresas que participaron en la implementación de la guía de mantenimiento de software lo siguiente,

- Encuesta de diagnóstico.
- Resultado de la encuesta
- Guía de mantenimiento
- Lista de chequeo de la aplicación de la guía.
- Informe de análisis del proceso de implementación de la guía y un informe propuesto de los ajustes a realizar en el equipo de mantenimiento, para que sea estudiado por los ingenieros del área de mantenimiento.
- Informe de acciones encaminadas a la mejora del modelo en sus fases, actividades, métricas, técnicas y herramientas para ser utilizadas según las sugerencias que para bien pudieran aportarse por los responsables de su aplicación en las empresas, las cuales permitirían elevar la calidad de la guía de mantenimiento de software propuesta.

Como este grupo de investigación lleva trabajando 12 años en línea de Ingeniería de Software y específicamente en temas de aseguramiento de la calidad, y que a futuro se tenga un centro que brinde apoyo a los empresarios en esta línea con asesorías y consultorías en temas de requerimientos, diseño y testing, se propone que para versiones futuras el grupo de investigación de la Universidad Católica de Pereira, *Entre Ciencia e Ingeniería*, continúe evaluando el uso de la guía Mantelasoftware o una mejora a la primera propuesta, con las recomendaciones sugeridas por los equipos de ingenieros de las áreas de mantenimiento de las empresas desarrolladoras de software.

Con los resultados de las investigaciones anteriores, se propone publicar un libro que contengan capítulos en temas de requerimientos, diseño, mantenimiento y testing.

Se recomienda que para trabajos futuros o actualizaciones y dada la naturaleza de la información y los datos que se tuvieron en cuenta para el estudio, es muy probable que hayan cambiado a la fecha de terminación del trabajo. También que se profundice en algunas respuestas que se dieron en la caracterización del mantenimiento de software, pues luego de aplicada la encuesta y analizada la información, quedaron más preguntas por realizar.

13 BIBLIOGRAFÍA

- Alain, A., Huffman Hayes, J., Alain , A., & Reiner , D. (2004). Software Maintenance Maturity Model (SMmm): The software maintenance process model. *J. Softw. Maint. And Evolution. No. of Figures: 4. No. of Tables: 7. No. of References: 117*, 19-39.
- April, A., Huffman, J., Abran, A., & Dumke, R. (2004). *Software Maintenance Maturity Model (SMmm): The software maintenance process model*. Kentucky, USA: John Wiley & Sons, Ltd.
- Arbeláez, M. A. (2000). *Las MIPYMES en Colombia: diagnostico*. Bogotá D.C.: Fedesarrollo.
- Becerra Sanchez, L. Y. (2007). *Propuesta metodológica para estandarizar el proceso de construcción y evaluación del producto software que permita a la MIPYMES medir la calidad del software*. Pereira: UCPR.
- Bennett, K., & Rajlich, V. (2000). *Software Maintenance and Evolution: a Roadmap*.
- Bohem, B. T. (2003). *Balancing Agility and Discipline - A Guide for the Perplexed*. Addison-Wesley.
- Bwch, E., & Kung, H.-J. (1997). Modeling Software Maintenance Requests: A Case Study. 41-42.
- Cohen, D., Lindvall, M., & Costa, P. (2003). *Agile Software Development*.
- Diez Canseco, O., & Mejia Figueroa, E. A. (s.f). *Diseño de una mejora continua en el area de operaciones de la empresa Courier Enlace Correos S.A.,utilizando la metodología PHVA*. Peru.
- Grubb, P., & Takang, A. A. (2003). *Software Maintenance, Concepts and Practice* (Segunda ed.).
- Gutierrez, J. (2007). *Metodologías Ágiles*.
- Herrera Caldas, V. A. (2015). *Desarrollo de un plan de gestión de mantenimiento de software para el Departamento de Sistemas de la Universidad Politécnica Salesiana basado en la Norma ISO/IEC 14764:2006*. Cuenca, Ecuador.
- IEEE. (1998). *1219: IEEE Standard for Software Maintenance*.

- IEEE. (2006). *IEEE 14764-2006 Software Engineering - Software Life Cycle Processes - Maintenance*.
- IEEE Computer Society. (2006). *IEEE Standard Dictionary of Measures of the Software Aspects of Dependability*. New York, NY: the Institute of Electrical and Electronics Engineers.
- IEEE Computer Society. (2014). *SWEBOK V3*.
- ISO/IEC. (2007). *ISO/IEC 14143: Information technology — Software measurement — Functional size measurement*.
- ISO/IEC. (2008). 12207 - Information Technology/ Software Life Cycle Processes.
- ISO/IEC. (2016). 29110 - Systems and software engineering - Lifecycle profiles for Very Small Entities (VSEs).
- Jansson, A.-S. (2007). *Software Maintenance and Process Improvement by CMMI*.
- Jones, C. (2006). *The Economics of Software Maintenance in the Twenty First Century*.
- Jones, C. (2008). *Estimación de costos y administración de proyectos de Software*. McGraw-Hill Interamericana.
- Jones, C., & Bonsignour, O. (2012). *The Economics of Software Quality*. Boston: Pearson.
- Joyanes Aguilar, L., Cuesta Meza, A., & López Trujillo, M. (2010). Caracterización de la industria del software en el Triangulo del Café - Colombia. *Entre Ciencia e Ingeniería, ISSN 1909-8367, Año 4. No. 7, 76-87*.
- Lehman, M. M., Ramil, J. F., Wernick, P. D., Perry, D. E., & Tursky, W. M. (1997). Metrics and laws of software evolution - the nineties view. En Proceedings of the 4th International Symposium on Software Metrics, METRICS '97. *IEEE Computer Society*, 20-32.
- Lehman, M. P. (1998). *Implications of Evolution Metrics o Software Maintenance*. EE.UU.: khoshgoftaar y Bennet.
- Martínez Marín, S. J., Arango Aramburo, S., & Robledo Velásquez, J. (2015). el crecimiento de la industria del software en Colombia: un análisis sistémico. *Revista EIA, ISSN 1794-1237 / Año XII / Volumen 12 / Edición N.23, 95-116*.
- Merchán, L., & Urrea, A. (2007). *Caracterización de las empresas pertenecientes a la industria emergente de software del sur occidente colombiano Caso red de parques*

- PARQUESOFT. Obtenido de Avances en sistemas e informática: <http://www.revistas.unal.edu.co/index.php/avances/article/viewArticle/9760>
- Ministerio de Administraciones Públicas de España. (2001). *Métrica V3*.
- Ministerio de las Tecnologías de la Información y la Comunicación. (2015). *Informe de caracterización del sector de software y tecnologías de la información en Colombia*. Bogota D.C.: departamento de Planeación Nacional.
- Mitchell, J. (2004). *Concepts in Programming Languages, 1st. Edition*. Cambridge: Cambridge University Press.
- Nasir, Z., & Abbasi, A. (2010). Framework for Software Maintenance and Support Phase. *Computer & Emerging Sciences*, 4.
- Osorio Martínez, Z., Irrazabal, E., Garzás, J., & Marcos, E. (2015). Aproximación a la Mejora de la Metodología Agil Mantema para la Adopción del Modelo de Calidad ISO/IEC 15504-ISO/IEC 12207:2008 en PYMEs. *Instituto Tecnológico de Orizaba, Veracruz, México*, 4.
- Pardo, C., Hurtado, J. U., & Collazos, C. (2010). Mejora de procesos de software ágil con agile - SPI Process. *DYNA*.
- Peláez Valencia, L. E., Toro Lazo, A., López Restrepo, J. F., & Ramírez Medina, A. (2012). Caracterización del proceso de desarrollo de software en Colombia: una mirada desde las Pymes productoras. *Revista Académica e Institucional, Páginas de laUCP, N° 92*, 89-98.
- Pino, F. J., García, F., & Piattini, M. (2006). Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas. *Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.2, No. 1*, 6-23.
- Pino, F., Ruiz, F., & Salas, S. (2008). *Informe Técnico: IT 23. Competisof. Agil Mantema*.
- Polo Usaola, M. (2000). *MANTEMA: una metodología para mantenimiento de software*. Castilla -La Mancha.
- Polo, M., Piattini, M., Ruiz, F., & Calero, C. (2000). MANTEMA: a Software Maintenance Methodology Based on the ISO/IEC 12207 Standard. *ALARCOS*, 77-80.
- Pressman, R. S. (2008). *Software Engineering: A Practitioner's Approach*.

- Pressman, R. S. (2010). *Ingeniería del Software - Un enfoque Práctico - 7° Edición*. México: Mc Graw Hill.
- Procolombia. (2014). *Oportunidades en la Industria de Servicios y Software & T.I.* Obtenido de <http://inviertaencolombia.com.co/sectores/servicios/software-y-servicios-de-ti.html>
- Ramírez Aguirre, P. A., & Ramírez Arias, C. (2010). *Estudio de las prácticas de la calidad del software implementadas en las Mipymes desarrolladoras de software de Pereira*. Pereira, Colombia: Universidad Tecnológica de Pereira.
- Singh, Y., & Goel, B. (2007). *A Step Towards Software Preventive Maintenance*. Delhi.
- Sneed, H. M., & Brössler, P. (2003). Critical Success Factors in Software Maintenance A Case Study. *Computer Society*, 8-17.
- Sommerville, I. (2011). *Ingeniería del Software* (Novena ed.). México: Pearson.
- Tripathy, P., & Naik, K. (2015). *Software Evolution And Maintenance. A practitioner's Approach*. Hoboken, New Jersey, Estados Unidos: Wiley.
- Xiaoping, W., Fang, Z., & Yi, S. (2014). *An Optimal Software Maintenance Policy Based on Reliability and Risk*. Beijing, China: Neural Networks (IJCNN).
- Yongchang, R., & Zhongjing, L. (2011). Software Maintenance Process Model and Contrastive Analysis. *Science and Technology*, 171.

14 ANEXOS

Anexo A. Encuesta para el diagnóstico inicial del Mantenimiento de Software

INSTRUMENTO PARA IDENTIFICAR ASPECTOS DE MANTENIMIENTO EN LAS EMPRESAS DESARROLLADORAS DE SOFTWARE EN LA CIUDAD DE PEREIRA					
OBJETIVO	Caracterizar las prácticas del Proceso de Mantenimiento de Software en las Empresas Desarrolladoras de Software de la ciudad de Pereira.				
Notas:					
1. Marque con una "X" la respuesta que considere aplique para su empresa.					
2. Cuando se pida su opinión, siéntase libre de escribir los detalles como lo considere necesario. No hay respuestas correctas o incorrectas.					
3. La información recogida durante esta encuesta será confidencial y no será difundida con colegas, superiores o terceros sin su permiso.					
4. La información es sólo para propósitos de investigación y no será publicada, de tal manera que ningún participante pueda ser identificado. Además, su nombre no será registrado.					
5. Acudimos a ustedes, para que entre todos encontremos la forma de diagnosticar, proponer y describir mejoras en pro de tener un proceso de mantenimiento de software en las empresas desarrolladoras en la ciudad de Pereira.					
ENCUESTA					
A. GENERALIDADES DEL MANTENIMIENTO					
1. ¿Años de creación de la empresa?					
1 a 5		5 a 10		10 ó más	
2. ¿Principal actividad de la empresa? Seleccione una opción					
Desarrollo de software		Soporte de Software		Mantenimiento de Software	
Otro. ¿Cuál?:					
3. ¿Tiene la empresa certificación de calidad?					
Sí ¿Cuál(es)?:				No	
4. ¿La empresa realiza mantenimiento al software que desarrolla?					
Sí		No			
Si la respuesta anterior es "Sí", responder la siguiente pregunta:					
4.1 ¿Tiene como base o modelo alguna guía o norma para el mantenimiento del software?					
Sí ¿Cuál(es)?:				No	
Si la respuesta de la pregunta 4.1 es "No", responder esta pregunta.					
4.2 En caso de no aplicar una guía de mantenimiento ¿Conoce para qué sirven y la forma en que puede ayudar a la empresa?					
Sí		No			
5. ¿Tiene conocimientos sobre Mantenimiento de Software?					
Sí		No			
6. ¿Qué tipo de mantenimiento de software realiza la empresa? (Puede seleccionar varias)					
Preventivo (Cambios que llevan a prevenir problemas antes de que ocurran)		Correctivo (Reparar errores de código, errores de diseño, errores de requerimientos)		Adaptativo (Los cambios debido a cambios en el entorno del sistema (HW, SO, otro SW de soporte))	
Perfectivo (Adicionar o modificar funcionalidades del sistema)					
7. De considerar útil la aplicación de un proceso de mantenimiento ¿haría uso de ella en su empresa?					
Sí		No			
8. ¿Realizan tercerización del mantenimiento del Software de sus productos?					
Sí		No			
9. ¿Dan garantía del SW? De ser afirmativo, ¿Cuánto tiempo?					
Sí		Año(s)		No	

B. PLANIFICACIÓN DEL MANTENIMIENTO			
10. ¿La empresa planifica las operaciones y recursos que se van a realizar en el mantenimiento del software?			
Si		No	
Si la respuesta anterior es "Si", responder las siguientes preguntas:			
10.1 ¿Se realiza un plan de mantenimiento durante el desarrollo del SW?			
Si		No	
10.2 ¿En la planificación sabe cuánto va a durar el mantenimiento del Software?			
Si		No	
C. ESTIMACIÓN DE COSTOS			
11. ¿Tiene un presupuesto aparte solo para el mantenimiento de software?			
Si		No	
12. ¿La empresa estima los costos del mantenimiento del software?			
Si		No	
Si la respuesta anterior es "Si", responder las siguientes preguntas:			
12.1 ¿Para estimar el costo que técnica utiliza?			
Paramétrico		Matemático	
			Juicio de Expertos
Otro. ¿Cuál?			
12.2 ¿Tiene conocimiento de cuánto cuesta el mantenimiento al producto?			
Si		No	
12.3 ¿Tiene en cuenta los costos de operación del Hardware donde va a funcionar el producto de Software?			
Si		No	
D. MÉTRICAS DEL MANTENIMIENTO			
13. ¿Cuánto tiempo promedio dedica el desarrollador para entender el cambio, mejora o corrección a realizar?			
Menos de 2 horas		De 2 a 5 horas	
			Más de 5 horas
14. ¿Cuánto tiempo promedio se demora en solucionar un defecto?			
Menos de 4 horas		De 4 a 9 horas	
			Más de 9 horas
15. ¿Registran cada uno de los defectos encontrados?			
Si		No	
Si la respuesta anterior es "Si", responder la siguiente pregunta:			
15.1 ¿Cuántos defectos se corrigen al producto por mes?			
De 0 a 10		De 11 a 30	
			De 31 a 60
			Más de 60
16. ¿Cómo miden la calidad del código, para tener una buena mantenibilidad sobre el mismo?			
17. ¿Sabe cuál es el Índice de Madurez de su Software (IMS)?			
Si		No	
E. PROCESO DE MANTENIMIENTO			
18. ¿Tiene suficientes recursos (tiempo y personal) para llevar a cabo las tareas de mantenimiento?			
Si		Más o menos	
			No
19. ¿Realiza un análisis de impacto de la corrección o mejora a realizar en el producto?			
Si		No	
Si la respuesta anterior es "Si", responder la siguiente pregunta:			
19.1 ¿Quién realiza el análisis?			
Líder de Desarrollo		Líder de Mantenimiento	
			Jefe del Departamento de Sistemas
Desarrollador de turno		Otro. ¿Quién?	
20. ¿Quién realiza las correcciones a los defectos reportados?			
Desarrollador de turno		Desarrollador quien realizó la parte afectada	
			Personal dedicado a mantenimiento
Otro. ¿Quién?			
21. ¿Le ha pasado que corrige algún defecto en el software y resulta dañando otra parte del software?			
Si		No	
22. ¿Realizan un plan de retiro del producto?			
Si		No	
23. ¿Han hecho el retiro de algún SW? De ser correcto, ¿Cuántos productos han retirado (número)?			
Si		(n) Productos	
			No
24. ¿Documentan el cambio, mejora o defecto a realizar, sus pruebas, solución y las opciones de implementación?			
Si		No	
25. Por favor indique tres de los errores más comunes que se presentan durante el mantenimiento de Software			
1.			
2.			
3.			

F. PERSONAL DE MANTENIMIENTO			
Para las preguntas a continuación, indique si está de acuerdo o en desacuerdo con las afirmaciones que se presentan. (1 = "está en total desacuerdo"; 5 = "está totalmente de acuerdo")			
AFIRMACIÓN		CALIFICACIÓN	
Es importante que el equipo de mantenimiento se encuentre dentro del desarrollo			
Es importante que el equipo de mantenimiento tenga buen conocimiento de la aplicación			
Es importante que el equipo de mantenimiento tenga buen conocimiento en técnicas y herramientas			
Es importante que el equipo de mantenimiento tenga buen conocimiento en lenguajes de programación			
Es importante que el equipo de mantenimiento se encuentre estable en la empresa			
Es importante que el equipo de mantenimiento se encuentre motivado			
Es importante que el equipo de mantenimiento reciban retroalimentación de la calidad de su trabajo			
G. TÉCNICAS Y HERRAMIENTAS DE MANTENIMIENTO			
26. ¿Cuenta con herramientas para registrar los defectos que se reportan?			
Si		Cuál(es)	
No			
27. ¿Cuenta con herramientas para la gestión de la configuración y versiones?			
Si		Cuál(es)	
No			
28. ¿Cuenta con herramientas para la gestión del conocimiento (Documentación, Lecciones aprendidas, etc.)?			
Si		Cuál(es)	
No			
H. EVOLUCIÓN O MEJORA DEL PRODUCTO			
29. ¿Mejoran/evolucionan el producto sin necesidad que el cliente reporte los defectos?			
Si		No	
Si la respuesta es sí, responder la siguiente pregunta:			
29.1 ¿Cada cuánto lo perfeccionan?			
De 1 a 6 meses		De 6 a 12 meses	De 1 a 2 años
De 3 a 5 años			
30. ¿Conoce las Leyes de Lehman, las cuáles hablan sobre la "Evolución del Software"?			
Si		No	

Anexo B. Formato de la Encuesta de Evaluación Inicial y Final de la aplicación de MANTELASOFT

EVALUACIÓN INICIAL EL CUMPLIMIENTO DE LA GUÍA DE MANTENIMIENTO DE SOFTWARE "MANTELASOFT"											
MANTENIMIENTO CORRECTIVO URGENTE											
FASE	No.	TAREA	RESUESTA					PUNTAJE POR PREGUNTA	PUNTAJE POR FASE	PUNTAJE POR CUMPLIMIENTO DE MANTELASOFT	INTERPRETACIÓN
			No sabe	No se cumple	Se cumple sin registros	Se cumple con algunos registros	Se cumple en alto grado de registros				
			0	1	2	3	4	5			
GENERALIDADES DEL MANTENIMIENTO	1	Recoger la información sobre solicitud de los servicios de mantenimiento									
	2	Preparar la propuesta de mantenimiento para el cliente									
	3	Elaborar el contrato de prestación de servicios de mantenimiento									
PLANIFICACIÓN DEL MANTENIMIENTO	1	Elaborar el cronograma de actividades									
	2	Adquirir conocimiento de la aplicación: Verificar el estado del software - Estudiar la documentación, el código de los programas, referencias cruzadas y su funcionamiento									
	3	Desarrollar planes: Estimar recursos necesarios (tiempo y personal) - Inventariar el sistema de información - Definir las herramientas para registrar los defectos que se reportan, para la gestión de la configuración y versiones y para la gestión del conocimiento (documentación, lecciones aprendidas, entre otras)									
	4	Definir procedimientos de petición de modificación									
	5	Preparar los entornos de pruebas: Realizar copias del entorno software para el trabajo									
	6	Recibir la petición de modificación: Recibir petición y radicarla									
	7	Decidir el tipo de mantenimiento requerido: Rechazar la petición, justificando la razón o aceptar la petición - Decidir las actividades de mantenimiento a aplicar									
PROCESO DE MANTENIMIENTO	1	Investigar y analizar las causas del error: Analizar la petición - Verificar el problema - Estudiar alternativas para implementar modificación - Listar elementos software a corregir									
	2	Realizar las acciones correctivas: Identificar las rutinas y bases de datos afectadas por la intervención									
	3	Diligenciar la documentación exigida: Documentar los cambios realizados									
	4	Ejecutar las pruebas unitarias: Comprobar la corrección de los cambios realizados									
	5	Pasar a producción el software: Enviar al entorno de producción el software corregido para su utilización									
MÉTRICAS DEL MANTENIMIENTO	1	Desarrollar Planes									
	2	Tiempo dedicado a la tarea									
	3	Número de puntos-función									
	4	Número de módulos									
	5	Número de líneas de código									
	6	Complejidad ciclomática media de cada módulo									
	7	Número de variables globales (visibles externamente) de cada módulo									
	8	Número de rutinas públicas (visibles externamente) de cada módulo									
	9	Decidir Tipo de Mantenimiento									
	10	Petición aceptada o rechazada									
	11	Causa de la petición: Cambios legales - Evolución del negocio - Mejora del proceso - Cambios en las políticas de negocio - Otra (¿cuál?)									
	12	Área o dominio funcional que ha presentado la petición									
	13	Investigar y Analizar Causas									
	14	Tiempo dedicado a la tarea									
	15	Número de puntos-función que previsiblemente se verán afectados por el error									
	16	Origen del error: Cambio en el análisis de requisitos - Cambio en el análisis de localización - Cambio en el análisis de diseño - Codificación									
	17	Causa del error: Desconocimiento del dominio de la aplicación - Desconocimiento del diseño o la implementación del sistema - Requisitos ambiguos o incompletos - Error sintáctico o semántico del lenguaje de programación - Tensión en la planificación - Fallo sistema encubierta - Discuido.									
	18	Investigar Acciones Correctivas									
	19	Tiempo dedicado a la tarea									
	20	Número de puntos-función añadidos									
	21	Número de puntos-función modificados									
	22	Número de puntos-función borrados									
	23	Número de módulos alterados									
	24	Número de líneas de código añadidas									
	25	Número de líneas de código modificadas									
	26	Número de líneas de código borradas									
	27	Número de defectos corregidos al producto por mes									
	28	Complejidad ciclomática media de cada módulo alterado									
	29	Número de variables globales (visibles externamente) de cada módulo									
	30	Número de rutinas públicas (visibles externamente) de cada módulo									
	31	Medición de la calidad del código									
	32	Índice de madurez del software									
	33	Análisis de impacto de la corrección									
	34	Número de retrabajos de software realizados									
	35	Cada cuánto perfeccionan el producto software									
ESTIMACIÓN DE COSTOS DE MANTENIMIENTO	1	Diligenciar Documentación									
	2	Tiempo dedicado a la tarea									
	3	Número de páginas añadidas a la documentación, según el estándar de la Organización									
	4	Ejecutar las pruebas unitarias									
PERSONAL DE MANTENIMIENTO	1	Pasar a producción									
	2	Tiempo dedicado a la tarea									
EVOLUCIÓN O MEJORA DEL PRODUCTO	1	Estimar los costos del mantenimiento del software: Definir la técnica utilizada									
	2	Identificar la técnica para estimar los costos de mantenimiento: Paramétrico - Matemático - Juicio de Expertos - Otro (¿Cuál?)									
	3	Definir el costo de operación del hardware donde va a funcionar el producto de Software									
	4	Determinar el costo total del mantenimiento al producto									
ACTIVIDADES FINALES DEL MANTENIMIENTO	1	Verificar si el equipo de mantenimiento: Se encuentra dentro del desarrollo - Tiene conocimiento de la aplicación - Tiene conocimientos en el uso de técnicas y herramientas - Tiene conocimiento en lenguajes de programación - Está laboralmente estable en la empresa - Se encuentre motivado - Recibe retroalimentación sobre la calidad de su trabajo									
	1	Registrar intervención									
	2	Actualizar base de datos: Incorporar las métricas a la base de datos histórica									
	3	Ejecutar paralelo: Realizar operaciones reales sobre el software que se va a retirar y el software nuevo (si se sustituye)									
	4	Notificar retirada: Notificar al Cliente la retirada del software									
	5	Almacenar datos del entorno antiguo									
	6	Entrega del inventario y de la documentación: Entregar al cliente todos los productos software generados y modificados									
	7	Traspaso de experiencia y formación: Enseñar las nuevas características del sistema al cliente									
8	Cesión definitiva del servicio: Finalizar la prestación de los servicios al Cliente										

Anexo C. Formato de la Encuesta de Evaluación Post Aplicación de MANTELASOFT

EVALUACIÓN PARA LA APLICACIÓN DEL PROTOTIPO DE GUÍA DE MANTENIMIENTO									
OBJETIVO	Identificar la calidad del contenido, los puntos fuertes y los puntos de mejora del prototipo de la guía de mantenimiento, aplicada en uno de los desarrollos o productos de dos empresas desarrolladoras de software o departamentos de sistemas.								
ENCUESTA									
Calificación:									
Excelente	5	Bueno	4	Aceptable	3	Deficiente	2	Malo	1
CONTENIDO DE LA GUÍA									Cal.
Conocimiento y dominio de los temas en la guía									
Claridad y orden en los puntos contenidos en la guía									
Es suficiente el contenido de la guía. Conceptos, técnicas, herramientas, etc.									
Utilidad de toda la información de la guía									
1. ¿Ejecutó la guía en su totalidad?									
Sí		No							
Si la respuesta anterior, es No, explique el por qué?									
2. ¿El uso de la guía le ha generado aportes en la resolución de problemas?									
Sí		No							
Si la respuesta anterior es Sí, describa cuáles?									
3. ¿Considera que el uso de la guía de mantenimiento debe realizarse constantemente en futuras actividades del desarrollo y entrega del producto?									
Sí		No							
4. ¿Considera que se deben añadir o eliminar pasos dentro del contenido de la guía?									
Sí		Cuáles?							
No									
5. ¿Cree usted que la ejecución de la guía aporta a la calidad del software?									
Sí		No							
6. ¿La guía ayudó a encontrar problemas o situaciones que antes no se habían detectado?									
Sí		No							
7. ¿La guía ayudó a tener mejor conocimiento de los costes del mantenimiento del producto?									
Sí		No							
8. ¿Con base en el documento de mantenimiento realizado, la empresa planea realizar mejoras a sus procesos?									
Sí		No							

Anexo D. Costo total de la investigación

1. Costo Global

RUBRO	VALOR (\$)
PERSONAL	60'048.600
MATERIALES	3'445.500
VIAJES	220.000
SALIDAS DE CAMPO	25.200
TOTAL	\$63.739.300

Fuente: Los Autores

2. Gastos de Personal

PERSONAL	FORMACIÓN ACADÉMICA	DEDICACIÓN	VALOR UNITARIO (\$)	CANTIDAD	VALOR TOTAL (\$)
INVESTIGADOR PRINCIPAL	INGENIERO DE SISTEMAS Y TELECOMUNICACIONES	15 meses	1.000.000	2	30.000.000
ASESOR DEL PROYECTO	MSC. JAIRO IVÁN VÉLEZ BEDOYA	13 meses	2.000.000	1	26.000.000
PRACTICANTES	INGENIERO DE SISTEMAS Y TELECOMUNICACIONES	2 meses	700.000	2	2.800.000
PARES EVALUADORES		2 meses	624.300	1	1.248.600
TOTAL					\$60.048.600

Fuente: Los Autores

3. Materiales, suministros y bibliografía

MATERIAL	VALOR UNITARIO (\$)	CANTIDAD	VALOR TOTAL (\$)
FOTOCOPIAS	50	100	5.000
HOJAS DIGITADAS	150	150	22.500
COMPUTADORES	1.200.000	2	2.400.000
USO COMPUTADOR (horas)	500	1.280	640.000
USO INTERNET (horas)	500	640	320.000
EMPASTADOS	10.000	1	10.000
ARGOLLADOS	6.000	6	36.000
PAPELERÍA Y ÚTILES DE OFICINA	6.000	2	12.000
TOTAL			\$3.445.500

Fuente: Los Autores

4. Descripción y justificación de los viajes

LUGAR	JUSTIFICACIÓN	VALOR PASAJE UNITARIO (\$)	CANTIDAD	VALOR TOTAL (\$)
TALLER DE LÍNEA	PRESENTACIÓN DEL PROYECTO PARA RETROALIMENTACIÓN	10.000	12	120.000
SUSTENTACIÓN PÚBLICA DEL PROYECTO	PRERREQUISITO DE GRADO	10.000	4	40.000
ASESORÍAS	ASESORÍAS PARA PRESENTAR EL PROYECTO DE GRADO	10.000	6	60.000
TOTAL				\$220.000

Fuente: Los Autores

5. Salidas de Campo

ASPECTO	VALOR UNITARIO (\$)	CANTIDAD	VALOR TOTAL (\$)
APLICACIÓN DE ENCUESTAS PARA EL ANÁLISIS INICIAL	1.800	6	10.800
APLICACIÓN DE LA GUÍA EN LAS DOS EMPRESAS DE MUESTRA	1.800	8	14.400
TOTAL			\$25.200

Fuente: Los Autores