

PA143-04

**APLICACIÓN, VALIDACIÓN Y REFINAMIENTO DE LA “GUÍA PARA LA
INTEGRACIÓN DE MÉTODOS FORMALES DE INGENIERÍA DE
REQUERIMIENTOS EN PROCESOS DE DESARROLLO ÁGIL (IMFIR-PDA)”**

DIEGO CAMILO ROMERO VARÓN

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
2017**

PA143-04

APLICACIÓN, VALIDACIÓN Y REFINAMIENTO DE LA “GUÍA
PARA LA INTEGRACIÓN DE MÉTODOS FORMALES DE
INGENIERÍA DE REQUERIMIENTOS EN PROCESOS DE
DESARROLLO ÁGIL (IMFIR-PDA)”

Autor:

Diego Camilo Romero Varón

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO
DE LOS REQUISITOS PARA OPTAR AL TÍTULO DE
MAGÍSTER EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Director

Miguel Eduardo Torres Moreno

Comité de Evaluación del Trabajo de Grado

<Nombres y Apellidos Completos del Jurado >

<Nombres y Apellidos Completos del Jurado >

Página web del Trabajo de Grado

<http://pegasus.javeriana.edu.co/~PA1710-2-AVRGMFIR/>

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA

MAESTRÍA EN INGENIERIA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
Mayo, 2017

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

Rector Magnífico

Jorge Humberto Peláez, S.J.

Decano Facultad de Ingeniería

Ingeniero Jorge Luis Sánchez Téllez

Director Maestría en Ingeniería de Sistemas y Computación

Ingeniera Angela Carrillo Ramos

Director Departamento de Ingeniería de Sistemas

Ingeniero Efraín Ortiz Pabón

Artículo 23 de la Resolución No. 1 de junio de 1946

“La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia”

AGRADECIMIENTOS

Antes que todo a Dios por darme las herramientas para poder cumplir con este reto tanto profesional como personal, a Yelis, mi esposa, que ha tenido el amor, sabiduría y paciencia para apoyarme en este gran proyecto, a mis padres que en sus oraciones y palabras me acompañaban en momentos de incertidumbre, a Miguel Torres, mi admiración y quien con su conocimiento y habilidades me guio a través de sus consejos críticos y en la precisión de sus observaciones, a mi familia que con sus pensamientos me apoyaban desde la distancia, a Sol@ti por darme la oportunidad y el espacio de aplicar todo el conocimiento adquirido. Además, también quiero expresar mi admiración, por Alexandra Pomares con quien tuve una exigente, pero oportuna asesoría para fundamentar mi proyecto de grado y a Rafael González por sus buenas sugerencias que me aportó a mi proyecto de grado. Finalmente, al cuerpo docente y a la pontificia universidad Javeriana que sin duda ha sido la mejor elección para fortalecer mi ámbito profesional.

Contenido

INTRODUCCIÓN.....	XVI
1. DESCRIPCIÓN GENERAL	1
1.1. OPORTUNIDAD Y PROBLEMÁTICA.....	1
1.2. OBJETIVO GENERAL	1
1.3. OBJETIVOS ESPECÍFICOS	1
1.4. METODOLOGÍA	1
1.4.1. <i>Diseño Quasi-Experimental en la ingeniería de Software</i>	2
1.4.2. <i>Aplicación de la metodología DSRM para la implantación y refinamiento del a Guía</i> 3	3
1.5. FASES METODOLÓGICAS DEL PROYECTO	4
2. MARCO TEÓRICO	5
2.1. KERNEL DE SEMAT	5
2.2. GUÍA PARA LA INTEGRACIÓN DE MÉTODOS FORMALES DE INGENIERÍA DE REQUERIMIENTOS EN PROCESOS DE DESARROLLO ÁGIL BAJO EL MODELO DE SEMAT (IMFIR-PDA).	7
2.2.1. <i>Iteraciones de la GUÍA</i>	8
2.3. PLANEACIÓN Y PROCESO DEFINIDO PARA LA MEDICIÓN DE LOS PROYECTOS PILOTO, CONTROL Y EXPERIMENTO SEGÚN EL MODELO DE MEDICIÓN ISO / IEC 15939: 2007 8	8
2.3.1. <i>Aplicación del modelo la ISO / IEC 15939: 2007</i>	9
3. DEFINICIÓN DEL PROYECTO	9
3.1. CARACTERIZACIÓN UNIDAD ORGANIZATIVA	9
3.2. IDENTIFICAR LAS NECESIDADES DE INFORMACIÓN	10
3.3. PROCESO DE INGENIERÍA DE REQUERIMIENTOS EN COMPAÑÍA DESARROLLADORA DE SOFTWARE.	10
3.3.1. <i>Obtención de requerimientos</i>	11
3.3.2. <i>Análisis de requerimientos</i>	11
3.3.3. <i>Especificación de requerimientos</i>	11
3.3.4. <i>Validación de los requerimientos</i>	11
3.3.5. <i>Gestión de requerimientos</i>	12
3.4. PROCESO DE DESARROLLO DE SOFTWARE	12
3.5. SELECCIÓN DE MEDIDAS.....	13

3.6.	ENCUESTA BASADA EN MODELO DE ACEPTACIÓN DE TECNOLOGÍA (TAM)	14
3.7.	DEFINICIÓN DE TÉCNICAS ESTADÍSTICAS PARA COMPARACIÓN DE RESULTADOS 16	
3.7.1.	<i>Técnica Estadística</i>	16
4.	EJECUCIÓN DE PROYECTOS	17
4.1.	INDUCCIÓN AL EQUIPO DE TRABAJO A LA GUÍA IMFIR-PDA Y MODELO DEL KERNEL SEMAT	17
4.2.	ROLES INVOLUCRADOS EN LOS PROYECTOS DE EJECUCIÓN; ERROR! MARCADOR NO DEFINIDO.	
4.3.	METODOLOGÍA Y TIEMPO DE APROPIACIÓN DE LA GUÍA	19
4.4.	ELEMENTOS USADOS DE LA GUÍA POR EL EQUIPO DE TRABAJO	19
4.5.	SELECCIÓN DE PROYECTOS	20
4.5.1.	<i>Proyecto Piloto</i>	20
4.5.2.	<i>Proyecto Control</i>	22
4.5.3.	<i>Proyecto experimento</i>	24
4.6.	RESUMEN DE LA EJECUCIÓN DE PROYECTOS	25
4.7.	CONSTRUCCIÓN DE APLICACIÓN DE SOFTWARE COMO APOYO A LA RECOLECCIÓN DE DATOS DEL PROYECTO EXPERIMENTO	27
5.	RESULTADO DE LA EXPERIMENTACIÓN.....	28
5.1.	RESULTADO DE DATOS RECOLECTADOS EN LOS PROYECTOS SOBRE MEDIDAS SELECCIONADAS.	28
5.2.	ANÁLISIS E INTERPRETACIÓN DE DATOS	29
5.2.1.	<i>Análisis de Cantidad De Requerimientos Definidos</i>	30
5.2.2.	<i>Análisis de Horas Invertidas para Entendimiento de los Requerimientos</i>	31
5.2.3.	<i>Análisis de horas consumidas en Desarrollo Vs pruebas</i>	32
5.2.4.	<i>Análisis de pruebas Vs Defectos encontrados</i>	34
5.2.5.	<i>Análisis de cantidad de Defectos encontrados en fase de pruebas funcionales del usuario</i> 35	
5.2.6.	<i>Análisis de velocidad del Sprint y valor ganado</i>	35
5.2.7.	<i>Análisis de consumo de horas en todas las fases del proyecto</i>	36
5.2.8.	<i>Análisis Proyección de horas proyecto Experimento</i>	37
5.2.9.	<i>Análisis sobre la aceptación subjetiva de la Guía</i>	39
5.2.10.	<i>Análisis de comparación de resultados por puntos de casos de uso UCP</i>	41
6.	PROPUESTA DE REFINAMIENTO Y CONCLUSIONES	42
6.1.	PROPUESTA DE REFINAMIENTO	42
6.1.1.	<i>Aplicación de Casos de Uso 2.0</i>	43

6.2.	CONCLUSIONES	56
6.3.	TRABAJOS FUTUROS	59
	ANEXOS	60
	REFERENCIAS	62

INDICE DE ILUSTRACIONES

Ilustración 1 Cosas con las que siempre trabajamos - Tomada de [31]	6
Ilustración 2 Cosas que siempre hacemos - Tomada de [31].....	6
Ilustración 3 Proceso desarrollo de Software de la Compañía Objeto del proyecto (Diagrama elaborado para este proyecto).	13
Ilustración 4 Comparación de requerimientos y casos de uso identificados por proyecto.....	30
Ilustración 5 Comparación de horas consumidas entre entendimiento de los requerimientos Vs Tiempo consumido en especificación	32
Ilustración 6 Comportamiento incremental del consumo de horas por etapa según proyecto.	36
Ilustración 7 Comparación de tiempos consumidos en fases de ingeniería de requerimientos	37
Ilustración 8 Niveles de detalle de los productos de trabajo - Tomado de [76].....	45
Ilustración 9 Cosas por hacer - Tomado de [76].....	46
Ilustración 10 Ejemplo de un diagrama de caso de uso - Tomado de [76]	48
Ilustración 11 Ejemplo de porciones definidos a partir de un caso de uso primario - Tomado de [76].....	51

INDICE DE TABLAS

Tabla 1 Problemáticas generadas en las metodologías ágiles. Traducida y Tomada de [16].	xiv
Tabla 2 Resumen de medidas seleccionadas y sus características	14
Tabla 3 Preguntas seleccionadas para la encuesta de aceptación	16
Tabla 4 Tabal de anexos y soportes adicionales en ejecución de proyectos.	26
Tabla 5 Resultados consolidados de las medidas seleccionadas.....	29
Tabla 6 Tasa promedio de diseño de casos de uso por cada requerimiento identificado.....	31

Tabla 7 Relación promedio de horas invertidas en entendimiento del requerimiento por cada requerimiento según proyecto.....	31
Tabla 8 Relación promedio de horas de tiempo consumido en especificación por caso de uso	32
Tabla 9 Relación promedio de horas de desarrollo y pruebas por caso de uso.....	33
Tabla 10 Tasa promedio de cantidad de casos pruebas diseñados para un cubrir un caso de uso	34
Tabla 11 Tasa de defectos por cantidad de casos de prueba.....	35
Tabla 12 Relación de tiempo promedio consumido para solución y pruebas de un defecto reportado.	35
Tabla 13 Resultados de velocidad promedio y valor ganado.....	36
Tabla 14 Indicadores sobre proyección de ejecución proyecto Experimento	39
Tabla 15 Porcentaje de aceptación según percepción de “Facilidad” por rol y etapa de proyecto	40
Tabla 16 Porcentaje de aceptación según percepción de “Utilidad” por rol y etapa de proyecto.	40
Tabla 17 Comparación de proyectos experimento y control por factores de complejidad para cálculo de puntos por casos de uso.	41
Tabla 18 Aspectos de factores de complejidad relevantes por su diferencia entre los proyectos.	42
Tabla 19 Relación de actividades de la Guía en fase de Obtención de Requerimientos frente a actividades de Caso de Uso 2.0.....	47
Tabla 20 Nivel de detalle por cada producto de trabajo en relación a fase de Obtención de Requerimientos.	48
Tabla 21 Relación de actividades de la Guía en fase de Análisis de Requerimientos frente a actividades de Caso de Uso 2.0.....	49
Tabla 22 Nivel de detalle por cada producto de trabajo en relación a fase de Análisis de Requerimientos	50
Tabla 23 Relación de actividades de la Guía en fase de Especificación de Requerimientos frente a actividades de Caso de Uso 2.0.....	52

Tabla 24 Nivel de detalle por cada producto de trabajo en relación a fase de Especificación de Requerimientos. 53

Tabla 25 Relación de actividades de la Guía en fase de validación de Requerimientos frente a actividades de Caso de Uso 2.0..... 54

Tabla 26 Nivel de detalle por cada producto de trabajo en relación a fase de Validación de Requerimientos. 55

Tabla 27 Tabla de relación de estados porciones de casos de uso asociados a las fases del proceso de ingeniería de requerimientos..... 56

Tabla 28 Resultados de tasas o promedios más representativos del análisis comparativo entre proyecto control y proyecto experimento. 57

ABSTRACT

In the last decade, agile methodologies have been a necessary evolution point in software engineering; However, software development companies that are involved in agile processes are facing new challenges that combine technical and social aspects, particularly focused on requirements engineering processes. Based on that, many companies have chosen to strengthen the agile principles with formal engineering practices, resulting in hybrid software development processes. This has had positive results in competitiveness, as well as in mitigating the major cause of failures in Software Engineering projects. The present work takes as reference the methodological guide "IMFIR-PDA" for the integration of formal methods of the requirement engineering into software development processes, under the SEMAT model, with the objective of presenting the results of its implementation and validation, through experimentation in the Quasi-Experimental research design of Post-test Design type in which will be evaluated the accomplishment level in the execution of two projects in real software development environments in a systematic, disciplined, quantifiable and controlled way. As a result of this, a comparative analysis will be presented, in addition to a refinement proposal to enrich the properties of the methodological guide, based on the results obtained.

RESUMEN

En la última década, las metodologías ágiles han sido un punto de evolución necesario en la ingeniería de software; no obstante, las compañías de desarrollo que incursionan en procesos ágiles, se enfrentan a nuevos retos que combinan aspectos técnicos y sociales, particularmente concentrados en los procesos de Ingeniería de requerimientos. A partir de esto, muchas compañías han optado por fortalecer los principios ágiles con prácticas formales de ingeniería, originando procesos de desarrollo de software híbridos. Esto ha tenido resultados positivos en la competitividad, así como en la mitigación de la mayor causa de fracasos en los proyectos de ingeniería de Software. El presente trabajo, toma como referencia la guía metodológica "IMFIR-PDA" para la integración de métodos formales de ingeniería de requerimientos en procesos de desarrollo de software, bajo el modelo de SEMAT, con el objetivo de presentar los resultados de su implantación y validación, a través de la experimentación en el diseño de investigación *Quasi-Experimental* de tipo *Post-test Design* con el que se evaluará el nivel en que se logra de una forma sistemática, disciplinada, cuantificable y controlada; la ejecución de dos proyectos en entornos reales de desarrollo de software. Resultado de esto, se presentará un análisis comparativo, además de una propuesta de refinamiento para enriquecer las propiedades de la guía metodológica a partir de los resultados obtenidos.

RESUMEN EJECUTIVO

Hoy en día la industria del Software desata un proceso de transformación para ser cada vez más eficientes buscando reducir cualquier desperdicio de recursos en su ejecución. Muchas de estas compañías, apuestan a las metodologías ágiles, mientras otras rediseñan sus procesos, incluyendo algunas prácticas ágiles sin perder el conocimiento adquirido a través su experiencia. Este conocimiento, para algunas, ha contribuido en su crecimiento, para otras, les ha permitido mantenerse competitivas en un mundo globalizado y ofertante[1].

En cuanto a las metodologías ágiles, su aporte, ha permitido a la industria del software depurar lo que en principio, era el conjunto de causas comunes que hacen parte del fracaso en diferentes compañías desarrolladoras de software[2]. Estas metodologías surgieron como un intento de abarcar de manera más formal, el alto volumen de cambios en los requisitos¹ de software y un enfoque especial en la satisfacción de las expectativas del cliente[3]. De estas metodologías las más comunes son: *eXtreme Programming (XP)*[4], *Scrum*[5], *Dynamic Systems Development Method (DSDM)*[6], *Adaptive Software Development (ASD)*[7] y *The Crystal family* [8]. Sin embargo, en la industria del software, en estas metodologías se han evidenciado falencias cuando se trata de proyectos con un alto nivel de complejidad y esto se debe, a que en estos proyectos existen una gran variedad de stakeholders, una larga lista de requisitos y un entorno rápidamente cambiante[9]. Los problemas de las metodologías ágiles en estos contextos se originan por una documentación débil, la falta de una arquitectura sólida y la falta de gestión en lo riesgos identificados durante el desarrollo de software[10], con esto, se puede concluir que estas falencias se originan en el proceso de ingeniería de requerimientos dentro de sus procesos ágiles.

Si bien los beneficios de las prácticas ágiles en la Ingeniería de Requerimientos (RE – por sus siglas en inglés) han recibido atención, existe un estudio que revela que estas prácticas suponen varios retos, que deben ser abordados cuidadosamente por las organizaciones de desarrollo con el fin de lograr los resultados esperados[11]. A continuación, se listan algunos problemas críticos identificados en un estudio relacionado[11].

Problemas	Porcentaje (%)
Requerimientos incompletos	13.1
Cliente poco involucrado	12.4
Falta de recursos	10.6
Expectativas no realistas	9.9
Falta de gestión de soporte	9.3
Cambios en los requerimientos	8.7
Falta de planeación	8.1
Requerimientos Inútiles	7.5

¹ Para este proyecto y en la documentación escrita el concepto requisito tiene igual significado que la palabra requerimientos, por tal razón, dependiendo del sustento literario que soporte algún contexto dentro de este documento o de sus anexos, se hará uso de una o de la otra palabra.

Tabla 1 Problemáticas generadas en las metodologías ágiles. Traducida y Tomada de [16].

A raíz de lo anterior, desde la industria del Software, se ha dado lugar a propuestas como lo es *Extended XP process mode* en XP[10], híbridos con enfoque de gobernabilidad para entregas ágiles de soluciones empresariales conocido como *DAD*[12], extensiones a SCRUM como “*agile requirements refinery*”[9]. Otras soluciones cuyo objetivo principal es fortalecer el proceso de requerimientos en la industria del Software, como la propuesta desarrollada por Juan Darío Murcia Torres, en el año 2014 en su proyecto de grado de la Pontificia Universidad Javeriana, en la que presenta la Guía[13] nombrada para este proyecto por sus siglas “IMFIR-PDA”. Esta guía tiene como estructura principal el modelo teórico del kernel de SEMAT, el cual se caracteriza por tener un conjunto de elementos esenciales que son universales a todos los esfuerzos en la ingeniería de software; definido en un lenguaje sencillo en el que se describen métodos y prácticas, resguardando los principios ágiles. La esencia del kernel de SEMAT está compuesto por “*las cosas con las que siempre trabajamos*” y “*las cosas que siempre hacemos*”[14], la cual es una propuesta metodológica que a través de modelos con elementos visuales y dinámicos, fortalece las etapas de la ingeniería de requerimientos en términos de gestión y control.

Esta propuesta según Murcia, se puede implementar en cualquier metodología de desarrollo ágil y por sus características, posiblemente en metodologías híbridas, situación problemática también de interés para este proyecto.

Profundizando en esta investigación preliminar, a pesar que existe en la literatura una variedad de propuestas orientadas a mejorar los procesos de desarrollo de Software basadas en metodologías ágiles, existen investigaciones que exponen la preocupación de gran parte de las organizaciones al aplicar metodologías ágiles en proyectos de gran magnitud. Este análisis[9], evidencia que el 72,22% de las compañías proveedoras de Software, dirigen sus esfuerzos a mejorar sus procesos de desarrollo de Software, proponiendo adaptaciones de metodologías ágiles incorporando prácticas formales o tradicionales y en varios casos, incluyendo prácticas propias basadas en la experiencia corporativa generando un contexto de trabajo metodológico híbrido, que permite a los gerentes de productos de software, fortalecer la recolección de requerimientos para proyectos complejos[9].

Lo anterior problemática se presenta, debido a que adoptar una metodología ágil o extensiones de metodologías ágiles como las referenciadas anteriormente, no significa ser “el paso seguro” a la evolución en términos de procedimientos de desarrollo de Software y más aún, cuando de por medio no se cuenta con un entorno de trabajo apropiado, pues no solo basta en enfocarse en la manera en “el Como” se hacen las cosas, soportado en el conocimiento técnico del equipo y la infraestructura con que se cuente, sino el contexto social donde se desarrolla; pieza fundamental cuando se trata de ingeniería de Software[15]. Por esto, la interacción que existe entre el talento humano a servicio de las compañías desarrolladoras de software, es fundamental y hace parte del valor diferenciador, para definir el modelo más apropiado orientado a la mejora de procesos de desarrollo de Software en estas compañías[16], cualquiera que sea la estrategia en búsqueda de esa evolución, se enfrentan a nuevos retos que para muchas compañías les ha representado fracasos significativos, al no considerar la dependencia que tienen estos modelos o metodologías con el comportamiento humano en un entorno corporativo. El simple hecho que un modelo aplique para alguna compañía no obedece que su resultado sea exactamente

igual al aplicar el mismo modelo en otra compañía, por lo que, en este ejercicio, juega un papel importante los estudios empíricos para la evaluación de procesos y actividades basadas en el ser humano. Es por esto, que para este proyecto, se hace uso de la experimentación a través de la investigación empírica, considerando que, la manera fundamental de evaluar nuevas propuestas es comparándolas con las existentes, teniendo en cuenta que la única evaluación real de una propuesta de mejora de proceso, es que las personas hagan uso de esta[15].

A partir de lo anterior, como resultado de este proyecto, se presenta el análisis comparativo de dos proyectos reales para una compañía desarrolladora de Software en Colombia, bajo el modelo alterno de diseño de investigación Quasi – Experimental[17], donde uno de los proyectos hace las veces de “proyecto Control” en el que no existe ninguna alteración y será sujeto de observación; y otro proyecto de desarrollo de software, donde se implanta la Guía metodológica propuesta por Murcia[13], denominado “proyecto Experimento”. Para esto se realiza un proceso riguroso de selección, recolección y administración de datos sobre las medidas previamente seleccionadas, posterior a esto, se presenta la propuesta de refinamiento para la guía.

INTRODUCCIÓN

El presente trabajo está enmarcado dentro del grupo de investigación ISTAR de la Pontificia Universidad Javeriana y corresponde, a la línea de profundización en Ingeniería de Software en su rama Ingeniería de Requerimientos. El desarrollo de este proyecto, plantea el aporte a la base de conocimiento de la ingeniería de software, como una solución a las metodologías ágiles o híbridas en su proceso de Ingeniería de Requerimientos; cuando se trata de proyectos complejos. Esta propuesta tiene como referencia la Guía metodológica “IMFIR-PDA”[13], la cual está construida bajo el modelo de aplicación del Kernel SEMAT [19], con esto, se busca demostrar el uso potencial que tiene SEMAT a través de la Guía, como un ejemplo de uso práctico, en un contexto real de trabajo para desarrollo de Software. Además de potencializar el uso SEMAT como propuestas de trabajo futuro, para diseñar soluciones de las diferentes problemáticas de la ingeniería de Software, por ejemplo, en la fase de pruebas o construcción de Software.

Este proyecto se propone como una solución a la emergente problemática de metodologías ágiles a nivel global. La práctica de este proyecto, promueve el uso de la guía para empresas pequeñas, medianas o grandes, que presentan problemas en la ingeniería de requerimientos. Con esto se pretende disminuir el porcentaje de fracaso en los desarrollos de software, fortaleciendo principalmente el crecimiento de la industria del software en Colombia, al mismo tiempo, generando confianza de inversionistas extranjeros, viendo a Colombia como una oportunidad competente en fábricas de desarrollo de Software.

El presente documento, describe el proceso que se llevó a cabo para la implantación y validación de la Guía y el análisis comparativo de los resultados obtenidos, durante la ejecución del proyecto Control y Proyecto Experimento, resultado de este análisis, se obtiene la propuesta de refinamiento y las conclusiones, que responden a la pregunta de investigación que dio origen a este proyecto de grado.

Este documento está compuesto de seis secciones definidas de la siguiente manera: la primera sección está orientada en exponer la problemática u oportunidad de mejora, los objetivos y en argumentar el soporte metodológico que sustenta este proyecto, en la sección dos, se presenta el marco teórico para contextualizar al lector sobre los ejes transversales del proyecto descritos en este documento, la tercera sección, describe la consideraciones experimentales para definir a los proyectos, la cuarta sección presenta un resumen de la ejecución del proyecto Quasi – Experimental, la quinta sección presenta el consolidado de datos recolectados y análisis e interpretación de datos, la sexta sección las conclusiones resultado de la implantación de la Guía, y las propuesta de refinamiento, proyectos futuros, anexos y referencias.

1. DESCRIPCIÓN GENERAL

Oportunidad y problemática

La problemática que busca resolver este proyecto, es la de evaluar la Guía *IMFIR-PDA*[13], a partir de un experimento riguroso en el que se estudia los resultados, de combinar aspectos sociales y técnicos en un contexto corporativo de la industria del Software, en el que se determine, si su implantación favorece el proceso de ingeniería de Requerimientos bajo la práctica de una metodológica de desarrollo híbrida.

Objetivo general

Enriquecer los procesos de ingeniería de requerimientos para favorecer la aplicación de metodologías ágiles en proyectos complejos, aplicando de manera experimental la guía (IMFIR-PDA) en una empresa colombiana de tecnología.

**(IMFIR-PDA): Guía para la integración de métodos formales de ingeniería de requerimientos en procesos de desarrollo ágil bajo el modelo de SEMAT.*

Objetivos específicos

- Implantar la guía (IMFIR-PDA) en una empresa colombiana de tecnología en proyectos de desarrollo de software con estimaciones entre 40 y 600 horas/hombre.
- Validar la guía (IMFIR-PDA) mediante el diseño cuasi experimental comparando dos proyectos de desarrollo de software (Grupo control y Grupo experimento) que parten de condiciones similares.
- Construir una aplicación de Software que sirva como apoyo a la recolección de datos generados de las técnicas de estadística que garantice una apropiada administración de los resultados obtenidos del desarrollo del objetivo 2 de este proyecto.
- Enriquecer la guía (IMFIR-PDA) a partir de los resultados de la investigación cuasi-experimental y los análisis efectuados sobre las experiencias obtenidas por el equipo de trabajo que participó en los 2 proyectos.

Metodología

Para el desarrollo de los objetivos de este proyecto, se fundamenta en dos metodologías diferentes pero complementarias “Quasi – Experimental y DSRM”. La primera de estas metodologías, se basa en el marco de diseño de investigación Quasi-Experimental[17] propuesta por *Cook & Campbell* en 1979, que también hace parte de la referencia literaria del libro “*Experimentation in Software Engineering*”[15], siendo este, el marco de referencia para los lineamientos prácticos y la manera en que se plantea evaluar los métodos o técnicas experimentales en la Ingeniería de Software. La segunda metodología, se basa en el framework que presenta *Ken Peffers* en el 2008 nombrado “DSRM”[18] que por sus siglas en ingles traduce “*Metodología de investigación para la ciencia del diseño*”, la cual se fundamenta en el paradigma de la ciencia basada en el diseño de *Hevner*[19].

A continuación, se describe brevemente el contexto de aplicación de cada una de estas metodologías de cara a la aplicación de este proyecto de grado.

1.1.1. Diseño Quasi-Experimental en la ingeniería de Software

Partiendo del concepto de la experimentación en la ingeniería de software, se ha evidenciado autores que tienen similares definiciones, entre estas, se resalta la del Profesor *Dag IK Sjøberg* en el año 2005 donde define[20] “*Un experimento controlado en ingeniería de software es un ensayo aleatorio, en el cual individuos o equipos (las unidades de estudio) realizan una o más tareas de ingeniería de software para comparar diferentes poblaciones, procesos, métodos, técnicas, lenguajes o herramientas*”.

Entrando en profundidad con este concepto, *Shadish en el 2001* define que para la ingeniería de Software el diseño Quasi-experimental[17] es una rama de los experimentos[21], siendo este diferente a los demás tipos de experimentos, partiendo del hecho de que carecen de asignación aleatoria de unidades de estudio a grupos experimentales (asignación al azar) [16], en los Quasi Experimentos hay exactamente un grupo experimental seleccionado para cada condición de tratamiento donde las unidades experimentales están expuestas a múltiples tratamientos, posiblemente en órdenes diferentes[22].

Esta rama de investigación experimental[23], es la que se pretende aplicar en este proyecto como se ha referenciado anteriormente, razón por el cual encaja en este contexto, se debe a que aplica cuando los proyectos no pueden ser aleatorios sin contar con criterios similares para una comparación objetiva. En algunos casos, puede ser imposible asignar al azar este tipo de grupos (Experimento y de Control); explicación que sustenta esta técnica de diseño de investigación, teniendo en cuenta que la asignación aleatoria significa que los grupos pueden no ser equivalentes y por consiguiente, presenta pocas garantías en la validez de la investigación que se pretende[16], [23]. Los Quasi - experimentos tienen ventajas significativas en los entornos de investigación al contar con mayores niveles de autenticidad al no ser artificiales[24]. Por tal motivo, se considera de las técnicas de investigación más utilizadas para analizar el comportamiento de las variables dependientes en un proyecto de investigación[23].

Este tipo de experimentos, son ejecutados en entornos naturales mediante la manipulación del hábitat social y/o mediante la introducción de cambios en el medio socio-técnico, no obstante, los cambios en las variables independientes también pueden ocurrir de forma natural en lugar de la manipulación [25]. Para este caso puntual, el experimento se trata de un cambio en la metodología de desarrollo de Software, para fortalecer la ejecución de las actividades relacionadas al proceso de requerimientos, implantando la guía *IMFIR-PDA* en uno de los grupos, el cual tomará el nombre de “Grupo Experimento” o “Proyecto Experimento”, obteniendo los beneficios al no ser totalmente intrusivos ni muy costosos[17], lo que favorece y es parte esencial para la ejecución real en el planteamiento deseado para este proyecto, al aplicar la guía metodológica en equipos de desarrollo de software en proyectos reales.

Las consideraciones para poder comparar los resultados de un proyecto experimento (Proyecto intervenido con la Guía) y el proyecto Control (proyecto de observación sin intervención de la Guía), están orientadas a la homogeneidad entre estos dos grupos, y para esto, se integran equipos de equivalente experiencia, antigüedad y categorizando a los clientes que cuenten con variables y contextos parecidos de trabajo, con esto, aunque la idea es reducir las diferencias que se puedan generar durante la ejecución del Quasi – Experimento, también se quiere

aprovechar la ejecución de la guía en un entorno natural de trabajo sin forzar a los grupos a perder su identidad.

- **Post-test grupo Experimento y grupo Control**

Los diseños cuasi-experimentales pueden ser fortalecidos mediante la adición de elementos de diseño cuidadosamente elegidos con el fin de reducir el número y la plausibilidad de las amenazas en la validez de la hipótesis[22]. En este caso se hace uso de la técnica Post-Test, pues se conoce como uno de los más comunes y apropiados para este tipo de proyectos[26]. Durante la ejecución de cada proyecto se plantea tomar los datos de las medidas según el procedimiento definido en el punto [3.5](#), sin embargo, solo serán procesadas, consolidadas, analizadas al finalizar cada proyecto (control y experimento), con esto se pretende evitar generar modificaciones durante la ejecución, buscando generar algún resultado conveniente sesgando la objetividad del experimento.

1.1.2. Aplicación de la metodología DSRM para la implantación y refinamiento del a Guía

El segundo marco metodológico para este proyecto, se basa en la teoría de la ciencia basado en el diseño, que trata de la creación y/o transformación de artefactos artificiales, ya sean materiales o inmateriales, incluyendo sus modalidades de interacción con los ambientes socio técnicos y culturales en los que se inserta[27], tales artefactos pueden incluir constructos, modelos, métodos e instancias[28]. También podrían incluir innovaciones sociales o nuevas propiedades de recursos técnicos, sociales y / o informativos[29]; En resumen, esta definición incluye cualquier objeto diseñado para una solución que se integra en un sistema. A continuación, se resumen los pasos de esta metodología aplicados al experimento:

Actividad 1 - Identificación del problema y motivación: Dado que la definición del problema en relación en este proyecto, se puede concluir como la oportunidad de implantar la guía en procesos de desarrollos de software reales, la manera en que se da alcance a esta actividad es mediante la ejecución de los proyectos piloto, control y experimento², además de una iteración de refinamiento de la Guía a partir de las recomendaciones de mejora presentadas este proyecto.

Actividad 2 - Definir los objetivos de una solución: Los objetivos pueden ser cuantitativos, por ejemplo, pueden ser en los términos cuantificables en que una solución sería deseable frente a los resultados comparativos con otros proyectos similares ejecutados previamente, o cualitativos, por ejemplo, una descripción de cómo se espera que un nuevo artefacto soporte soluciones a problemas no tratados hasta ahora[18]. En relación a este proyecto, esta actividad se satisface con la aplicación de la guía y los resultados que de este experimento resulten.

Actividad 3 - Diseño y desarrollo: Se crea un artefacto o se presentan nuevas propiedades de los recursos técnicos, metodológicos y / o informativos[29]". En este caso se proponen dos escenarios para cubrir esta actividad, el primero es el diseño y desarrollo de la aplicación de la Guía como un artefacto teórico induciéndola en un contexto práctico. El segundo escenario

² La definición de los proyectos (Piloto, Control y Experimento) y sus características se mencionan en el punto [4.1](#)

parte de una esquema de propuesta de refinamiento a partir de mejoras identificadas para la Guía[13] o consideraciones a tener en cuenta, basado en que, conceptualmente, un artefacto de investigación basado en el diseño puede ser cualquier objeto en el que genere una contribución a una investigación.

Actividad 4 – Demostración: Demostrar el uso del artefacto para resolver uno o más casos del problema. Para esta actividad solo adopta para la ejecución experimental implantando la Guía en su estado actual.

Actividad 5 – Evaluación: Esta actividad implica dos tipos evaluaciones diferentes, la primera se basa en la evaluación a partir de la comparación de los resultados del proyecto experimento con los resultados reales del proyecto control, basado en las métricas seleccionadas a partir de prácticas del modelo **ISO / IEC 15939: 2007** (se detalla en el punto [2.3.1](#) de este documento). La segunda evaluación se realiza a la propuesta de refinamiento de la Guía a partir de una comparación subjetiva, partiendo del juicio de los integrantes sustentada en la experiencia obtenida de la ejecución de los proyectos (Piloto y Experimento), mediante una encuesta de aceptación [[3.6](#)], con esto, conocer en un contexto de utilidad y usabilidad si estas mejoras son consideradas favorables para la guía metodológica.

Actividad 6. Comunicación. En cuanto a la comunicación se pretende desde la publicación de este proyecto de grado en un sitio público, como de la propuesta de un artículo indexado en una revista nacional garantizando el rigor en su contenido, esto alimenta la base de conocimiento en la ingeniería de software en su área de ingeniería de requerimientos, permitiendo un aporte para trabajos futuros de investigadores y demás profesionales en el ejercicio.

En conclusión, *Peffer*s señala que la aplicación de la metodología puede darse en casi cualquier actividad[18], teniendo en cuenta la existencia de un artefacto, que todavía no ha sido formalmente evaluado en proyectos reales, como una solución para el dominio de problema explícito en el que se utilizará. Otro escenario es el resultado de otro dominio de la investigación como lo es en este caso la propuesta refinamiento de la Guía[13]. Es por esta razón, se explican dos contextos diferentes (uno de los escenarios es para la implementación de la guía en su estado actual y un segundo escenario para la propuesta de refinamiento), donde según el enfoque expuesto, se aplican algunas actividades (no todas) de la metodología de *Peffer*s.

Fases Metodológicas del Proyecto

Partiendo de las metodologías que estructuran a este proyecto, se presentan las siguientes fases a ejecutar en cumplimiento de los objetivos 1,2 y 4:

- Fase 1: Diseño de investigación Quasi – Experimental.
- Fase 2: Aplicación preliminar (ejecución proyecto Piloto).
- Fase 3: Ejecución del diseño y aplicación formal de la guía.
- Fase 4: Análisis y comparación de resultados.
- Fase 5: Refinamiento de la guía IMFIR-PDA.

Para el objetivo 3 se realiza una aplicación de software que permita almacenar las medidas seleccionadas para la comparación de los proyectos.

2. MARCO TEÓRICO

El presente trabajo de grado tiene como objetivo enriquecer los procesos de ingeniería de requerimientos en metodologías ágiles o híbridas cuando se trata de proyectos complejos, para esto, la opción que se propone para resolver esta problemática, es a partir de la aplicación de la guía (IMFIR-PDA)[13] en proyectos de desarrollo de software reales, dentro de una compañía que tiene como objetivo desarrollar software comercial en Colombia. Es por esto, antes de exponer la ejecución y los resultados de esta aplicación, es primordial generar una lectura interpretativa de este documento, lo que hace preciso contextualizar algunos parámetros que sirvan de ejes conceptuales sobre los que se sustenta este proyecto de grado. Para empezar, se explicará las principales características de SEMAT[14], posterior a esto, exponer como la Guía[13] hace uso de esta base teórica “SEMAT” para generar un resultado práctico, por último, los argumentos requeridos y seleccionados para comparar el experimento Quasi - Experimental haciendo uso a la recomendación del modelo de medición ISO / IEC 15939: 2007.

Kernel de SEMAT

SEMAT nació por la necesidad de cambiar fundamentalmente la forma en que la gente trabajaba con la ingeniería de software. Esta iniciativa fue tomada por Ivar Jacobson, Pan-Wei Ng, Paul E. McMahon, Ian Spence y Svante Lidman[14], quienes crearon lo que muchos autores llaman “La esencia de la Ingeniería de Software: El núcleo de SEMAT”[30].

SEMAT se enfoca fundamentalmente en dos objetivos principales, y que son independientes el uno del otro:

1. Encontrar un Kernel de elementos ampliamente aceptados.
2. Definición de una base teórica sólida.

Para definir el Kernel, lo principal fue identificar un terreno común que de alguna u otra forma se manifestaba como un núcleo de elementos esenciales que son universales a todos los esfuerzos de desarrollo de software y un lenguaje sencillo para describir métodos y prácticas. Luego de definidos los elementos esenciales, estos fueron clasificados en 2 grandes grupos, “las cosas con las que siempre trabajamos” cuando se desarrolla software (Ver ilustración 1), y el grupo dentro de “las cosas que siempre hacemos” en dichos proyectos (Ver ilustración 2). Estos dos grupos son lo que se conoce en SEMAT como la esencia del Kernel el cual provee:

1. Un marco de pensamiento para que el equipo de desarrollo razone sobre el progreso que están logrando y los beneficios de sus esfuerzos.
2. Un terreno común para la discusión, mejoramiento, comparación e intercambio de métodos y prácticas de ingeniería de software.
3. Un marco para que los equipos ensamblen mejores prácticas de Ingeniería de software y mejoren continuamente el trabajo.
4. Un método de medición para evaluar la calidad del software y la eficacia de los métodos producidos.

- Es una forma de ayudarle a los equipos a comprender dónde están, qué deberían hacer luego y dónde necesitan mejorar.

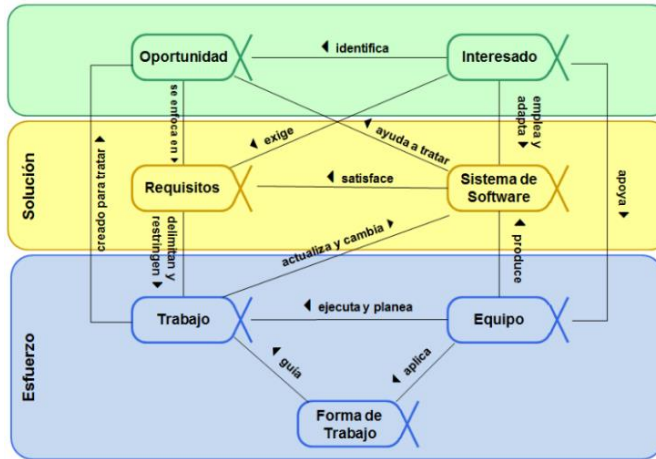


Ilustración 1 Cosas con las que siempre trabajamos - Tomada de [31]

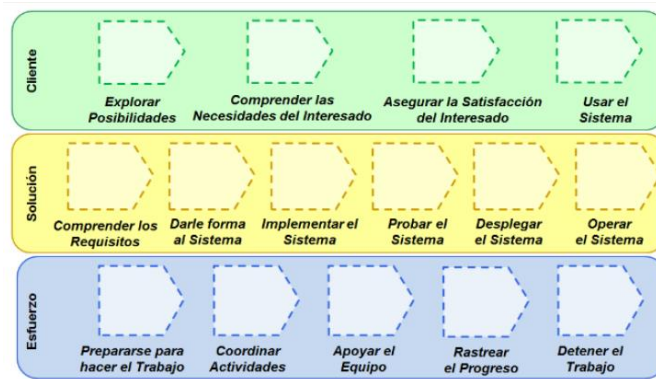


Ilustración 2 Cosas que siempre hacemos - Tomada de [31]

Los Alfas

Son elementos fundamentales del Kernel y esenciales a la Ingeniería de Software. en la guía, Murcia[13] relaciona a las alfas como conjuntos de estados, en vez de elementos de trabajo (como documentación o herramientas de software), ya que así representan el progreso y salud del proyecto, y a la Ingeniería de Software en general y no a metodologías en específico.

Han sido identificados 7 Alfas, los cuales, como se mencionó anteriormente, corresponden a “las cosas con las que siempre trabajamos”. Cada Alfa tiene estado una serie de Estados, quienes de igual manera se encuentran asociados una lista de chequeo que especifica los criterios necesarios para alcanzar todo el estado.

Guía para la integración de métodos formales de ingeniería de requerimientos en procesos de desarrollo ágil bajo el modelo de SEMAT (IMFIR-PDA).

La guía (**IMFIR-PDA**), a partir de su contenido publicado, se hace referencia de manera resumida a los siguientes puntos relevantes con el objetivo de contextualizar al lector, es importante aclarar, de contar con el interés de conocer a fondo lo que la guía ofrece, remitirse a versión completa en su publicación web disponible³. A continuación se realiza una breve reseña sobre la guía[13]:

La Guía cuenta con método denominado de Way-Of[31], la forma en que son propuestos los aspectos del Way-Of en los que se dividen de la siguiente manera:

- **Way-Of-Thinking:** En el caso de esta guía, se pretende que la forma de pensar sea en términos de lo ágil y que los procesos tanto del proyecto, como de la organización, utilicen estos principios.
- **Way-Of-Working:** La forma de trabajar va a ser a través de SEMAT y la implementación de su Kernel. Esto implica que el desarrollo de los procesos va a ser gestionado por medio de los Alfas.
- **Way-Of-Modeling:** El modelado que ofrece la guía se basa en RML (*Requirements Modeling Language*) presentados en el libro de *Joy Beatty Anthony Chen “Visual Models for Software Requirements”* [32] De estos modelos a trabajar con la guía son:
 - Modelo de Objetivos del Negocio
 - Organigrama
 - Modelo de Flujo de Procesos
 - Mapa del Ecosistema
 - Diagrama de Datos del Negocio
- **Way-Of-Supporting:** En la Guía se propone el uso de SEMAT como principal herramienta en la forma de trabajar al interior del proyecto. En este aspecto se encuentra incluido el aseguramiento de elementos como la trazabilidad y la gestión del proyecto.

³ <http://pegasus.javeriana.edu.co/~CIS1410IS07/>

2.1.1. Iteraciones de la GUÍA

Una vez que se tiene el claro los conceptos metodológicos con que opera la guía, se define pasos iterativos con los que se pretende facilitar la operatividad de la aplicación de la guía, a continuación, se presenta los pasos iterativos[13]:

- **Determinar el estado actual del proyecto:** Verificar en las Cartas Alfa de Estado los Estados en los que se encuentra actualmente.
- **Determinar los Estados objetivo:** Luego de haber establecido los Estados en los que se encuentra actualmente, verificar los Estados que siguen en las Cartas Alfa de Estado. De la misma manera se seleccionan los Ítem de Requerimiento, los requerimientos que se desea gestionar en la iteración y los estados de este que se intentará cumplir.
- **Determinar cómo se van a lograr dichos Estado objetivo:** En este punto se verifica la lista de chequeo de los Estados objetivo, y el/los estados de los Ítem de Requerimiento que hayan determinado anteriormente. Con el equipo se establece las actividades que se deberán llevar a cabo para poder cumplir completamente con dichos Estados.
- **Trabajar para cumplir con los Estados objetivo:** Se considera el trabajo bajo practicas agiles, mantener la comunicación con el equipo, ya que esto ayudará a superar obstáculos más rápidamente.
- **Verificar el trabajo se esté haciendo:** Mientras se realizan todas las actividades, es conveniente que se haga seguimiento a los miembros del equipo, a fin de determinar si se está realizando el trabajo, y si se está haciendo de acuerdo a las reglas ya establecidas y a los “Way-Of”.
- **Retroalimentación del trabajo realizado en la iteración:** Una vez se ha culminado la iteración, se espera que todo lo planeado en un comienzo se haya cumplido a cabalidad, y con la calidad esperada. Allí se establecen puntos de fallo y fortalezas. Aprendan de lo sucedido, y creen una nueva forma de trabajar para la siguiente iteración, que incluya las posibles mejoras. De igual manera, establezca si los Estados objetivo que se había estipulado cumplir durante la iteración, efectivamente se lograron.

A partir de lo anterior, la Guía integra a SEMAT para fortalecer al proceso de nuevos requerimientos sin perder su condición de proceso de metodología de desarrollo ágil o híbrida, aplicando las alfas consideradas por el autor que precisan en cada etapa del proceso de requerimientos definidos por la Guía⁴.

Planeación y proceso definido para la medición de los proyectos piloto, control y experimento según el modelo de medición ISO / IEC 15939: 2007

El propósito del proceso de medición es recoger, analizar y reportar los datos relativos a los productos desarrollados y procesos implementados dentro de una organización, estas

⁴ Para mejor detalle de las actividades por cada estado remitirse a la Guía, documento anexo con el nombre: Guía_(IMFIR-PDA).pdf

mediciones vienen siendo la base de la ingeniería de software empírica [33]; los investigadores que se desempeñan en este campo, aducen que sólo se puede estar seguro de la contribución de los métodos y técnicas de ingeniería de software, cuando se aportan pruebas concretas con que se pueda evidenciar los beneficios que sus autores ofrecen[34]. Esto sucede, cuando al objeto de medición se aplica una métrica a partir de valor numérico (cuantitativo) o mediante un atributo (cualitativo), al comparar estos valores, se aumenta la precisión de generar conclusiones objetivas, sobre la calidad, eficiencia o eficacia de los procesos de software, herramientas, métodos u otros artefactos[35].

2.1.2. Aplicación del modelo la ISO / IEC 15939: 2007

Las organizaciones maduras de desarrollo de software a nivel mundial utilizan ampliamente el modelo de información de la norma ISO / IEC 15939 como medio para identificar las necesidades de información e implementar sistemas de medición[36]. Partiendo de lo anterior, para este proyecto, el modelo de definición de medidas se basa en la ISO / IEC 15939: 2007[37], el cual define un proceso aplicable a las disciplinas de ingeniería de software y gestión de sistemas. El proceso define las actividades de medición requerido para: planear, seleccionar[38], aplicar, administrar adecuadamente, las medidas y la información obtenida sobre estas, y cómo determinar si los resultados del análisis son válidos. El proceso de medición es flexible, adaptable, lo cual permite apropiarse el modelo a la necesidad de este proyecto de grado. Para conocer a detalle el contenido que se llevó a cabo para el proceso de medición basado en la ISO / IEC 15939 remitirse al *anexo 12 “Definición Variables a Medir.docx”*.

En cuanto a la consecución del diseño Quasi – Experimental, las métricas seleccionadas deben considerarse transversales a los dos proyectos y ser orientadas a los resultados[39]. Estas medidas deben estar relacionadas en dimensiones de Tiempo, Calidad y Riesgo[40], puesto que el objeto de medición en la aplicación de la Guía, es efectivamente comprobar si su implantación proporciona beneficios en la ingeniería de requerimientos, bajo la ejecución de metodologías ágiles o híbridas; teniendo en cuenta, el talento humano a servicio de una compañía de la industria de desarrollo de Software en Colombia. Por esta razón, para este proyecto se fundamenta en Integrar los procedimientos de medición con el proceso de software que se está midiendo [41].

3. DEFINICIÓN DEL PROYECTO

En este segmento, se describen las características identificadas para seleccionar las medidas con que se compararán los proyectos (Experimento y Control), producto de la ejecución del proceso de medición basado en el modelo ISO 15939: 2007.

3.1. Caracterización unidad organizativa

El presente proyecto, se realiza en una Compañía⁵ colombiana desarrolladora de Software especializada en soluciones integrales, para la gestión y automatización de los procesos de cobranza y otros procesos de negocio. Es reconocida en el mercado nacional e internacional, por su producto comercial y los excelentes servicios de consultoría en cobranza, apoyando a diferentes procesos dentro de compañías financieras catalogadas como ‘AAA’⁶.

Actualmente esta compañía, cuenta con certificaciones NTC/ISO 9001: 2008[43] y ISO/IEC 27001:2013[44, p. 2013] y se encuentra en proceso de certificación CMMI-SVC Level 3[45].

3.2. Identificar las necesidades de información

La compañía donde se ejecutará el proyecto control como el proyecto experimento, se encuentra en una transición organizativa, de pasar de una filosofía de desarrollo tradicional a una de desarrollo ágil, sin embargo, por ser proveedor de diferentes compañías en diferentes sectores de la industria colombiana, esta compañía propone modelos híbridos de desarrollo de software, para adaptarse y garantizar el equilibrio que se requiere en la industria, la cual se encuentra en un cambio social tecnológico[46], [47]. Es claro, que la filosofía ágil tiende al crecimiento y por lo tanto las metodologías ágiles en Colombia, cada vez se hacen más fuertes[48], no obstante, esta industria se encuentra en un estado en que todavía los diferentes clientes, se aferran a los largos contratos y a la documentación extensa pero “completa” frente a lo requerido, haciendo el proceso de desarrollo de software robusto, pero al mismo tiempo, con un alto nivel de confiabilidad para el cliente[49]. Esta situación se presenta, a partir de las mismas transformaciones que ha tenido la evolución del software, que de cierta manera, esta transición ha hecho que los clientes sean escépticos a estos principios ágiles, donde las garantías, se convierten en la confianza que exista entre el cliente y el proveedor de software[10].

A continuación, se describe el proceso de desarrollo de Software de la compañía objeto de este experimento (Ver *Ilustración 3*) y como aplica la ingeniería de requerimientos dentro de este proceso.

3.3. Proceso de ingeniería de Requerimientos en compañía desarrolladora de Software.

El proceso de ingeniería de requerimientos en la compañía desarrolladora de Software, se fundamenta en un proceso tradicional de ingeniería de requerimientos[35] en cascada, que incluye prácticas ágiles en etapas de construcción y pruebas. A continuación, se describe el

⁵ Por compromisos de confidencialidad de la compañía con sus clientes se mantiene anónimo el nombre de la compañía desarrolladora de Software como el de sus clientes, debido a que los proyectos seleccionados son proyectos reales que dejan entrever situaciones presentadas que deben permanecer bajo un nivel de confidencialidad.

⁶ AAA: La más alta calificación de una compañía relacionadas al sector financiero, fiable y estable en cuanto a inversiones estables según Standard & Poor's Financial Services LLC (S&P) [42].

proceso de ingeniería de requerimientos para la compañía en cuestión:

3.4. Obtención de requerimientos

En esta fase se realizan actividades tales como levantamiento de información con los usuarios, a partir de entrevistas y cuestionarios como técnicas de obtención de requerimientos, esta actividad la realiza un ingeniero especialista en Análisis funcional y de negocios.

3.5. Análisis de requerimientos

Esta actividad también es realizada por el ingeniero analista funcional quien hizo parte del ejercicio de obtención de los requerimientos, el ejercicio de análisis, en ocasiones, involucra la validación y participación activa de los ingenieros más expertos de la compañía, en temas de desarrollo de software, orientadas al modelo de negocio de software de paquete o comercial como se conoce en la industria.

3.6. Especificación de requerimientos

Esta fase del proceso se realiza un ejercicio documental donde la especificación llega a un nivel únicamente de definición funcional mas no técnica, y para esto, se hace uso de diagramas dinámicos de UML como los casos de uso y diagramas de actividades, esta documentación está orientada 100% al cliente o usuarios funcionales, los formatos que se utilizan para esto, son el de especificación de los requerimientos⁷, que describe en un contexto general la funcionalidad a partir de los requerimientos funcionales y no funcionales identificados. Este documento viene acompañado de una descripción por caso de uso⁸ que es un segundo documento que reúne los requerimientos agrupados por casos de uso, en el que se describen los flujos (alternos, principal, subflujos), cuerpo del caso de uso, el diagrama de casos de uso, actores, extensiones e inclusiones, definición de casos de éxito o casos fallidos, sus correspondientes precondiciones y resultados esperados.

3.7. Validación de los requerimientos

La validación de los requerimientos se hace inicialmente por parte del entendimiento y aprobación del gerente de proyectos, quien debe valorar y tener la claridad de poder interpretar el documento funcional. En segunda instancia, se procede a la validación y aprobación del usuario solicitante del desarrollo, esto puede ser un cliente o un usuario interno de la compañía desarrolladora de Software, esto último, como practica de mejora continua entre las actividades de desarrollo e investigación de la compañía. En algunos casos el equipo de desarrollo hace parte del proceso de validación, para garantizar que el desarrollo este orientado a una solución reusable, que permita fortalecer el portafolio comercial.

⁷ Ver documento anexo 18: FO-SF-070 EspecificacionFuncional

⁸ Ver documento anexo 19: FO-SF-071 CasosDeUso

3.8. Gestión de requerimientos

Durante todo el proceso de ingeniería de requerimientos se establece controles a los requerimientos, a partir del versionamiento, control de cambios y la administración que surja de la política de gestión de la configuración de la compañía. La gestión de los requerimientos está bajo la responsabilidad del gerente de proyecto.

3.9. Proceso de Desarrollo de Software

Una vez se culmina una primera versión documental previamente aprobada por el cliente, inicia una fase donde el proceso de desarrollo de software se torna a su versión más ágil, haciendo uso de buenas prácticas de SCRUM [5] y XP[4]. Esta nueva instancia se activa en el momento que la especificación funcional, se asigna para la estimación inicial y socialización del grupo de requerimientos previamente identificados, esto se hace durante el Sprint Planning [50] del grupo de desarrollo. El gerente de proyecto y el analista funcional, quienes gestionaron el requerimiento desde su fase de obtención de requerimientos, se vuelven los interlocutores del proyecto, asumiendo el rol de Product Owner [50], el gerente, a este punto, si así lo considera, puede hacer partícipe al usuario quien realmente solicitó el desarrollo, sin embargo, la representación del requerimiento sigue siendo el gerente de proyectos.

Durante el Sprint Planning se toma los requerimientos definidos, se desglosan en tareas a desarrollar y se realiza la actividad de planeación y estimación basada en la técnica de Planning Poker[51]. De esta manera se concluye con las actividades que se entregarán, agregando estas tareas con su respectiva estimación en el Sprint Backlog[50], a partir de esto, se inicia las iteraciones correspondientes; consumiendo Sprints[50] cuantos sean necesarios según capacidad y velocidad del equipo de desarrollo.

Al finalizar el ciclo de cada Sprint, en su fase de Sprint Review[50] se valida cada entregable con el product Owner / Cliente y el equipo de testing, quienes recibirán, la funcionalidad para validar y garantizar el nivel de calidad considerado para la entrega a los usuarios finales, además de esto, se hace un ejercicio de retrospectiva referente a los entregables y lecciones aprendidas durante el Sprint culminado. Todos los issues o defectos encontrados durante las pruebas de testing, se plantean como actividades nuevas a resolver en el próximo Sprint Planning, de esta misma manera, aplica para los casos reportados en una instancia de pruebas funcionales del usuario y de certificación.

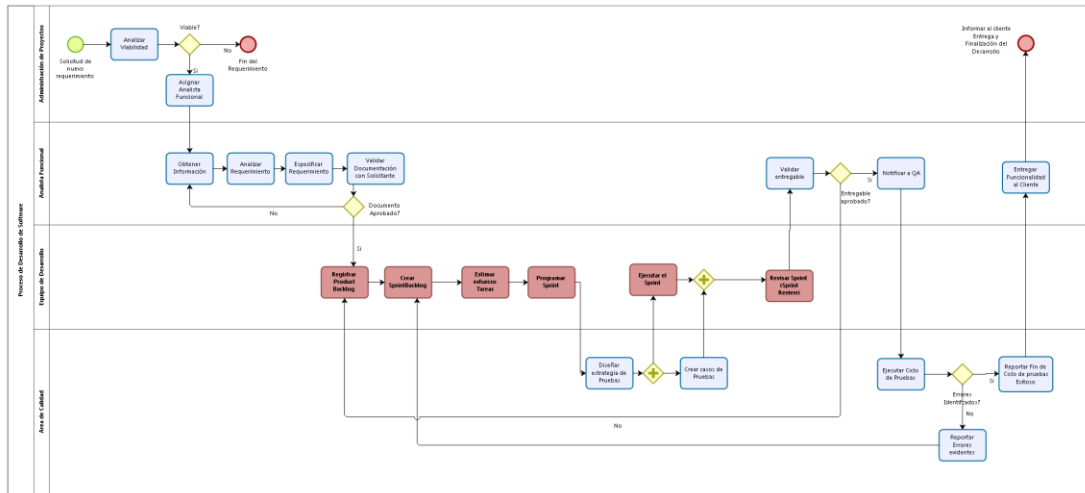


Ilustración 3 Proceso desarrollo de Software de la Compañía Objeto del proyecto (Diagrama elaborado para este proyecto).

3.10. Selección de Medidas

A continuación se relacionan las métricas seleccionadas⁹[38] que se aplicarán a los proyectos control y experimento, estas variables, son consideradas a partir de un común denominador que permita comparar resultados diferenciadores en la implantación de la guía [52], [53]. A continuación, se relaciona una tabla en el que resumen las medidas seleccionadas y algunas de sus características:

Métrica	Tipo escala	Descripción	Responsable Captura
Cantidad De Requerimientos Definidos en los Proyectos [54], [55]	Ordinal	Comparar entre los proyectos objetos de medición a partir de la cantidad de requerimientos definidos en el Product Backlog	Scrum Master/ Gerente de proyectos
Velocidad Real Promedio De Sprints De Los Proyectos[56], [57]	Ordinal	Número de horas reales consumidas por el equipo de desarrollo por cada Sprint.	Scrum Master/ Gerente de proyectos
Porcentaje Valor Ganado (Planeado Vs Entregado) [40], [58]	Ordinal	Porcentaje de productividad del equipo de desarrollo.	Scrum Master/ Gerente de proyectos

⁹ Para conocer el detalle del proceso de las medidas seleccionadas para este proyecto, remitirse al anexo 12 “Definición_variables_a_medir.docx”.

Métrica	Tipo escala	Descripción	Responsable Captura
Horas Invertidas Para Entendimiento De Los Requerimientos [54], [55]	Ordinal	Cantidad de horas que requiere el equipo de desarrollo para entendimiento de los requerimientos.	Scrum Master/ Gerente de proyectos
Horas Reales Tomadas Para Desarrollo ¹⁰	Ordinal	Cantidad de horas que requiere el equipo de desarrollo para cumplir con cada uno de los entregables por cada proyecto.	Scrum Master/ Gerente de proyectos
Horas Reales Para Pruebas Internas [59]	Ordinal	Horas totales reales destinadas para las pruebas funcionales en cada uno de los proyectos.	Scrum Master/ Gerente de proyectos
Cantidad de defectos en fase de pruebas internas[60]	Ordinal	Son la cantidad total de defectos identificados durante las pruebas realizadas por el equipo de desarrollo en cada uno de los proyectos.	Scrum Master/ Gerente de proyectos
Cantidad de horas Consumidos en Resolución de Incidentes ¹¹	Ordinal	Son la sumatoria de horas totales que le tomó al equipo de desarrollo resolver los casos reportados como incidentes durante las pruebas internas por cada proyecto.	Scrum Master/ Gerente de proyectos
Cantidad de defectos encontrados en fase de pruebas funcionales del usuario[60]	Ordinal	Es la cantidad de errores o incidentes totales reportados por los usuarios funcionales de cada proyecto una vez entregada la versión funcional de los resultados de los Sprint.	Scrum Master/ Gerente de proyectos
Entrevista al equipo de trabajo sobre su percepción de la aplicación de la guía metodológica	Nominal	Aplicación de encuesta de aceptación bajo la metodología (TAM[61]) al equipo que participo en los proyectos[62].	Scrum Master/ Gerente de proyectos

Tabla 2 Resumen de medidas seleccionadas y sus características

Después de estructurar cada una de las medias, es necesario detallar algunas otras características, por ejemplo, donde se almacenarán los datos recolectados, el responsable de recolectar estos datos, la periodicidad con que se capturan, como se analizará cada medida, instancia en el tiempo en que tomará dicha información o bajo que consideraciones, se debe tener en cuenta para alimentar los datos que generé confiabilidad para el objetivo que se pretende alcanzar (esta información se encuentra detallada en el anexo 12).

3.11. Encuesta basada en modelo de aceptación de tecnología (TAM)

En este proyecto, uno de los objetivos específicos es evaluar los resultados de la aplicación de la guía en proyectos reales de desarrollo de Software. Esta evaluación contiene dos tipos de análisis valorativo, uno a partir de resultados objetivos, partiendo de las métricas establecidas

¹⁰ La métrica seleccionada se argumenta en el principio de libre escogencia de la ISO 15939:2007 [37] al tomar las medidas que se considere y genere un valor en la comparación de los proyectos.

y los resultados cuantitativos que se pueden generar, al comparar dos proyectos de similares condiciones[15] donde solo en uno de estos se aplicará la guía.

El segundo tipo de análisis se basa en la apreciación subjetiva, donde se evalúa la Guía desde la percepción respecto a la facilidad de uso y su utilidad, en un proceso tecnológico[63], en este caso, en las actividades correspondientes a la ingeniería de requerimientos dentro de un proceso de desarrollo de software híbrido, entendiendo su aplicación en el contexto del principio de SEMAT en el que define la ingeniería de software como un conjunto de patrones a ejecutar[14], por lo que se podría afirmar, que la Guía[13] puede aplicarse en cualquier proceso independiente de la metodología de desarrollo de software en que se trabaje, aun aclarando, que el reto no está en la aplicación misma de la guía, sino en la adaptación que puede tener la guía en un proceso de desarrollo real, en este caso, en una compañía de la industria del Software Colombiana, sin perder o intervenir en el conocimiento tácito adquirido a través de los años por la compañía objeto del proyecto. Por lo anterior es importante resaltar que el conocimiento tácito constituye un activo valioso que se ha construido según el modelo de negocio de la compañía, ajustando las líneas de producción de desarrollo de software, por lo que para la industria, esta experiencia, en muchos casos, no es deseable sustituirla y mucho menos desaprovecharla, es por esta razón, lo que se pretende, es complementar este conocimiento proporcionando un beneficio a nivel de gestión y control de proyectos, en la etapa inicial donde se acuerdan los requerimientos.

En este proyecto para obtener el resultado subjetivo, se realiza mediante la aplicación de “*technology acceptance model*” por su siglas en inglés (TAM)[61].

Por lo anterior, se procede a diseñar un modelo de encuesta¹², en que la primera tarea es definir a quien va a estar dirigida, el caso de este experimento, se realiza sobre los roles que están directamente relacionados al proceso de Ingeniería de Requerimientos, se trata del Analista Funcional y Gerentes de Proyectos.

Partiendo de lo anterior, se define el conjunto de preguntas enfocadas a las variables a medir, basado en la percepción respecto a la utilidad y facilidad:

Variable	Pregunta
Facilidad	¿Nivel que considera que la guía metodológica es intuitiva y fácil de leer?
Facilidad	¿Nivel que considera que se debe tener de conocimiento en ingeniería de Software para entender la guía?
Facilidad	¿Qué nivel considera usted que la guía aportaría al proceso de ingeniería de requerimientos?
Facilidad	¿Qué nivel considera que la guía es útil en metodologías ágiles?
Facilidad	¿Qué nivel de acoplamiento puede tener la guía en ecosistemas ágiles?
Facilidad	¿Qué tan positivo considera la inclusión de la guía en proyectos de desarrollo de Software?
Facilidad	¿Cuándo leyó por primera vez la guía, como consideró la intención que tuvo para implantarla en un ejercicio práctico de ingeniería de Software?

¹² Ver formato de encuesta de aceptación en el anexo 17 FO-01- Aceptación

Variable	Pregunta
Utilidad	¿Nivel que considera que la Guía metodológica es concreta?
Utilidad	¿Nivel que considera que la guía es aplicable a los escenarios reales de desarrollo de software
Utilidad	¿Considera que la guía es útil aplicándose en metodologías tradicionales?
Utilidad	¿Qué tanto favorece la guía en el desempeño del equipo de desarrollo?
Utilidad	¿Considera que los modelos propuestos por la guía son de utilidad?
Utilidad	¿Las alfas descritas en la guía son útiles para la gestión y definición de los requerimientos?
Utilidad	¿Bajo qué nivel recomendaría para que otros equipos de desarrollo de software implanten la guía?
Utilidad	¿Qué tanto considera usted que la guía favorece al refinamiento de los requerimientos?

Tabla 3 Preguntas seleccionadas para la encuesta de aceptación

A partir de la selección de estas preguntas, se procede a realizar la evaluación a los actores anteriormente definidos en las siguientes etapas del proyecto experimento.

- Al finalizar la fase de aprendizaje de la guía (Inducción).
- Al finalizar la ejecución del proyecto Piloto.
- Al finalizar la ejecución del proyecto experimento.
- Al Evaluar el refinamiento propuesto después de ser expuesto.

Al finalizar con la última encuesta (Ejecución proyecto Experimento), la totalidad de los resultados serán analizados y puestos en comparación para generar conclusiones en la que se espera conocer en qué etapa del proyecto, la guía genera el nivel más alto de aceptación, respecto a la percepción de utilidad y facilidad de uso, así mismo, que etapa tiene los resultados menos favorables.

3.12. Definición de técnicas estadísticas para comparación de resultados

En este contexto, la utilidad de técnicas estadísticas nace de la variabilidad que puede surgir de la observación del comportamiento de los proyectos, dentro de sus procesos de desarrollo bajo condiciones aparentemente estables[64].

Para el esquema de análisis de datos propuesto para este proyecto, hace suponer el uso de la técnica que estén orientados al “diseño de experimentos” [65], en este escenario, estadística descriptiva.. Este tipo de estadística, se utiliza con fin de comprender mejor la naturaleza de los datos e identificar puntos de datos anormales o falsos (llamados valores atípicos)[15].

3.13. Técnica Estadística

Dentro de la ejecución del Quasi – Experimento[17], se estarán recolectando los datos necesarios para la medición y posterior comparación, a partir de técnicas estadísticas como lo son el análisis de las diferencias de medias y el análisis de varianza, con estas técnicas, se pretende comprobar las diferencias entre cada una de las medidas capturadas durante el desarrollo del proyecto [52].

4. EJECUCIÓN DE PROYECTOS

En este segmento se presenta la descripción de cómo se ejecutó el proyecto experimental bajo el diseño de investigación Quasi-Experimental, se referencia la preparación del equipo respecto al conocimiento de la guía y sus elementos, los roles que participaron en la ejecución, las consideraciones de los proyectos piloto, experimento y control y las características de la aplicación de apoyo construida para almacenar y administrar los datos de las medidas seleccionadas para la comparación de los proyectos.

4.1. Inducción al equipo de trabajo a la guía IMFIR-PDA y modelo del Kernel SEMAT

Para la implantación de la Guía en el equipo de trabajo, se ejecutaron actividades orientadas al aprendizaje, como presentaciones magistrales, lecturas de la guía y algunas otras investigaciones literarias sobre aproximaciones de aplicaciones de SEMAT. El objetivo es plantear un mecanismo apropiado de socialización y enseñanza de los componentes de la guía IMFIR-PDA al equipo de trabajo, que estará participando en los proyectos piloto y experimento.

A partir de una búsqueda de antecedentes sobre la guía IMFIR-PDA[13], no se identificaron registros formales donde haya sido implantada en proyectos de desarrollo reales en Colombia, a raíz de esto, se realiza una investigación de posibles aplicaciones de SEMAT en proceso de ingeniería¹³, de ser posible de la industria de software nacional, o en una segunda instancia, aplicada en países latinoamericanos quienes tienen un entorno de trabajo similar.

Cursando esta investigación, se realizó el contacto con personas activas en agilísimo y en promover a SEMAT en el campo académico, entre estas personas está Luis Antonio Salazar[66] quien ha sido un proponente de la teoría de SEMAT en conferencias y seminarios [67], Salazar afirma¹⁴: *“en Colombia, si bien se resalta el interés en el sector por promover los fundamentos y conceptos teóricos, aun no se logra superar los límites académicos puesto que SEMAT es un teoría relativamente reciente”*, sin embargo, desde este mismo sector académico es donde se ha incursionado en la utilización del *Kernel*[14], a partir de propuestas e investigaciones de proyectos grado, “paso previo” para abrirse camino en el mundo empresarial del Software. Dentro del sector académico, como representante está el profesor Carlos Mario Zapata presidente del “Comité ejecutivo del capítulo de SEMAT para Latinoamérica” [30] y profesor de la universidad Nacional de Colombia, quien confirma y da muestra del interés que tiene este sector por mejorar los procesos de ingeniería de software y así mismo, promover su aprendizaje desde un sentido crítico y al mismo tiempo propositivo, consiguiendo entender la ingeniería de Software de una manera común generando software de calidad y en tiempo reducido. No obstante, El registro literario de aplicaciones de SEMAT en Colombia (*Ver en detalle anexo 1:*

¹³ Ver Anexo 1 Resumen de aplicaciones de SEMAT en Latinoamérica. docx.

¹⁴ Esta afirmación se logra mediante una teleconferencia realizada con Luis Salazar, actividad de investigación realizada para este proyecto de grado.

Resumen de aplicaciones de SEMAT en Latinoamérica), se originan desde proyectos a nivel Postgrados, ejemplo de esto es la investigación que tiene como título “*Una representación basada en Semat y RUP para el Método de Desarrollo SIG del Instituto Geográfico Agustín Codazzi*”[68] de la universidad Nacional en su sede de Antioquia.

4.2. Roles involucrados en los proyectos de ejecución

A continuación, se relacionan los roles que ejecutaron los proyectos de prueba piloto y los involucrados en el diseño Quasi – Experimental (Experimento y Control):

- **Sponsor:** Persona que patrocina económicamente el proyecto y quien toma las decisiones definitivas por parte del cliente, normalmente es el jefe de cartera o algún representante responsable de la Vicepresidencia Financiera.
- **Stakeholder:** Son todas aquellas que estarán relacionadas a este proyecto de manera indirecta o directa, la intervención de estas personas afecta al proyecto de forma positiva o negativa.
- **Usuario Líder (Product Owner):** Persona que lleva la representación de todo el equipo de usuarios por parte del cliente en términos de definiciones funcionales del requerimiento.
- **Equipo de trabajo:** Personas del equipo proveedor de diferentes disciplinas que aportan a la consecución de la solución entorno al requerimiento definido por el cliente.

A continuación, se contextualiza las funciones de los roles que participan de los diferentes equipos de trabajo¹⁵.

Ingeniero de Desarrollo: Desarrolla Software de acuerdo con las especificaciones de calidad de la empresa y los criterios determinados por el cliente.

Ingeniero de pruebas / Tester: Encargado de asegurar que los productos y servicios de la organización sean revisados en todo el alcance del proyecto, para lograr la satisfacción del cliente final en cuanto al diseño y desarrollo de los productos.

Director de Proyectos: Dentro de la compañía donde se ejecuta el proyecto de grado, la dirección de proyectos se basa en metodología PMI[69] y sus funciones están orientadas a una PMO por sus siglas en inglés (*project management office*), en este caso administra los proyectos y los recursos generales, además de realizar un seguimiento y adecuado control de estos mismos.

Gerente de Proyectos (Scrum Master): Realizar actividades administrativas de planificación, organización, dirección y control de los recursos a su cargo (personal, presupuesto, equipo y

¹⁵ Para conocer las características (nivel de experiencia, formación técnica y competencias) de cada perfil, remitirse al anexo 11 Perfiles_involucrados_proyectos.xls

materiales) para satisfacer los requerimientos técnicos, de costo y de tiempo, que permitan finalizar con éxito el o los proyectos bajo su responsabilidad, según se haya presupuestado.

Analista Funcional: Rol encargado del proceso de ingeniería de requerimientos, desde la fase de obtención hasta la fase misma de validación, también encargado del mantenimiento del requerimiento dentro de este proceso.

Analista Funcional Líder: Rol con un nivel de experiencia superior al de un analista funcional, también coordinar el equipo de analistas funcionales, ejerciendo control y seguimiento a las actividades asignadas a este equipo.

Ingeniero implementador: Ejecutar la implementación del software comercial en nuevos clientes.

Líder Técnico: Apoya a las tareas a los desarrolladores, además genera lineamientos a partir del conocimiento, solucionando situaciones técnicas que se presenten durante la ejecución de los diferentes proyectos

Consultor: Especialista en la aplicación de Software de cobranzas con experiencia en administración y gestión de cartera.

4.3. Metodología y tiempo de apropiación de la Guía

Para el aprendizaje, se comparte la guía por correo electrónico a cada uno de los integrantes, el objetivo de compartirla es exponer la guía para una apropiación directa sin intervenciones, generando un conocimiento y al mismo tiempo un punto de vista por parte de los integrantes de manera individual, que permita la discusión acercando a un primer resultado de aceptación de uso de la guía como se pretende a partir de (TAM) [70].

Posterior a esto, se convoca a una reunión donde participan cada una de los integrantes, allí se lee la guía segmento por segmento, se interpretan los conceptos y se aclaran las inquietudes, de tal forma, que los integrantes empiecen a cerrar la brecha entre lo entendido y la teoría de SEMAT aplicada a la guía[13].

Para la siguiente reunión orientada a la inducción de la Guía[13], además de compartir los resultados de la investigación y hacer un repaso general de la guía, la persona con el rol de gerente de proyectos, propone los proyectos donde se ejecutará la prueba piloto y la ejecución Quasi - Experimental.

4.4. Elementos usados de la guía por el equipo de trabajo

Una vez se selecciona el proyecto, se procede a realizar un diagnóstico a partir de los componentes que ofrece la guía, puesto que la guía permite tomar los componentes que se adecuen al proceso en el que se esté implementando al criterio del equipo de trabajo, por esta razón se define los componentes que se tomarán de la guía para implantarlos dentro del proyecto piloto y el proyecto Experimento.

En acuerdo con el equipo de trabajo, en términos de utilidad y evitar generar un trauma en el proceso de análisis y especificación, se acuerda usar únicamente los siguientes modelos propuestos por la guía:

- Modelos de Objetivos de Negocio.
- Organigrama.
- Modelo de Flujo de Procesos.

De los demás diagramas, no se consideran, puesto que implicaría retrasar los avances y compromisos adquiridos con el cliente, también teniendo en cuenta, que ningún integrante del equipo de trabajo tiene experiencia en aplicar estos modelos, por lo que la primera vez puede tomar más del tiempo comprometido.

En cuanto al proceso de Ingeniería de requerimientos, se planea hacer uso de todos sus puntos atendido a las recomendaciones de las alfas a usar por cada etapa del proceso.

4.5. Selección de proyectos

Para este proyecto de grado se plantea la ejecución de tres proyectos: el proyecto piloto, proyecto control y proyecto experimento, de esta manera, los proyectos deberán tener características similares en su entorno para los casos del proyecto control y experimento. De tal forma como se presentó en la propuesta de este proyecto de grado, su marco de trabajo está delimitado bajo las siguientes condiciones: una estimación mínima de 400 horas de ejecución entre construcción y pruebas, catalogarse como un proyecto de nivel complejo según criterio de la compañía donde se ejecuta el proyecto y por último, el entorno de desarrollo natural se fundamente en prácticas ágiles que para este caso, son prácticas que permita entregas tempranas y comportamientos adaptativos.

4.6. Proyecto Piloto

Para la selección del proyecto Piloto, se pone en consideración factores especiales que por experiencia de la compañía, son las dificultades más comunes en la ingeniería de requerimientos[11] identificadas en el marco teórico de este proyecto. La idea de ejecutar un proyecto piloto, es la de preparar al equipo, enfrentándolos a una primera experiencia que de forma iterativa, enriquezca la apropiación de la guía para que en el momento de la ejecución proyecto experimento, no existan escenarios desconocidos que afecte negativamente el resultado de este proyecto.

Los factores de decisión de este proyecto son[9], [11]:

- Ubicación aislada entre los interesados – usuarios y el equipo de desarrollo.
- Demora y reprocesos en proyectos anteriores por mala comunicación entre los interesados – usuarios y el equipo de desarrollo.
- Relación con el cliente relativamente desgastada, oportunidad de mejora, que a partir de directrices de la guía mediante sus alfas, implica una comunicación más asertiva y frecuente.
- La estimación del requerimiento inicial es aproximada a la requerida para adelantar el proyecto piloto.

Características del Proyecto Piloto

A continuación, se resumen, las siguientes características para el proyecto piloto:

- **Tipo de Proyecto:** Mejora de la aplicación implementada para agilizar un proceso puntual.
- **Nivel del proyecto:** Medio¹⁶.
- **Horas estimadas para el desarrollo de este requerimiento:** (80 – 120) horas.
- **Roles que intervendrán en este proyecto:** A continuación, se describe los roles y experiencia de las personas que participan para el proyecto Experimento:

Equipo de trabajo Proveedor de Software¹⁷:

- Un analista funcional con 2 años de experiencia.
- Un gerente de proyectos con 3 años de experiencia en gestión de proyectos.
- Un Líder técnico que estará acompañando en esta definición con más de 5 años en conocimiento funcional, técnico y de negocio.
- Desarrollador con experiencia 3 años dentro de la compañía como ingeniero de desarrollo y otros 2 años más de experiencia en otras compañías.
- Un ingeniero implementador con 1 años de experiencia en configuración de componentes de la aplicación.

Equipo definido con el cliente:

- Un usuario experto Jurídico.
- Un usuario líder con experiencia en la aplicación y que realiza control a la operación de cobranza actual.
- Un asesor externo con experiencia en ejecución de procesos operativos o de negocio tercerizados (BPO)¹⁸.
- Gerente de proyecto que estará validando los avances y estará acompañando al usuario para garantizar la asesoría en aspectos técnicos.
- Sponsor, que aprueba el proyecto, evalúa periódicamente los avances del proyecto, decide los aspectos de negocio que sean escalados por parte del usuario o gerente del proyecto.

¹⁶ Criterio de medición utilizado por el equipo de desarrollo para segmentar los tipos de requerimientos por parte de la compañía objeto del proyecto, en este caso Mediano se considera por aquellas funcionalidades que necesitan ser extendidas, cambiar el comportamiento de una funcionalidad, o implementar un nuevo componente.

¹⁷ Para conocer las características (nivel de experiencia, formación técnica y competencias) de cada perfil, remitirse al anexo 11 Perfiles_involucrados_proyectos.xls

¹⁸ Business Process Outsourcing (BPO)

4.7. Proyecto Control

El proyecto Control hace parte de los grupos de proyectos a analizar bajo el diseño Quasi – Experimental[17], en este caso, es el grupo o proyecto que no es afectado con la inserción de un artefacto, solo se realizan actividades de observación y tomas de datos sobre las medidas seleccionadas, sin intervención alguna que sesgue los resultados esperados. Este tipo de escenarios se denomina “Grupo control no equivalente”, aclarando que, aunque el grupo control respecto al grupo Experimento poseen características homogéneas (cantidad de personas, experiencia del equipo, complejidad del proyecto, tipo de proyecto, roles que participan y horas estimadas) no es exactamente el mismo equipo del proyecto Experimento por lo que se considera no equivalente[21].

Consideraciones de la ejecución del Proyecto Control

En esta ocasión, una compañía AFP solicita implementar el módulo de cobro Jurídico en su versión estándar, agregándole a su vez, funcionalidades que requieren de desarrollo de Software, en este contexto de trabajo, se describen las consideraciones por el cual este proyecto es seleccionado como “Control” y punto de referencia para el proyecto “Experimento”. Estas consideraciones son:

Alto volumen de reglas de negocio Este proyecto cuenta con requerimientos de alto impacto para el proceso de cobranza, esto se debe a que algunos de estos, implica gestión, control y liquidación, de gastos y honorarios en los que derivan a la afectación del presupuesto principal de la compañía cliente, el volumen de reglas identificadas son significativas para esta compañía proveedora de Software, agregando la agudeza que implica la criticidad del afectar directamente el presupuesto.

Falta de claridad en el alcance por parte del cliente: Este proyecto cuenta con definiciones que operacionalmente ya existen bajo un procedimiento, donde no cuentan con un apoyo tecnológico de aplicaciones empresariales, diferentes a las herramientas ofimáticas conocidas, por esta razón, el alcance tiende a especificarse mucho más rápido en comparación con el proyecto “Experimento”, sin embargo, la poca disponibilidad de tiempo por el usuario líder y de la reducida información suministrada para dar inicio al proyecto, generan un riesgo importante en la definición funcional de la aplicación, esta situación se amplifica por la adversidad que implica una comunicación remota para la gestión de requerimientos con este cliente.

Relación con el cliente: Uno de los inconvenientes más relevantes, es la distancia entre el cliente y el proveedor, este cliente se encuentra ubicado en la ciudad de San Salvador (Salvador) y la sedes de la compañía desarrolladora de Software se encuentre ubicadas en Bogotá y Medellín(Colombia), por lo que en acuerdos contractuales enmarcados en un modelo SaaS, los servicios ofrecidos se ejecutan de manera remota mediante teleconferencias y/o videoconferencias, para la gestión y atención de nuevos requerimientos y sus actividades derivadas como las reuniones, las sesiones de levantamiento de información, presentaciones, demostraciones y pruebas con el cliente.

Características del Proyecto Control:

- **Tipo de Proyecto:** Subyace del proceso de cobranza jurídica para construir componentes adicionales que soporte el proceso jurídico de manera específica.
- **Nivel del proyecto:** Grande¹⁹.
- **Horas estimadas para el desarrollo de este requerimiento:** (600 – 1000) horas.
- **Roles que intervendrán en este proyecto:** A continuación, se describe los roles y experiencia de las personas que participan para el proyecto Control:

Equipo de trabajo Proveedor de Software:

- Un líder de analistas funcionales con 3 años y 6 meses de experiencia quien estará apoyando la definición en términos funcionales, técnico y de negocio.
- Un gerente de proyectos con 5 años de experiencia en gestión de proyectos y 2 años como Líder Técnico.
- Un consultor de negocio con más de 10 años de experiencia que estará acompañando en el proceso de definición del requerimiento orientándolo a una solución de Software.
- Desarrollador con experiencia de 2 años y 8 meses de la compañía como ingeniero de desarrollo.
- Un tester 3 años de experiencia como analistas de calidad.
- Un ingeniero implementador con 3 años de experiencia en configuración de componentes de la aplicación.

Equipo del cliente:

- Un usuario líder del proceso de cobranzas Jurídico, en este caso, no posee un conocimiento experto en los módulos previamente implementados de la aplicación, sin embargo, conoce a detalle el estado actual del proceso donde se incorporará la solución de estos nuevos componentes, por esta razón es encargado de la definición del alcance, pruebas y aceptación del producto solicitado.
- Gerente de proyecto que estará validando los avances y estará acompañando al usuario para garantizar la asesoría en aspectos técnicos.
- Sponsor, que aprueba el proyecto, evalúa periódicamente los avances del proyecto, decide los aspectos de negocio que sean escalados por parte del usuario o gerente del proyecto.

¹⁹ Criterio de medición utilizado por el equipo de desarrollo para segmentar los tipos de requerimientos por parte de la compañía objeto del proyecto, en este Generalmente para nuevas funcionalidades o módulos que requieren cambios en el core (framework).

4.8. Proyecto experimento

En el Proyecto Experimento, es donde se implantará La Guía, siendo esta, el artefacto a introducir dentro del proceso de desarrollo de Software en esta compañía, adaptado al proceso de nuevos requerimientos en el que afectará las variables dependientes[15].

Por otro lado, el equipo del proyecto experimento, ha madurado en cuanto a la apropiación de la guía a partir de la ejecución del proyecto piloto, obteniendo de esto lecciones aprendidas, cerrando algunas brechas de conocimiento y mejorando la adaptación de la guía al modelo de negocio con que opera la compañía objeto de este proyecto.

Consideraciones de la ejecución del Proyecto Experimento

Este proyecto tiene tres grandes consideraciones por el cual fue seleccionado como proyecto experimento, estas consideraciones son:

Alto volumen en la definición de reglas de negocio: Este requerimiento tiene un componente configurable muy alto, por el cual está sujeto a la variación que pueda presentar en la definición de reglas, según el comportamiento del modelo de cobranzas implementado.

Falta de claridad en el alcance por parte del cliente: Este requerimiento nace de una iniciativa para apresar un segmento de clientes, que por ley protege a las cooperativas en términos crediticios adquiridos por sus afiliados. Este argumento, que tiene como expectativa generar una gran efectividad en la cartera, es principiante y poco explorado en el sector de las cooperativas, aunque la ley 79 tiene descrito el entorno del procedimiento de cobranza para quienes apliquen, de momento no hay evidencia de procesos bien definidos que logre su objetivo, no existe indicadores ni métricas conocidas que sirva como fuente de referencia para estructurar un proceso como este hasta el momento. Por esta razón la cooperativa Z (por términos de confidencialidad el nombre se mantiene en reserva), conociendo la oportunidad que se tiene al aplicar esta implementación, desea incursionar en esta solución, teniendo claro que su alcance no está completamente definido en cuanto al procedimiento que deberá llevar la solución de software, adicionando, que actualmente el equipo encargado por parte de la cooperativa, para sacar adelante este proyecto, cuenta con más de 5 personas opinando desde el ámbito legal como de su operatividad.

Relación con el cliente: Esta cooperativa ha crecido vertiginosamente siendo de las cooperativas líderes del mercado, teniendo como aliado estratégico la empresa desarrolladora de software. Por esta misma razón, la relación ha sido muy exigente haciendo valer la calidad del servicio ofrecida por la compañía proveedora de software. Sin embargo, la situación de poder contar con un cliente crítico y exigente (Cooperativa Z), genera una oportunidad de permitir involucrarlo en el proceso de desarrollo de manera transversal, siguiendo la sugerencia de SEMAT, que desde las alfas señaladas en la guía, manifiesta como una de las variables primordiales para la ejecución del proyecto.

Características del Proyecto Experimento

- **Tipo de Proyecto:** Subyace del proceso de cobranza jurídica, creando una alternativa de gestión de cobro al proceso de cobranza actual.
- **Nivel del proyecto:** Grande²⁰.
- **Horas estimadas para el desarrollo de este requerimiento:** (600 – 1000) horas.
- **Roles que intervendrán en este proyecto:** A continuación, se describe los roles y experiencia de las personas que participan para el proyecto Experimento:

Equipo de trabajo Proveedor de Software²¹:

- Un Analista funcional con 2 años de experiencia quien estará apoyando la definición en términos funcionales.
- Un gerente de proyectos con 3 años de experiencia en gestión de proyectos.
- Desarrollador con experiencia de 2 años y 5 meses de la compañía como ingeniero de desarrollo.
- Un teste 3 años de experiencia como analistas de calidad.
- Un ingeniero implementador con 1 años de experiencia en configuración de componentes de la aplicación.

Equipo del cliente:

- Un usuario experto Jurídico.
- Un usuario líder con experiencia en la aplicación y que realiza control a la operación del proceso Ley 79.
- Un asesor externo con experiencia en ejecución de procesos operativos o de negocio tercercerizados (BPO)²².
- Gerente de proyecto que estará validando los avances y estará acompañando al usuario para garantizar la asesoría en aspectos técnicos.
- Sponsor, que aprueba el proyecto, evalúa periódicamente los avances del proyecto, decide los aspectos de negocio que sean escalados por parte del usuario o gerente del proyecto.

4.9. Resumen de la Ejecución de proyectos

Para la ejecución de los proyectos, se anexan tres documentos que describen a detalle cómo se llevaron a cabo los proyectos Piloto, Control y Experimento. Se describe como fue el aporte de

²⁰ Criterio de medición utilizado por el equipo de desarrollo para segmentar los tipos de requerimientos por parte de la compañía (SOLATI) objeto del proyecto, en este Generalmente para nuevas funcionalidades o módulos que requieren cambios en el core (framework).

²¹ Para conocer las características (nivel de experiencia, formación técnica y competencias) de cada perfil, remitirse al anexo 11 de Perfiles_involucrados_proyectos.xls

²² Business Process Outsourcing (BPO)

la guía (Proyecto Experimento) y que eventualidades surgieron para esta implantación, sobre estos documentos se encuentra relacionados los soportes de bitácoras de horas consumidas, el de defectos detectados en los desarrollos y el seguimiento de horas consumidas en los Sprints, los documentos anexos son:

Anexo Principal	Soportes Adicionales
Anexo 7: Resumen Ejecución Proyecto PILOTO.docx	<ul style="list-style-type: none"> • Anexo 4: Bitácora_Horas_Consumidas_Proyecto_PILOTO.xlsx • Anexo 10: Errores_y_Tiempos_Proyecto_PILOTO.xlsx • Anexo 15: Valor_Ganado_Velocidad_Sprint_PILOTO.xlsx
Anexo 8: Resumen Ejecución Proyecto CONTROL.docx	<ul style="list-style-type: none"> • Anexo 2: Bitácora_Horas_Consumidas_Proyecto_CONTROL.xlsx • Anexo 8: Errores_y_Tiempos_Proyecto_CONTROL.xlsx • Anexo 13: Valor_Ganado_Velocidad_Sprint_CONTROL.xlsx
Anexo 9: Resumen Ejecución Proyecto EXPERIMENTO.docx	<ul style="list-style-type: none"> • Anexo 3: Bitácora_Horas_Consumidas_Proyecto_EXPERIMENTO.xlsx • Anexo 9: Errores_y_Tiempos_Proyecto_EXPERIMENTO.xlsx • Anexo 14: Valor_Ganado_Velocidad_Sprint_EXPERIMENTO.xlsx

Tabla 4 Tabal de anexos y soportes adicionales en ejecución de proyectos.

Los soportes adicionales tienen la siguiente descripción:

- **Bitácora de horas consumidas por cada proyecto**

Documento diligenciado en formato de hoja de cálculo en el que se registran datos de fecha, actividad, responsable y tiempo real que toma cada actividad que constituye las etapas correspondientes al proceso de Ingeniería de Requerimientos (obtención, análisis, especificación y validación), Diseño, Desarrollo, pruebas y despliegues.

- **Archivo de control de Sprints por cada proyecto**

Documento administrado por el gerente de proyectos asignado, en el que lleva el control de horas consumidas en las actividades ejecutadas por cada Sprint en fase de desarrollo y pruebas según proceso híbrido de desarrollo de Software, en este documento se lleva registro de velocidad de cada Sprint y el valor ganado según tareas atendidas según planeación.

- **Errores y Tiempos**

Documento que contiene únicamente los defectos aceptados por el equipo de desarrollo, registrados en el sistema de gestión de incidencias de la compañía desarrolladora de Software, este detalle se encuentran en una hoja de cálculo, discriminado por los defectos reportados en las fase de pruebas funcionales ejecutadas por el Tester del equipo asignado, y los defectos resultantes de pruebas funcionales realizadas por el usuario, además de esto contempla el tiempo que le tomó al ingeniero de desarrollo dar solución a cada caso reportado.

4.10. Construcción de aplicación de software como apoyo a la recolección de datos del proyecto experimento

Para este proyecto, se construyó una aplicación de Software que permite administrar (CRUD²³) de las medidas y sus características, para que posteriormente se permita almacenar los datos correspondientes a cada una de estas, estas medidas pueden ser seleccionadas para un proyecto o varios proyectos (Ej: Piloto, Control, Experimento). En el caso que una medida aplique para más de un proyecto en común, la aplicación permite realizar comparaciones graficas usando la técnica de desviación estándar.

Esta aplicación, más allá de soportar funcionalmente la captura, almacenamiento y graficas de los datos en este proyecto experimental, también está diseñada, para soportar otros tipos de proyectos que cuenten con un contexto de investigación experimental similar, aportando a posibles trabajos futuros a este proyecto de grado o cualquier otra iniciativa de ámbito académico. A Continuación, se describen la características funcionales y técnicas de esta aplicación.

Características Funcionales:

- Administración de Medidas – CRUD.
- Administración de usuarios - CRUD con restricción de acceso.
- Administración de proyectos - CRUD.
- Asociación de medidas a diferentes proyectos.
- Grafica de resultados comparativos a partir de la variación y desviación estándar.

Características Técnicas:

- **Framework:** CakePHP[71].
- **Arquitectura de Software:** La aplicación cuenta con una arquitectura basada en “Modelo-Vista -Controlador”- (MVC).
- **Servidor HTTP:** Apache[72]. mod_rewrite (2.2 o Superior)
- **Bases de Datos:** MySQL Community Server (GPL) [73] (5.1.10 o mayor).
- **Lenguaje de programación:** PHP[74] (5.6.0 o mayor).
- **Licencia:** MIT license[75].

Para la creación de esta aplicación se realizó mediante practicas agiles, en el que se hizo uso de captura e historia de usuarios, posterior se procede al diseño de bases de datos, debido a que el Framework utilizado, recrea el diseño de clases y sus relaciones a partir del diseño de bases de datos relacional, facilitando la construcción de la aplicación concentrando el esfuerzo en la definición del modelo de datos, en la normalización y cardinalidad correspondiente. Como soportes documentales al análisis y diseño de esta aplicación se encuentran los siguientes entregables:

²³ CRUD: Acrónimo de Crear, Leer, Actualizar y Borrar" algún objeto funcional de una aplicación de software.

- Historia de usuarios (Anexo 23).
- Diagrama de clases (anexo 24).
- Diagrama de Entidad Relación (anexo 25).

La información correspondiente a su instalación se encuentra documentado en el anexo: Manual de instalación aplicación de apoyo.pdf, como también allí se referencia el manual de usuario de la aplicación.

5. RESULTADO DE LA EXPERIMENTACIÓN

Este segmento contempla la recopilación de los resultados de la ejecución experimental, se presenta el análisis realizado respecto a las métricas y la comparación que surge entre estas, además se pone en evidencia la interpretación de estos mismos resultados partiendo desde una observación a lo largo de la ejecución de este proyecto experimental en un contexto real.

5.1. Resultado de datos recolectados en los proyectos sobre medidas seleccionadas.

Partiendo del proceso de recolección de datos definido en el punto [3.5](#) se procede a la captura de datos y como parte del control para garantizar la confiabilidad y calidad de los datos; durante la ejecución de los proyectos, se programan reuniones de seguimiento con cada gerente del proyecto y se solicita la evidencia de los documentos que soportan estos datos (ver soportes adicionales tabla 4 del punto 4.6).

A partir de estos soportes, se generan mecanismos de seguimiento que proporcionan un espacio propicio para la orientación oportuna de la implementación de la Guía en la ejecución de los proyectos Piloto y Experimento. Como soporte a estos seguimientos, se registran actas de seguimiento²⁴.

Al finalizar la ejecución de los proyectos, los datos que se encuentran detallados en los soportes anteriormente descritos, se llevan al Software diseñado para consolidar los datos de las medidas (Ver punto 4.10).

En la tabla 5 se presentan los resultados consolidados de las medidas seleccionadas en el punto [3.5](#), además de esto, para efectos de análisis del resultado de la hipótesis del proyecto Quasi-Experimental, se consideran algunos otros datos que a través de la ejecución de los proyectos se identificaron como relevantes.

²⁴ Las actas de seguimiento se encuentran en el anexo 16 descritos en la sección [6.4](#) de este documento.

Medidas	Toma de medidas Proyecto Piloto	Toma de Medidas Proyecto Control	Toma de Medidas Proyecto Experimento	Observaciones
Medida Cantidad De Requerimientos Definidos Para Un Proyecto.	6	20	45	Cantidad de componentes identificados que se caracterizan por ser una unidad funcional desarrollable y posiblemente independiente.
Cantidad de casos de uso definidos	6	23	49	
Cantidad de Sprints totales en Desarrollo y Pruebas	4	18	16	Según la metodología de desarrollo de la compañía proveedora, solo aplica Sprints para fase de desarrollo y pruebas de los requerimientos.
Horas Invertidas Para Entendimiento De Los Requerimientos (Obtención y Análisis de requerimientos)	16,5	68,5	92,6	Horas reales que, para definir el requerimiento, hace parte de las variables a medir (Ver tabla 02).
Cantidad horas consumidas para especificación de los requerimientos	26	131	210	Horas reales definidas para la especificación de los requerimientos.
Horas Reales Tomadas Para Desarrollo	124	318	313	Hace parte de las variables de medición del proyecto (Ver tabla 02). Solo se tomaron las horas exclusivas para construcción sin contar con ajustes y solución de incidencias a nivel de desarrollo.
Medida Velocidad Real Promedio De Sprints De Los Proyectos.	89,1	92,0	91,7	Hace parte de las variables de medición del proyecto (Ver tabla 02).
Cantidad de Desarrolladores que participaron	1	1	1	
Medida Valor Ganado (Planeado Vs Entrega)	90%	91%	93%	Hace parte de las variables de medición del proyecto (Ver tabla 02).
Casos de pruebas diseñados	32	167	415	
Horas Reales Para Pruebas Internas	28	233	183	Solo se toman las horas consumidas de acuerdo a la planeación de ejecución de pruebas, no se tiene en cuenta las pruebas que surgen de los incidentes solucionados por desarrollo, ni tampoco las pruebas con los usuarios.
Cantidad de defectos en fase de pruebas internas	5	122	44	Hace parte de las variables de medición del proyecto (Ver tabla 02).
Cantidad de Defectos encontrados en fase de pruebas funcionales del usuario	3	17	9	Hace parte de las variables de medición del proyecto (Ver tabla 02).
Cantidad De horas Consumidos En Resolución De Incidentes	18	83	56	Hace parte de las variables de medición del proyecto (Ver tabla 02).
Cantidad De horas consumidas en pruebas de incidentes reportados	11	58	20	Hace parte de las variables de medición del proyecto (Ver tabla 02).
Total, de Horas consumidas para ejecución del proyecto	182,5	1054,3	1155,2	Sumatoria de horas totales consumidas para cualquier actividad relacionada a los proyectos ver anexo Bitácora Horas Consumidas Proyecto (Piloto, Control, Experimento).

Tabla 5 Resultados consolidados de las medidas seleccionadas

5.2. Análisis e Interpretación de datos

Para el análisis de los datos, se utiliza la estadística descriptiva para comparar los datos reco-

lectados, en este caso los datos del Proyecto Control (PC) Vs los datos del Proyecto Experimento (PE). Cabe resaltar que la comparación de estos datos, no se realiza directamente entre los resultados de los proyectos, sino se realiza, sobre la variación que existe entre los datos del mismo proyecto en términos de tiempo o fases, y los resultados de esta variación, son los datos que se comparan en este análisis.

Realizando un primer diagnóstico sobre los proyectos, el consumo de horas totales para la ejecución del PE es de **1155,2 Horas** y para el PC **1054,3 horas**, con una cantidad de Sprints ejecutados de **16 Sprints** para PE y **18 Sprints** PC, resultados que a simple vista y a diferencia de las situaciones presentadas por cada uno en sus ejecuciones; resultan aproximados y propicios para evaluar la hipótesis de este proyecto, con este preámbulo se presenta a continuación el análisis descriptivo entre las medida seleccionadas.

5.2.1. Análisis de Cantidad De Requerimientos Definidos

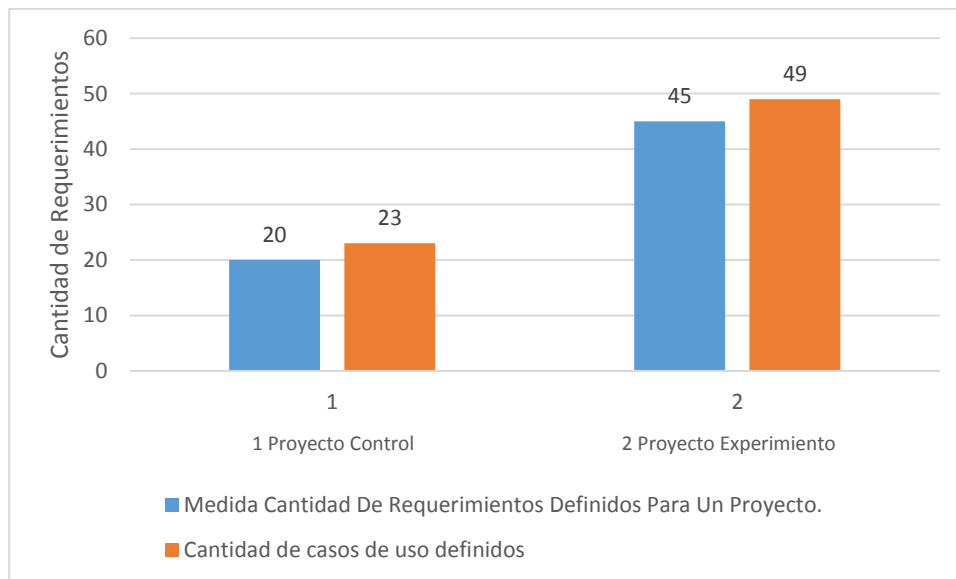


Ilustración 4 Comparación de requerimientos y casos de uso identificados por proyecto

Inicialmente como se puede observar en la Ilustración 4, hay una diferencia significativa entre los requerimientos establecidos para el PE (45 Requerimientos) Vs el PC (20 Requerimientos), de inmediato, se puede concluir que el PE es 2,25 veces mayor al PC en cuanto a la cantidad de requerimientos definidos, así mismo se presenta la relación en la cantidad de casos de uso, 23 para PE y 49 para PC, sin embargo, el punto de referencia para la comparación de los proyectos, está en la diferencia de la tasa que existe entre estos dos datos (Casos de uso y Cant. Requerimientos) por cada proyecto, en este caso se presenta en los siguientes resultados:

Proyecto	Requerimientos	Casos de uso	Tasa Promedio de casos de uso por Req.
Proyecto Control	20	23	1,15 CU / Req

Proyecto	Requerimientos	Casos de uso	Tasa Promedio de casos de uso por Req.
Proyecto Experimento	45	49	1,08 CU / Req

Tabla 6 Tasa promedio de diseño de casos de uso por cada requerimiento identificado

La relación presentada en la tabla 6, se puede entender como la cantidad de requerimientos documentados por cada caso de uso, basándose en “el como” el Analista Funcional interpreta un requerimiento en el que pueda describir los pasos o las actividades que deberán realizarse para llevar a cabo una funcionalidad. Bajo este análisis, se puede observar que la interpretación de los Analistas Funcionales para este tipo de requerimientos tiende a ser aproximados, pues cuentan con una desviación estándar de “0,04” entre estas tasas, siendo apropiado para la comparación de estos proyectos en las etapas de desarrollo de software siguientes.

5.2.2. Análisis de Horas Invertidas para Entendimiento de los Requerimientos

Para esta medida, solo se tiene en cuenta el tiempo consumido en actividades relacionadas a la Obtención y Análisis de Requerimientos, para convergir al punto en donde los Analistas Funcionales comprenden los requerimientos, sin antes pasar a etapas posteriores del proceso de Ingeniería de Requerimientos. Este consumo de horas (68,5 Horas para PC y 92,6 horas para PE) denominado con la sigla “HIPER” para esta comparación, si se analiza la cantidad de requerimientos identificados por cada proyecto, en relación al tiempo que le toma entender cada requerimiento se presenta los siguientes datos:

Proyecto	Cantidad de requerimientos	HIPER	HIPER X requerimiento
Proyecto Control	20	68,5 horas	0,29 horas / REQ
Proyecto Experimento	45	92,6 horas	0,48 horas /REQ

Tabla 7 Relación promedio de horas invertidas en entendimiento del requerimiento por cada requerimiento según proyecto

Como se puede observar, para el PE el Analista Funcional le toma **0,48 horas** entender un requerimiento antes de especificarlo, esta cifra equivale a un incremento del **60%** entre lo que le toma al analista funcional entender un requerimiento en el proyecto experimento Vs lo que le toma entender un requerimiento para el proyecto control. Este resultado se puede interpretar, que el incremento en el tiempo está relacionado al tiempo que dedico el Analista Funcional cumplir con las sugerencias de la guía para aceptar cada estado de las diferentes alfas en las etapas de obtención y análisis de requerimientos. A continuación, se muestra una gráfica representativa de este comportamiento:

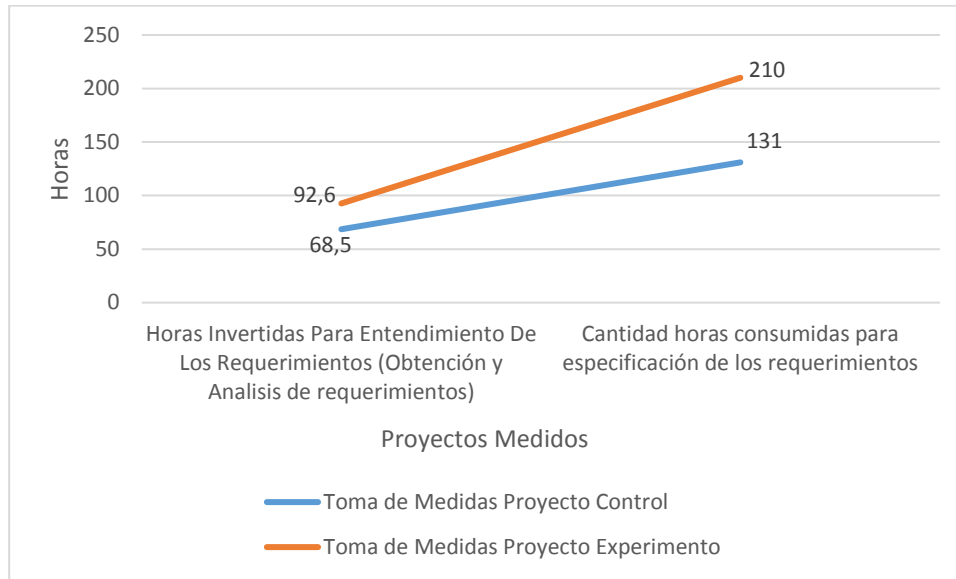


Ilustración 5 Comparación de horas consumidas entre entendimiento de los requerimientos Vs Tiempo consumido en especificación

Adicional a lo anterior, para poder profundizar en este análisis donde la Guía ofrece su metodología como aporte al proceso de desarrollo de Software, se incluye el tiempo consumido en fase de especificación (131 horas para PC y 210 horas para PE), con este nuevo suministro, se identifica una importante variación en el tiempo entre detallar un caso de uso y el tiempo consumido para especificación del mismo.

Proyecto	Casos de uso	Tiempo Especificación	Promedio horas en especificación X Caso de uso
Proyecto Control	23	131	6,7 horas / CU
Proyecto Experimento	49	210	4,28 horas / CU

Tabla 8 Relación promedio de horas de tiempo consumido en especificación por caso de uso

De la anterior tabla se puede interpretar que, para el PC existe una mayor dedicación por parte del Analista Funcional para construir los casos de uso, sin embargo, como se explica anteriormente, para este proyecto, los casos de uso cuentan con un número mayor de funcionalidades lo que implica realizar una documentación más extensa. Como soporte de este análisis, se realiza un promedio de páginas por documento de casos de uso (*FO-SOF-071-CASO DE USO*) y para el PC tiene en promedio 8 paginas, mientras que para el PE tiene un promedio de 5 páginas.

5.2.3. Análisis de horas consumidas en Desarrollo Vs pruebas

Para esta comparación se tienen en cuenta dos medidas, “Horas Reales Tomadas Para

Desarrollo” y “Horas Reales Para Pruebas Internas” para la primera, el PE cuenta con **313 Horas** dedicadas a construcción de código (No incluye actividades de ajustes y solución de incidencias), así mismo, para el PC el consumo de horas relacionadas a esta misma actividad es de **318 horas**.

En el caso de las pruebas, para este análisis solo se tiene en cuenta las horas consumidas para ejecución de pruebas internas en su ciclo inicial (no incluye segundos ciclos, regresiones o Smoke Test), con esto, identificar el consumo real ejecutado según lo planeado en esta primera etapa de un proceso de ejecución de pruebas, como resultado, el PE tiene un consumo **183 horas** y PC **233 horas** correspondientemente, a simple vista no representa una diferencia importante, sin embargo, si se valora estos resultados en comparación con las horas consumidas en desarrollo, se observa una relación favorable para el PE, por lo que el tiempo dedicado para pruebas según el desarrollo es del **58,46%**, un valor aceptable bajo el criterio de la compañía quien informa que para este tipo de desarrollos, aunque solo es un punto de referencia, las pruebas se consideran óptimas cuando no se supere el 60% de las horas desarrolladas, situación diferente PC que cuenta con un **73,27%**.

Partiendo de los datos anteriores en fase de pruebas y desarrollo, agregando la relación de casos de uso (PC=23 CU y PE=49 CU) se procede al siguiente análisis:

Proyecto	Casos de uso	Tiempo Desarrollo	Tiempo Pruebas	Promedio horas Desarrollo X Caso de uso	Promedio de horas Pruebas X Caso de uso
Proyecto Control	23	318	233	13,8 horas / CU	10,13 horas / CU
Proyecto Experimento	49	313	183	6,38 horas / CU	3,73 horas / CU

Tabla 9 Relación promedio de horas de desarrollo y pruebas por caso de uso

Como se puede evidenciar en la tabla 9 hay una eficiencia en el desarrollador para el PE, pues cuenta con una relación favorable en la utilización del tiempo de desarrollo Vs los casos de uso a desarrollar, si bien los casos de uso del PC cuentan con más contenido, el PE representa una relación de **2,13 veces** la cantidad de casos de uso que los diseñados para el PC. Con lo anterior se puede interpretar que el desarrollador del PE comprendió de mejor manera la información descrita en la especificación de los requerimientos, optimizando el tiempo invertido para construir la solución.

Además del análisis anterior, se evalúa la relación que existe entre la tasa promedio que dura el desarrollador en construir la respectiva funcionalidad, basado en el detalle definido en el documento de caso de uso correspondiente, como se puede observar, la tasa promedio que tiene el PC supera el doble (2,16 veces) del tiempo promedio que se tomó para atender el desarrollo por caso de uso en el PE. Esta variación entre estos proyectos genera una desviación estándar de 5,24 la cual se muestra como una desviación importante entre estos datos de los proyectos. Por otro lado, el anterior resultado, guarda las mismas proporciones en la comparación que se presenta para el consumo de horas en fase de pruebas según la relación de caso de uso, pues la diferencia que existe en estas tasas promedio es de (10,13 – 3,73) 6,4 horas adicionales requeridas para probar un caso de uso en el PC de lo que toma hacer esto mismo en el PE, favoreciendo al PE, evidenciando una eficiencia en ejecución de pruebas partiendo de la documentación de los requerimientos, por lo que se podría interpretar que el Tester en el

ejercicio de diseño de casos de pruebas como la misma ejecución de las pruebas, asume de mejor manera la información suministrada en los documentos de especificación y casos de uso y en las socializaciones realizadas para comprender los requerimientos.

Con el análisis realizado en este punto, se interpreta que el PE cuenta con una tasa más efectiva (6,30 horas x CU) de tiempo invertido en desarrollo de casos de uso; reduciendo en términos de tiempo a un **46,23%** de lo que se realizó para el PC. En el caso de etapas de pruebas, también se evidencia una tasa más efectiva (3,73 horas por CU) en el PE para la ejecución de pruebas por caso de uso, reduciendo a un **36%** el tiempo necesario para ejecutar las pruebas relacionadas a un caso de uso, de lo que se tomó para las pruebas de un caso de uso en el PC.

Proyecto	Casos de uso	Casos de Prueba	Tasa de casos de prueba por un caso de uso
Proyecto Control	23	167	7,26 casos de pruebas por un caso de uso
Proyecto Experimento	49	415	8,46 casos de pruebas por un caso de uso

Tabla 10 Tasa promedio de cantidad de casos pruebas diseñados para un cubrir un caso de uso

En la tabla 10 se puede evidenciar la relación promedio de cantidad de casos de prueba por cada caso de uso, arrojando un resultado en el que por cada caso de uso se diseñan aproximadamente 8,46 casos de prueba, cerca de un caso de prueba adicional de lo que se refiere al PC, en este escenario se puede interpretar que para el PE se incrementó la cantidad de diseño de casos de prueba en un **16,52%**, posiblemente al encontrar un alcance claro y bien detallado que permita evidenciar posibles escenarios de pruebas a ejecutar.

5.2.4. Análisis de pruebas Vs Defectos encontrados

Este análisis aplica para las pruebas internas realizadas por el Tester asignado en cada equipo, cabe aclarar que este diseño de casos de pruebas es validado por el líder del equipo Testing como medida de control del proceso interno de la compañía desarrolladora de software. De lo anterior, para el proyecto PC se diseñaron 167 casos de pruebas y para el PE 415, en esta ocasión la diferencia que se presenta entre estos proyectos es alta, teniendo una desviación de 175 casos de prueba.

Adicional a esto, la cantidad de defectos por cada proyecto en relación de casos de prueba, se evidencia una dependencia importante para los resultados del PE, en el cual, mientras en el PE se evidencia un defecto por cada 9 casos de prueba, la tasa que existe en PC es de 1,36; evidenciando una tasa superior en el que se encuentra un número mayor de defectos es una cantidad menor de casos pruebas en PC. A continuación, se presenta tabla 11 de tasa de defectos por casos de pruebas según proyecto:

Proyecto	Casos de Prueba	Defectos Encontrados	Tasa de defectos por cantidad de casos de prueba
Proyecto Control	167	122	1,36 casos Prueba / Defecto
Proyecto	415	44	9,4 casos Prueba / Defecto

Proyecto	Casos de Prueba	Defectos Encontrados	Tasa de defectos por cantidad de casos de prueba
Experimento			

Tabla 11 Tasa de defectos por cantidad de casos de prueba

Siguiendo con el análisis, se estudia el tiempo invertido para solución de estos defectos y el tiempo consumido para efectuar las pruebas sobre los defectos reportados; generando la siguiente información:

Proyecto	Defectos Encontrados	Tiempo Consumido En solución de defectos	Tiempo consumido en ejecución de pruebas sobre defectos reportados	Promedio de tiempo consumido en solución un defecto	Promedio de tiempo consumido para validar un defecto
Proyecto Control	122	83	58	40 minutos / Defecto	28 minutos / Defecto
Proyecto Experimento	44	56	20	1 hora y 16 minutos / Defecto	27 minutos / Defecto

Tabla 12 Relación de tiempo promedio consumido para solución y pruebas de un defecto reportado.

Como se puede evidenciar en la tabla 12, para el PE aunque cuenta con menos casos reportados siendo un proyecto con un alcance con mayor número de requerimientos identificados y así mismo cantidad de casos de uso, presenta una tasa promedio donde el desarrollador tarda 1 hora 16 minutos(aprox) en solucionar un caso reportado como defecto, equivalente a 1,8 veces lo que dura el desarrollador para dar una solución en promedio a un defecto reportado en el PC. Así mismo, también se evidencia el tiempo promedio que tarda un Tester en probar nuevamente los defectos reportados entre los dos proyectos, en esta ocasión los resultados son similares, pues la diferencia que existe entre estos dos proyectos, según el promedio, al PC le toma un minuto más probar los defectos reportados y solucionados previamente, lo que no representa una desviación significativa en la comparación de los proyectos (28 Minutos para PC y 27 Minutos PE), sin embargo, este resultado indica que el comportamiento de los Tester partiendo de sus prácticas y conocimiento tienden a tener una media constante de la manera en que asumen las pruebas en segundos ciclos de pruebas, independientemente del proyecto.

5.2.5. Análisis de cantidad de Defectos encontrados en fase de pruebas funcionales del usuario

Como parte del análisis de datos, se descarta esta medida para fijar el resultado de la hipótesis, partiendo de que cada usuario (cliente) cuenta con métodos de pruebas totalmente diferentes y sus contextos de trabajo se basan en su conocimiento del negocio y no técnico, por lo que las pruebas, en ocasiones resultan ser sesgadas, básicas o poco rigurosas, generando un resultado no apropiado para la comparación de los proyectos.

5.2.6. Análisis de velocidad del Sprint y valor ganado

Para este proyecto, tanto para la medida de velocidad del Sprint y la de Valor ganado, se puede determinar que el equipo tuvo un rendimiento constante en los entre los dos proyectos, no

obstante, aunque se evidencia mejores resultados entre el proyecto Experimento que el proyecto Control, no hay una diferencia significativa, que pueda concluir que la implantación de la Guía aumento el desempeño del equipo durante la ejecución del Sprint, o que los compromisos de entregas, mejoraron notablemente frente a la planeación de cada Sprint., por lo que los resultados, son aproximados y la variación en este caso, no es considerable y puede variar por el simple hecho, que son personas diferentes quienes intervinieron en la fase de construcción como el de las pruebas entre estos proyectos.

Proyecto	Velocidad Promedio por Sprint	Porcentaje de Valor Ganado
Proyecto Control	92,0	91%
Proyecto Experimento	91,7	93%

Tabla 13 Resultados de velocidad promedio y valor ganado.

5.2.7. Análisis de consumo de horas en todas las fases del proyecto

La siguiente grafica (Ilustración 6), presenta los datos acumulados de consumo de horas, con el objetivo de evidenciar el comportamiento entre los proyectos independientemente del consumo de horas efectuado en cada fase. Como se observa, el crecimiento es incremental y simétricos señalando una conducta esperada, respondiendo a la intención de comparar dos proyectos con similares condiciones.

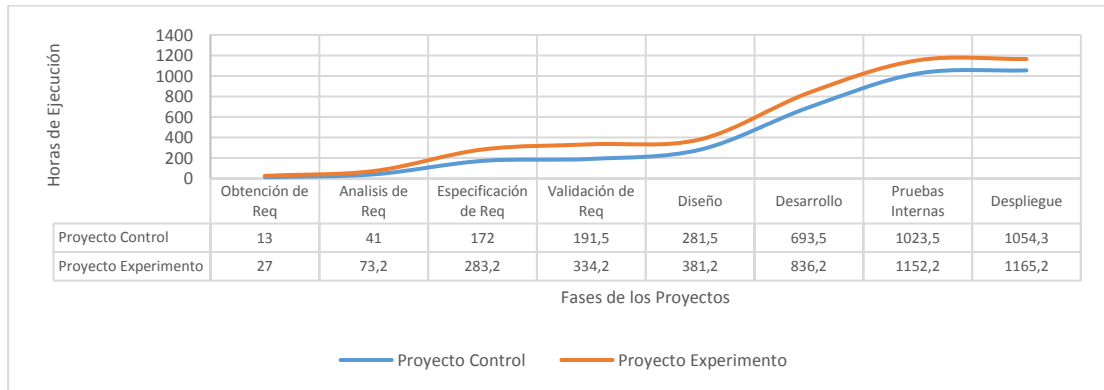


Ilustración 6 Comportamiento incremental del consumo de horas por etapa según proyecto.

No obstante, a partir de esta misma gráfica, se pronuncia una curva acentuando el consumo de horas entre las fases de análisis y especificación, así mismo, proporcionalmente se reduce la relación de consumo de horas en fases de pruebas y desarrollo generando una correlación positiva entre estas fases. Lo anterior permite interpretar que, el esfuerzo invertido en fases iniciales es compensado por los resultados evidenciados en las últimas fases del proceso de desarrollo de software.

Por otro lado, realizando un “zoom” al comportamiento de consumo de horas dentro del proceso de ingeniería de requerimientos, a continuación, se relaciona una comparación de las fases del proceso de Ingeniería de Requerimientos para los dos proyectos:

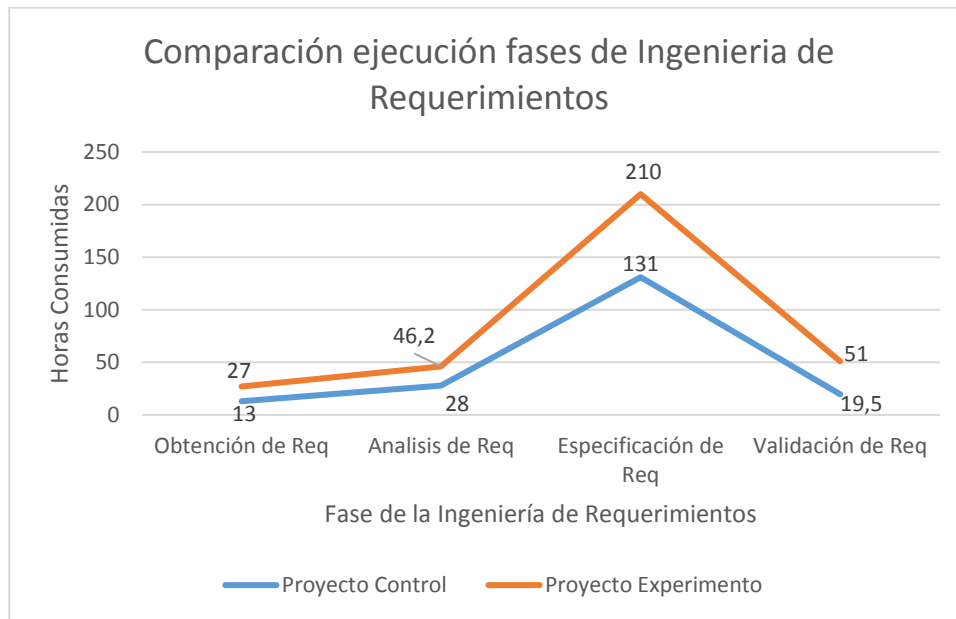


Ilustración 7 Comparación de tiempos consumidos en fases de ingeniería de requerimientos

Como se observa en la gráfica anterior, aunque los dos proyectos cuentan con un comportamiento similar, se evidencia que el PE se incrementa el tiempo invertido por cada fase, en especial, entre las fases de Análisis de Requerimientos y la Fase de Especificación, retornando en el tiempo la disminución en la brecha entre los proyectos en fase de validación de requerimientos.

5.2.8. Análisis Proyección de horas proyecto Experimento







Dentro del análisis y como resultado de la comparación, se procede a realizar una proyección de consumo de horas partiendo de los datos identificados en los puntos anteriores, lo que se pretende, es simular una estimación del incremento de tiempo en el caso en que el proyecto experimento no se hubiese implantado la Guía, esta proyección se realiza con las tasas y promedios resultantes del PC, el cual es punto de referencia para el diseño de investigación Quasi – Experimental de este proyecto.

A continuación, se presenta la tabla 12, que para mejor interpretación se explica cada columna:

- **Indicador:** Resultado de tasas o promedios que surgieron del análisis en los resultados de la comparación de los datos recolectados por el proyecto Experimento y el proyecto Control.
- **Tasa resultante en el proyecto Control:** Tasa que resulta de los cálculos aplicados por cada indicador (tasa /promedio) para el proyecto control, ejemplo, para el primer indicador, se puede interpretar que 0,29 Horas /req es el promedio que toma entender un requerimiento antes de proceder con la especificación.
- **Resultados actuales proyecto experimento:** Son los resultados que se presenta de la operación aritmética según el indicador por su tasa o promedio correspondiente, por ejemplo, para el primero indicador se tiene en cuenta la cantidad de requerimientos

definidos para el proyecto control (45 requerimientos) multiplicado por el promedio de horas invertido para entendimiento de un requerimiento, esto resulta (21,6 horas).

- Resultado de proyección proyecto experimento:** Esta columna cuenta con los datos de la operación aritmética entre el indicador (tasa o promedio) que surgió del proyecto control (punto de referencia) frente al resultado de cada variable usada para esta comparación, ejemplo, el primero indicador, está multiplicando la tasa que resultado del proyecto control (0,29 horas por requerimiento) por la cantidad de requerimientos identificados para el proyecto experimento (45 requerimientos), generando la proyección por cada indicador, que en este caso es de 13,05 horas lo que hubiese tomado entender todos los requerimientos del proyecto Experimento.
- Diferencia entre resultado actual Vs Proyección (proyecto Experimento):** Esta columna representa el resultado del indicador el cual se calcula de la resta entre el resultado de la columna “Resultados de proyección proyecto Experimento” con la columna “Resultados Actuales proyecto Experimento”, generando una diferencia; que de ser negativo se entiende como un resultado que no favorece el uso de la guía al proyecto experimento según proyección, de ser positivo el resultado, demuestra que el uso de la guía probablemente disminuye la tasa o promedio que representa alguna actividad o fase del proceso de ingeniería de Software, pues se puede interpretar que, no haber implantado la guía se supone un consumo de horas superior al que se supone el uso de esta variable en un proyecto.

Indicador	Tasa resultante en el Proyecto Control	Resultados Actuales proyecto Experimento	Resultados de Proyección Proyecto Experimento	Diferencia entre resultado actual Vs Proyección (proyecto Experimento)
Promedio “HIPER” por requerimiento	0,29 horas / REQ	$(0,48 * 45) = 21,6$ Horas	$(0,29 * 45) = 13,05$ Horas	 $13,05 - 21,6 = -8,1$
Tasa Promedio de casos de uso por Req.	1,15 CU / Req	$(1,08 * 45) = 48,6$ Casos de uso	$(1,15 * 45) = 51,75$ casos de uso	 $51,75 - 48,6 = 3,15$
Promedio horas Desarrollo por Caso de uso	13,8 horas / CU	$(6,38 * 49) = 312,8$ Horas	$(13,8 * 51,75) = 714,15$ horas	 $714,1 - 312,8 = 401,3$
Promedio de horas Pruebas por Caso de uso	10,13 horas / CU	$(3,73 * 49) = 182,77$ Horas	$(10,13 * 51,75) = 524,22$ horas	 $524,2 - 182,7 = 341,4$
Promedio de casos de prueba por casos de uso	7,26 caso de prueba / CU	$(8,46 * 49) = 414,5$ CU	$(7,26 * 51,75) = 375,7$ Casos de prueba.	 $375,7 - 414,5 = -38,8$
Tasa de defectos por cantidad de casos de prueba	1,36 casos Prueba / Defecto	$(415/9,4) = 44,1$ Defectos	$(375,7/1,36) = 276,25$ Defectos	 $276,25 - 44,1 = 232,4$



Indicador	Tasa resultante en el Proyecto Control	Resultados Actuales proyecto Experimento	Resultados de Proyección Proyecto Experimento	Diferencia entre resultado actual Vs Proyección (proyecto Experimento)
Promedio de tiempo consumido para solución por defecto	0,66 horas / Defecto	$(1,27 * 44) = 55,88$ Horas	$(276,25 * 0,66) = 182,49$ Horas	 $182,4 - 55,8 = 126,2$
Promedio de tiempo consumido para validar defectos reportados	0,46 horas / Defecto	$(0,45 * 44) = 19,8$ horas	$(0,46 * 276,25) = 127,07$ Horas	 $127,0 - 19,8 = 107,2$

Tabla 14 Indicadores sobre proyección de ejecución proyecto Experimento

Como resultado en el análisis, en la tabla 14 se observa un indicador positivo para solo dos resultados comparativos (tasa o promedio), el primero es “horas invertida para entendimiento de los requerimientos (HIPER)” donde el análisis conduce a concluir que, al no hacer uso de las sugerencias de la guía; no se genere horas adicionales para entender los requerimientos como lo propone la guía, el cual posiblemente genere un efecto en cadena afectando los demás indicadores que se presentan en la tabla. El segundo indicador que favorece al proceso actual de Ingeniera de Requerimientos de esta compañía, es el de “Promedio de casos de prueba por casos de uso”, en este escenario, entendiendo que la documentación de especificación es el insumo principal para diseñar los casos de pruebas, se puede interpretar que al tener un alcance limitado origina un diseño de casos de prueba de la misma manera, por lo tanto en este caso, estos dos soportes (casos de pruebas y casos de uso) son directamente proporcionales dejando una covarianza positiva en su relación.

Claramente este análisis parte de un supuesto, sin embargo los resultados que se presentan en esta comparación, permiten concluir que el proyecto control toma dos veces más sobre las horas consumidas del resultado actual, es decir, se proyecta un incremento de 976,1 horas adicionales al PE en relación a las 592,85 horas consumidas implantando la Guía durante la ejecución del proyecto; teniendo en cuenta únicamente los promedios seleccionados para este análisis, equivalente a un incremento del 163% en consumo de horas de la manera en que se ejecuta el proceso tradicional.

5.2.9. Análisis sobre la aceptación subjetiva de la Guía

A partir de la validación a través de la encuesta de aceptación 3.6, realizada al equipo que ha participado en los proyectos donde está relacionada la Guía, se ejecuta en tres escenarios diferentes: por primera vez después de finalizar la inducción a la guía, la siguiente vez se aplica al

finalizar la ejecución del proyecto piloto y por último, al finalizar el proyecto experimento, de lo anterior se generan los siguientes resultados consolidados²⁵:

Facilidad	Inducción	Piloto	Experimento	Refinamiento	Total Promedio
Gerentes	84,29%	62,86%	82,86%	85,71%	78,93%
Analistas	87,14%	60,00%	77,14%	84,29%	77,14%

Tabla 15 Porcentaje de aceptación según percepción de “Facilidad” por rol y etapa de proyecto

Utilidad	Inducción	Piloto	Experimento	Refinamiento	Total Promedio
Gerentes	78,89%	62,22%	88,89%	97,78%	81,94%
Analistas	86,67%	53,33%	71,11%	86,67%	74,44%

Tabla 16 Porcentaje de aceptación según percepción de “Utilidad” por rol y etapa de proyecto.

Como se puede observar, si se analiza la tabla 14 y tabla 15, desde el inicio (inducción) hasta el final (refinamiento), la aceptación de la Guía, desde el punto de vista Facilidad y Utilidad fue creciendo a medida que se iba implementando, esto se debe a la experiencia que lograba el equipo a medida que se ejecutaban los proyectos, sin embargo, también se evidencia, en la implantación de la Guía en el proyecto piloto, los resultados no fueron los mejores, debido a la frustración de no poder entender cómo aplicar algunos estados de los alfas, además de los inconvenientes para el uso del Kanban como lo sugiere la Guía.

También, a partir de las tablas 14 y 15, se evidencia que el grupo de Analista funcionales que estuvieron expuestos al uso de la guía, presentan una valoración inferior a la que presentan los gerentes, la interpretación a este suceso, se puede explicar, que, para los gerentes de proyecto el estado, la salud y la visibilidad del proyecto, son características primordiales en la gestión de proyectos y así mismo, le dan una relevancia mayor a la del Analista Funcional, quienes consideran que de estos estados, algunos son intrínsecos en su actuar en el rol como Analistas funcionales y no le encuentran lo suficientemente importante.

En conclusión, se evidencia que la percepción de uso y la facilidad son favorables para los usuarios directos, como también se evidencia, que la percepción en la facilidad, tiene un porcentaje de aceptación más alto que el de utilidad, por lo que se puede interpretar, que esto se debe, a situaciones de complejidad que se manifestaron en el uso de modelos, en aplicar KANBAN y en la interpretación de algunas actividades de los estados al no contar con ejemplo

²⁵ Para observar el detalle de los resultados por cada encuesta, remitirse al documento anexo: Resultados Consolidados Encuesta Aceptación.xlsx

suficientes o un con marco metodológico claro, en que se pueda ver con precisión cada uno de estos elemento para el uso completo de la guía.

5.2.10. Análisis de comparación de resultados por puntos de casos de uso UCP

Respecto el análisis comparativo por puntos de casos de uso, en la tabla 17, se presenta los factores de evaluación para generar los puntos por casos de uso, se evidencia que en el promedio de puntos asignados por instrucciones por casos de uso (UUCW) es mayor para el PC que para el PE, como se explica en el análisis del punto 5.2.2, se debe a la especificación que implica detallar las instrucciones que deberá realizar el sistema, por lo tanto entre más instrucciones mayor es el tiempo consumido en la especificación documental de los requerimientos asociados.

Proyecto	UUCW- Peso de los Casos de Uso sin Ajustar	UAW- Pesos de Actores Sin ajustar	TFC - Factor de Complejidad Técnico	ECF -Factor de Complejidad Ambiental	UCP- Puntos Casos de Uso
Promedio Proyecto Experimento	44,082	2,551	2,672	0,897	114,927
Promedio Proyecto Control	50,217	2,739	2,691	0,896	138,201

Tabla 17 Comparación de proyectos experimento y control por factores de complejidad para cálculo de puntos por casos de uso.

Respecto al factor de peso por actores sin ajustar por sus siglas en inglés “AUW”, el PC tiene un promedio mayor de participación de actores por casos de uso, como sucede también con el promedio resultado del factor UUCW (pesos de casos de uso sin ajustar), por lo que permite interpretar que para el PC en un mismo caso de uso, se agruparon más acciones que lo que se hizo para el PE, posiblemente esto se deba a que, las pautas de la Guía propone tareas que permite forzar al analista a delimitar funcionalmente los casos de uso, haciéndolos específicos y más sencillos de interpretar para las futuras etapas del proceso de desarrollo de software.

En cuanto a la comparación de factor de Complejidad Técnica (TFC) como Factor de Complejidad Ambiental (ECF), presentan en su promedio resultados aproximados, confirmando la similitud de los proyectos respecto a las características tanto en su entorno de ejecución (personas, herramientas, conocimientos, habilidades, entre otras) como la complejidad definida a los aspectos técnicos que propone los puntos por casos de uso.

Dentro de los factores técnicos resalta la diferencia de los aspectos “Procesamiento Interno” y “Código Reutilizable”; dando una puntuación más alta significativamente al PC.

En cuanto a los factores ambientales, los aspectos diferenciales son “Capacidad de Analista Líder” y “Motivación”, favoreciendo al PC, sin embargo, un factor que contrarresta el resultado general de ECF para este mismo proyecto, está en el porcentaje de asignación de “Personal tiempo Parcial” (ver tabla 18) el cual, para el PC es menor que PE con una diferencia de 25% aprox en la asignación del Analista Funcional como del desarrollo.

Factor de Complejidad Técnica	Promedio Proyecto Experimento	Promedio Proyecto Control	Factor de Complejidad
El código debe ser reutilizable	2,061	3,043	TCF
Personal tiempo parcial	-3,000	-2,000	EFC
Capacidad del analista líder	1,429	2,000	EFC
Motivación	3,469	2,565	EFC

Tabla 18 Aspectos de factores de complejidad relevantes por su diferencia entre los proyectos.

Para conocer el detalle e impacto que tiene cada factor por cada proyecto ver anexo 27 (PlantillaEstimacion_PuntosCasosdeUso_Experimento.xlsx).

PROPUESTA DE REFINAMIENTO Y CONCLUSIONES

En este último segmento, se plantea la propuesta de refinamiento, partiendo del conocimiento adquirido en procesos de ingeniería de requerimientos integrando procesos ágiles y aspectos sociales en entornos reales. También incluye las conclusiones generadas a partir de lecciones aprendidas y la experiencia adquirida durante la ejecución de los proyectos que hicieron parte del Diseño Quasi experimental.

Propuesta de Refinamiento

Mediante la ejecución de los proyectos Piloto, Control y Experimento, permitieron al equipo de trabajo adquirir experiencias sobre el uso de guía metodológica, también se construyeron lecciones aprendidas y sugerencias para fortalecer la guía en su uso. Muchas de estas propuestas, están orientadas a robustecer el proceso de especificación y captura de los requerimientos, teniendo en cuenta que la guía propone soluciones basados en los modelos propuestos por *Joy Beatty* y *Anthony Chen*, en su libro “*Visual Models for Software Requirements*”[32]. Entre los modelos propuestos, se encuentran la categoría de modelo orientados a personas, que describen a los Stakeholders del sistema, sus procesos de negocio y sus objetivos. Entre los más reconocidos en la industria están los Casos de Uso, por lo que, en consenso con el equipo de trabajo, se consideró, que los Casos de Uso pueden aportar significativamente al proceso cerrando la brecha socio-técnica, teniendo en cuenta, que este es el primer acercamiento donde las necesidades empiezan a transformarse en requerimientos (funcionales o no funcionales) de sistema. Además, también se considera la incursión que ha tenido los casos de uso en los diferentes sectores por lo que es un medio aceptado abiertamente, siendo un medio común de comunicación, cuando se trata de la interacción de clientes con sus proveedores de Software. Cabe aclarar que, en la ejecución de los proyectos, la definición de los requerimientos se llevó a cabo

mediante casos de uso en su versión tradicional, Por lo anterior, a partir de las diferentes propuestas originadas por el equipo²⁶, se consolidó la idea de fortalecer la guía estructurando sus fases de ingeniería de requerimientos a través de los casos de uso, sin embargo, la disertación del equipo concluía que, la inclusión de los casos de uso a la Guía deberá alinearse en su filosofía de principios ágiles y evitar conseguir el nivel del formalismo que impera en las metodologías tradicionales.

En la compañía desarrolladora de software donde se ejecutaron los proyectos Piloto, Control y Experimento, pretende incluir la guía como reservorio metodológico para la gestión de requerimientos. Además, considera, que el uso de la guía de manera frecuente hará que los usuarios de esta misma, adopten sus prácticas inconscientemente a través de su aplicación reiterativa, pues el esquema funcional con que está diseñada, parte de una estructura sistemática, señalando una secuencia lógica lo que se supone, fortalecerá la base de conocimiento tácito de cada individuo de quien la usa, favoreciendo al proceso de desarrollo de software en sus etapas tempranas.

A partir de lo anterior se propone el siguiente cambio en la guía metodología objeto de este proyecto de grado:

6.1.1. Aplicación de Casos de Uso 2.0

Para esta compañía los casos de uso siguen siendo un factor determinante para especificar las funcionalidades solicitadas por los clientes, de tal manera que los requerimientos se descubran, compartan y entiendan en términos comunes, en la manera de “como” una necesidad se resuelve a nivel de aplicación de software. Estos modelos se aplican sin importar si la necesidad conduce a un desarrollo grande o pequeño, pues este modelo permite de manera concreta lo que tendrá y no tendrá la solución. Además de lo anterior, los casos de uso resumen el alcance y permite evidenciar el progreso, pues hace parte del repositorio que sirve como punto de referencia para la validación y pruebas de los requerimientos; y de la prioridad señalada en el momento del desarrollo de Software.

Cabe aclarar que los casos de uso si bien es cierto son empleados por metodologías tradicionales, también pueden aportar dentro de la ejecución de metodologías ágiles y por supuesto en las metodologías híbridas. Sin embargo, no son los casos de uso tradicionales que ha ofrecido UML a través de la historia de la Ingeniería de Software, en este caso, se hace referencia a la práctica de Casos de Uso 2.0[76] basadas en lineamientos escalables y ágiles propuestos principalmente por *Ivar Jacobson* quien también hace parte del grupo creador de la esencia del Kernel de SEMAT, esta propuesta se fundamenta en el libro “Use Case 2.0 “[76] del cual se tomará como referencia varios apartados del mismo para explicar el contexto en que se podrá incorporar en la Guía.

Inicialmente se cita la contextualización y definición de Casos de Uso 2.0:

²⁶ Algunas de las propuestas realizadas por los integrantes del equipo se llevan como propuestas de trabajo futuro, ver punto XX trabajos futuros.

“Los Casos de Uso 2.0 existen como una práctica probada y bien definida, compatible con muchas otras prácticas de desarrollo de software tales como Integración Continua, Arquitectura Intencional y Desarrollo Dirigido por Pruebas. También, trabaja con todas las prácticas populares de administración. En particular, es tan liviano y flexible como para apoyar a equipos que trabajan de una manera ágil, al mismo tiempo, tiene la plenitud y el rigor requerido para soportar los equipos que se requieren para trabajar de una manera más formal o en un entorno en cascada.

Los Casos de Uso 2.0 son:

Livianos: tanto en su definición como en su aplicación

Escalables: ajustables para equipos y sistemas de todos los tamaños

Versátiles: ajustables para todos los tipos de sistemas y enfoques de desarrollo

Fáciles de usar: los modelos de casos de uso pueden ponerse rápidamente en su lugar y se pueden crear porciones para cubrir las necesidades de los equipos.”

La propuesta de Casos de Uso 2.0 se construyó partiendo de experiencias obtenidas recopiladas en más de 20 años desde que se aplicaron los Casos de Uso como lenguaje modelado Unificado UML [77]. La estrategia de los casos de uso tiene un alcance mucho más amplio que solo la captura de requisitos, lo que significa que los Casos de Uso 2.0 apoyan el análisis, diseño, planeación, estimación, seguimiento y prueba de sistemas. No establecen como se debería planear o manejar el trabajo de desarrollo o como se debería diseñar, desarrollar o probar el sistema, sin embargo, proporcionan una estructura para la adopción exitosa de sus prácticas seleccionadas de administración y desarrollo.

Como propuesta de mejora a la Guía, se propone la integración de algunos componentes de Casos de Uso 2.0 teniendo como referencia que *“Es posible implementar los Casos de Uso 2.0 en parte, los resultados serían indeterminados y la práctica que se emplee no sería la de los Casos de Uso 2.0”*. Esta integración es posible partiendo de una misma filosofía con que se fundamenta SEMAT, teniendo en cuenta que al ser escalable supone aplicabilidad para los diferentes entornos metodológicos de desarrollo de Software y así mismo se afirma en el documento de referencia Uses Cases 2.0:

“Los Casos de Uso 2.0 trabajan con todos los enfoques de desarrollo de software populares como:

- *Enfoques iterativos dirigidos por listas de pendientes, como Scrum, EssUP y OpenUP*
- *Enfoques basados en flujo de una pieza, como Kanban*
- *Enfoques todo en uno, como la Cascada tradicional”.*

Lo anterior cumple como complemento al objetivo de este proyecto de grado en “Enriquecer la Guía” ampliando su alcance de aplicación más allá de las metodologías ágiles, además, partiendo del sustento anterior, se procede a exponer la propuesta de integración de componentes que ofrece Casos de Uso 2.0 para enriquecer la Guía metodología objeto de este proyecto de grado.

Los Casos de Uso 2.0 cuentan con los “*Productos de trabajo*” que se puede entender como un homólogo de “*las cosas con las que siempre trabajamos*” de SEMAT, también se encuentra la versión de “*las cosas siempre hacemos*” traducido en Casos de Uso 2.0 como las “*Cosas por hacer*”. La esencia que se propone al integrar los Casos de Uso 2.0 con la Guía, se basa en la combinación de “*Productos de trabajo*” y “*Cosas por hacer*”, por esto, se realiza una breve descripción²⁷ de los cinco “*productos de trabajo*”, estos son:

- **Información de soporte:** El propósito de la información de soporte es capturar términos importantes que se emplean para describir el sistema y las reglas de negocio que no caben dentro del modelo de casos de uso.
- **Casos de Prueba:** El propósito de un caso de prueba es proporcionar una definición clara de los que significa completar una porción de los requisitos. Un caso de prueba define un conjunto de entradas de prueba y los resultados esperados, con el propósito de evaluar si un sistema trabaja correctamente o no.
- **Modelo de Casos de Uso:** El propósito de un modelo de casos de uso es capturar todas las formas útiles de emplear un sistema, en un formato accesible que captura los requisitos de un sistema y se puede usar para conducir su desarrollo y pruebas.
- **Narrativa de Casos de Uso:** El propósito de la narrativa de un caso de uso es contar la historia de cómo el sistema y sus actores trabajan en conjunto para alcanzar una meta específica.
- **Realización de Casos de Uso:** El propósito de la realización de un caso de uso es mostrar como los elementos del sistema, tales como componentes, programas, procedimientos almacenados, archivos de configuración y tablas de la base de datos, colaboran en conjunto para ejecutar un caso de uso.

Cada uno de los anteriores productos ofrece un nivel de detalle que va desde el nivel más liviano hasta un nivel completamente estructurado y documentado, la cantidad de detalle en el producto de trabajo se aumenta a medida que se baja en las columnas, ampliando y expandiendo su contenido.



Ilustración 8 Niveles de detalle de los productos de trabajo - Tomado de [76]

²⁷ Para mayor información consultar el libro *Uses Cases 2.0*[76]

Como recomendación es empezar siempre lo más liviano posible detallando lo estrictamente necesario, sin embargo, la experiencia del equipo juega un papel importante en el nivel de detalle con que se quiere la información de Casos de Uso, entre menos experiencia del equipo, más nivel de detalle se quiere en la definición de los diferentes productos de trabajo.

Cuando se habla de “Cosas por Hacer” Significa que el trabajo se divide en un número de actividades esenciales que se deben adelantar, pensando que los casos de uso están orientados a proporcionar un valor real al equipo. En la Ilustración 08, se representa en la manera en que se agrupan las actividades que se usan para descubrir, ordenar y verificar los requisitos, también las actividades que se usan para modelar, implementar y probar el sistema; cabe aclarar que, para efectos de integrar los casos de uso 2.0 con la Guía, las actividades homologas en que se ejecutan los casos de uso y que se hacen punto de encuentro con el proceso de ingeniería de requerimientos ofrecidos por la guía, solo se tendrán en cuenta las actividades de descubrir, ordenar y verificar los requerimientos.



Ilustración 9 Cosas por hacer - Tomado de [76]

Con el contexto anterior, a continuación, se presenta la propuesta de integración de Casos de Uso 2.0 para la guía en sus diferentes etapas del proceso de ingeniería de requerimientos.

Obtención de requerimientos: Como lo describe la guía, esta etapa se hace lo concerniente a todo el levantamiento de información relevante para el desarrollo del nuevo producto como solución de software, para esta fase, el aporte de Casos de uso 2.0, está relacionado a las actividades “por hacer” en la que se define los Actores y Casos de uso, para suplir las siguientes actividades:

- Acordar las metas del sistema
- Acordar el comportamiento del nuevo sistema
- Definir el alcance de las versiones del sistema
- Acordar el valor que el sistema proporciona

- Identificar las formas para emplear y probar el sistema

Remitiéndose a la Guía, la actividad correspondiente al cumplimiento de identificar los actores y modelos de casos de uso, está relacionada al cumplimiento de la actividad “Se han identificado los usuarios que usarán el sistema”, del estado “Concebido” del alfa “**Requirements / Requerimientos**” la cual está directamente asociada a la definición de los actores que intervienen en cada caso de uso. La relación que existe entre los actores y las diferentes acciones funcionales de lo que se espera del sistema, se cumple con la actividad “Han sido establecidas las necesidades de los usuarios que generan la oportunidad” y “Ha sido confirmado que es necesaria una solución basada en software” del estado “Solución Necesaria” del alfa “**Opportunity / Oportunidad**”, a continuación, se presenta la tabla 17 para una mejor explicación:

Alfa	Estado	Actividad de estado	Actividad Caso de Uso 2.0
Requerimientos	Concebido	Se han identificado los usuarios que usarán el sistema	Definir los actores
Oportunidad	Solución Necesaria	<ul style="list-style-type: none"> • Han sido establecidas las necesidades de los usuarios que generan la oportunidad • Ha sido confirmado que es necesaria una solución basada en software 	Definir el modelo de caso de uso

Tabla 19 Relación de actividades de la Guía en fase de Obtención de Requerimientos frente a actividades de Caso de Uso 2.0

Aplicación de Casos de Uso 2.0 en la Ejecución:

La manera que se plantea integrar la Guía con los Casos de Uso 2.0 en esta fase, en términos de ejecución, es a través del cumplimiento de los siguientes “Productos de Trabajo”, se sugiere anteceder en el orden que se presenta en la tabla 17, los cuales se exponen como complementarios a las instrucciones dadas en el Ítem de ejecución de la fase de “Obtención de Requerimientos” de la Guía, para esta fase el nivel de detalle que se sugiere es entre “bosquejo” y “Meramente lo esencial”, a continuación se presenta tabla 18 en el que se relaciona el producto de trabajo y el nivel de detalle que se espera:

Producto de Trabajo	Nivel de Detalle Propuesto	Descripción
Modelo de Casos de Uso	Con Valor Establecido	El detalle requerido debe estar relacionado en buscar crear la primera versión de lo que se espera de un modelo completo de casos de uso, donde la prioridad se concentra en identificar los actores y casos de uso más importantes (los primarios), que son los que dan valor al sistema.
Narrativa de Casos de Uso	Brevemente descrita	En fase de obtención de Requerimientos solo es necesario contar con el nivel más liviano partiendo del modelo de casos de uso, donde solo se captura la meta del caso de uso y el actor que lo inicia.

Producto de Trabajo	Nivel de Detalle Propuesto	Descripción
Información de Soporte	Iniciada	Es solo un resumen de los términos más obvios y de las áreas a tratar, solo se identifica las reglas de negocio más relevantes y las restricciones que desde el principio el cliente ha manifestado, se adicionará mayor detalle en las siguientes fases de Ingeniería de Requerimientos.
Casos de Prueba	Con ideas de pruebas formuladas	El nivel más liviano de detalle; solo captura la idea inicial que informará el caso de prueba, se propone en el momento que se está obteniendo los requerimientos; el Analista Funcional, junto al cliente logren definir la manera en que se pueda probar esa funcionalidad, el nivel de explicación se espera “grosso modo”.
Realización de Casos de Uso	N/A	No Aplica para esta fase.

Tabla 20 Nivel de detalle por cada producto de trabajo en relación a fase de Obtención de Requerimientos.

Para esta fase, el principal objetivo, se basa en dibujar diagramas que están constituidos de un conjunto de actores, acciones (casos de uso) y la ilustración de relaciones entre estos mismos elementos, a partir de esto, la descripción narrativa es lo suficientemente básica para entender quienes activan su ejecución y cuál es su propósito, además de definir las condiciones y restricciones básicas. Los productos de trabajo se pueden capturar de muchas formas distintas que incluyen una Wiki, un tablero, diapositivas de PowerPoint, un documento de MS Word o una herramienta de modelado, ver ejemplo de modelo casos de uso Ilustración 10.

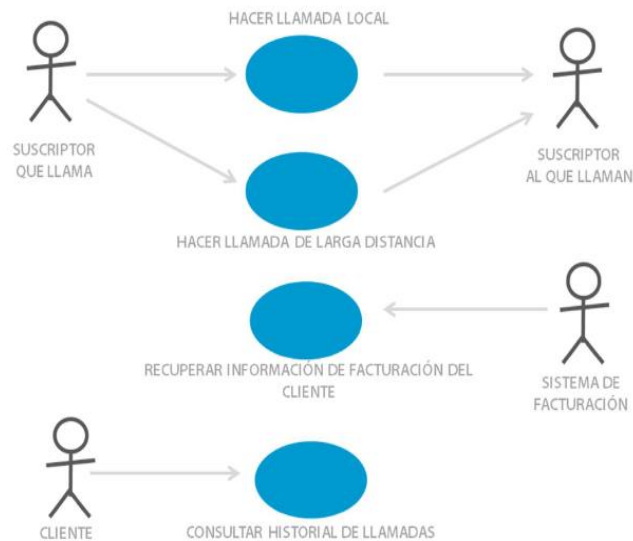


Ilustración 10 Ejemplo de un diagrama de caso de uso - Tomado de [76]

Análisis de requerimientos: En esta fase, toda la información recopilada en la fase anterior empieza a relacionarse para detectar los límites e inconsistencias lógicas que requieran de más detalle o simplemente se descarte por no contar con una viabilidad en su funcionamiento o alcance, para esto, lo que propone Casos de Uso 2.0, a partir de los productos trabajados en la fase de obtención de requerimientos, se procede a explotar los diagramas identificados en porciones posiblemente independientes, esta división de casos de uso se realiza con el siguiente objetivo:

- Crear elementos de tamaño adecuado con los cuales el equipo trabaje
- Estar acorde con el tiempo y el presupuesto disponible
- Entregar el más alto valor a los usuarios y a los demás interesados
- Demostrar el progreso crítico del proyecto o el entendimiento de las necesidades

Integrando la Guía con los Casos de Uso 2.0, mediante las actividades relacionadas a las alfas propuestos por la Guía, esta relación se ve representada en los alfas **Requirements / Requerimientos** en su estado “Delimitado” y en el alfa **“Opportunity / Oportunidad”** en su estado “Valor Establecido”, bajo el cumplimiento de las siguientes actividades:

Alfa	Estado	Actividad de estado	Actividad Caso de Uso 2.0
Requerimientos	Delimitado	<ul style="list-style-type: none"> • Los interesados tienen un entendimiento compartido del alcance de la solución propuesta. • Se ha acordado la forma en que los requerimientos serán descritos. 	Dividir los casos de uso
Oportunidad	Valor Establecido	Los resultados deseados y requeridos de la solución, están claros y cuantificados.	

Tabla 21 Relación de actividades de la Guía en fase de Análisis de Requerimientos frente a actividades de Caso de Uso 2.0

Aplicación de Casos de Uso 2.0 en la Ejecución:

Se sugiere que en la ejecución de los productos de trabajo se realicen en el orden que se proponen en la tabla 19 para mayor entendimiento y fluidez. Por otro lado, Aún el caso de uso más simple cubre múltiples historias, por esta razón es necesario dividir los casos de uso para seleccionar las historias a implementar, la división se debería hacer con los interesados para asegurarnos de que todas las porciones creadas sean de valor y valga la pena implementarlas. En esta fase no es necesario identificar todas las porciones de casos de uso de una sola vez, solo defina las porciones que se crea suficientes.

Producto de Trabajo	Nivel de Detalle Propuesto	Descripción
Modelo de Casos de Uso	Con límites establecidos	Los actores y casos de uso primarios capturan los motivos principales por los cuales se construye el sistema y cómo los usuarios obtendrán valor del sistema. Quizás no proporcionan suficiente valor para configurar y ejecutar el sistema. En estos casos, los actores y casos de uso secundarios son necesarios

Producto de Trabajo	Nivel de Detalle Propuesto	Descripción
		para habilitar y soportar la operación efectiva del sistema.
Narrativa de Casos de Uso	Con resumen Básico	El caso de uso se debe resumir para entender su tamaño y complejidad. Este nivel de detalle también posibilita un manejo efectivo del alcance, puesto que el resumen permite que las distintas partes del caso de uso se prioricen unas con otras y, si es necesario, se destinen a distintas versiones.
Información de Soporte	<ul style="list-style-type: none"> • Simplemente Definida • Modelada e ilustrada 	Los dos niveles de detalle, se proponen como opciones según el criterio a definir por el equipo de trabajo, Opción 1, “Simplemente Definida” se incluye un resumen de los términos más obvios y de las áreas a tratar (definidos en la fase anterior). La segunda opción es “Modelada e Ilustrada” Se puede adicionar más detalle a la información de la opción 1 transformando las definiciones básicas en modelos que, precisamente, capturen las definiciones, sus atributos y sus relaciones y proporcionando ejemplos del mundo real para clarificar las porciones de casos de uso correctas. En este caso se puede relacionar el diagrama BPMN que ofrece la Guía.
Casos de Prueba	Con escenario seleccionado	Para esta fase, se debe presentar un escenario que el Tester pueda ejecutar. Este escenario se basa en el contexto donde se adelanta la narrativa del caso de uso. Se debe asegurar, que cada porción del caso de uso cuente con uno o más escenarios de prueba candidatos. El arte de crear casos de prueba efectivos es seleccionar el subconjunto correcto de escenarios de prueba potenciales para cumplir con la idea de prueba y definir claramente cuando la porción está hecha.
Realización de Casos de Uso	N/A	No Aplica para esta fase.

Tabla 22 Nivel de detalle por cada producto de trabajo en relación a fase de Análisis de Requerimientos

A continuación, se representa mediante la ilustración 11 un ejemplo de porciones definidos a partir de un caso de uso:



Ilustración 11 Ejemplo de porciones definidos a partir de un caso de uso primario - Tomado de [76]

Especificación de requerimientos: Partiendo de la Guía, durante esta etapa como parte de especificación esta la opción de documentar los requerimientos obtenidos y analizados anteriormente, se verifican si son aceptados por los interesados, además de revisar que sean coherentes, la especificación se aplica como un segundo filtro para detectar inconsistencias. En relación a los Casos de Uso 2.0, se incluye la actividad de “Cosas por hacer” denominada “preparar una porción de caso de uso”, para cumplir con las siguientes acciones:

- Preparar la porción para la implementación (Construcción de Software).
- Definir claramente lo que significa la construcción de Software en relación al éxito de la porción.
- Definir las características requeridas (por ejemplo, los requisitos no funcionales).
- Enfocar el desarrollo de software en las pruebas que se deben cumplir.

Para esta fase, la integración de la Guía con los Casos de Uso 2.0, se realiza mediante el cumplimiento de los siguientes estados de las alfas involucrados según la actividad explícita en la siguiente tabla:

Alfa	Estado	Actividad de estado	Actividad Caso de Uso 2.0
Interesados	De acuerdo	Los representantes de los interesados están de acuerdo sobre las expectativas mínimas para el siguiente desarrollo del nuevo sistema de software.	Preparar una porción de caso de uso
Requerimientos	Coherente	<ul style="list-style-type: none"> • Los requerimientos son capturados y compartidos con el equipo y los Stakeholders. • Los requerimientos comunican las características esenciales del sistema a ser liberado. • Los escenarios de uso más importantes para el sistema pueden ser explicados. 	

Alfa	Estado	Actividad de estado	Actividad Caso de Uso 2.0
		<ul style="list-style-type: none"> El equipo comprende lo que será liberado, y está de acuerdo con liberarlo. 	

Tabla 23 Relación de actividades de la Guía en fase de Especificación de Requerimientos frente a actividades de Caso de Uso 2.0

Aplicación de Casos de Uso 2.0 en la Ejecución:

En cuanto a relación de los productos de trabajo y su nivel de detalle, se toman las mismas porciones definidas en la fase de análisis, y se procede a su especificación partiendo de los productos de trabajo identificados al nivel sugerido en la tabla 21, cabe aclarar que las opciones disponibles están sujetos al nivel de integración que exista en el equipo y la experiencia que lleve consigo para ejecutar este tipo de proyectos, por ejemplo, si el equipo es pequeño (3-5 personas) y trabaja de manera colaborativa con el cliente, entonces los resúmenes básicos proporcionarán una forma liviana de descubrir los requisitos por lo que el nivel de detalle “Ampliado” es suficiente. Si se está trabajando en un entorno rígido donde hay poco acceso a los expertos, entonces el nivel de detalle “Expandido” es apropiado para cubrir los vacíos en el conocimiento del equipo. No todos los casos de uso se deben especificar al mismo nivel de detalle; los casos de uso más importantes y con alto nivel de impacto en los riesgos tengan más detalle que los otros.

Producto de Trabajo	Nivel de Detalle Propuesto	Descripción
Modelo de Casos de Uso	Estructurado	Dependiendo del nivel de agilidad con que se trabaje puede llegar a un nivel de “Con Límites del Sistema Establecido” sin embargo, para cuando se trata de una especificación más formal se sugiere estructurar el modelo de casos de uso es la manera de lidiar con las redundancias. Para sistemas grandes y complejos, estructurar el modelo de casos de uso puede ayudar a entender y eliminar desperdicios y ayuda a encontrar elementos reusables.
Narrativa de Casos de Uso	<ul style="list-style-type: none"> Con resumen esencial Completamente descrita 	Se presenta dos niveles de detalle a considerar por el equipo de trabajo, la primera opción está orientada a definir las responsabilidades del sistema y sus actores, a diferencia al Resumen Básico definido en la fase anterior. La opción 2 Las narrativas de los casos de uso se pueden usar para proporcionar una especificación bastante detallada de los requisitos, haciéndolas evolucionar. El detalle extra puede ser necesario para cubrir la falta de experiencia dentro del equipo, la falta de acceso a los

Producto de Trabajo	Nivel de Detalle Propuesto	Descripción
		interesados o para comunicar efectivamente requisitos complejos.
Información de Soporte	<ul style="list-style-type: none"> Definida detalladamente 	Este nivel se sugiere, cuando es necesario aclarar la información proporcionando explicaciones más detalladas y material de apoyo como ejemplos detallados, fuentes y referencias cruzadas. La información de soporte es más complicada, con más precisión, con referencias cruzadas y usa técnicas de especificación formal, de no ser necesario se puede optar llegar hasta el nivel de “Modelada e ilustrada”.
Casos de Prueba ²⁸	<ul style="list-style-type: none"> Con variables identificadas Con variables fijas 	Se cuenta con dos niveles de detalle, Opción 1 “Con variables identificadas” Un caso de prueba toma algunas entradas, manipula los estados del sistema y produce algunos resultados. En este nivel de detalle, se definen explícitamente los rangos aceptables para las variables clave involucradas en el escenario. Opción 2 “Con variables Fijas” Se proporciona explícitamente valores específicos para todas las variables involucradas en el caso de prueba. Este nivel de detalle es apropiado para los casos de prueba manuales, ya que se cuenta con toda la información que un verificador necesita para ejecutar consistentemente el caso de prueba.
Realización de Casos de Uso	N/A	No Aplica para esta fase.

Tabla 24 Nivel de detalle por cada producto de trabajo en relación a fase de Especificación de Requerimientos.

Validación de Requerimientos: La Guía, describe, que, la validación se realiza para confirmar si los requerimientos definidos mediante las fases anteriores verdaderamente cumplen con la expectativa de los interesados, que para efectos de integración con los casos de Uso 2.0 se ve puede ver reflejado con “Analizar una porción de casos de uso” de las “Cosas por hacer”, por lo que antes de empezar a codificar, se debe analizar cada porción identificada con el objetivo de:

- Entender su impacto en los elementos del sistema que se emplearán para su desarrollo.

²⁸ Para los casos de pruebas existe un nivel más avanzado “Programada /Automatizada” esta opción queda sujeta a la experticia del equipo de Testing de la compañía que desee implementar, debido a que se requiere conocimiento en pruebas automatizadas.

- Definir las responsabilidades de los elementos que se afectan del sistema.
- Definir cómo interactúan los elementos del sistema para ejecutar el caso de uso.

A continuación, para mayor aclaración se expone en la tabla 23 la manera en que se asocia “Analizar una porción de casos de uso” respecto a las actividades de los estados de las alfas.

Alfa	Estado	Actividad de estado	Actividad Caso de Uso 2.0
Requerimientos	Aceptable	<ul style="list-style-type: none"> • Los interesados aceptan que los requerimientos describen una solución aceptable. • El valor es claro 	Dividir los casos de uso
Oportunidad	Viable	<ul style="list-style-type: none"> • El riesgo asociado a la solución es aceptable y manejable. • Las razones para el desarrollo de una solución basada en software son entendidas por todos los miembros del equipo. • La solución puede ser desarrollada y desplegada teniendo en cuenta las restricciones. 	

Tabla 25 Relación de actividades de la Guía en fase de validación de Requerimientos frente a actividades de Caso de Uso 2.0

Aplicación de Casos de Uso 2.0 en la Ejecución:

Dentro de la validación de requerimientos como lo expone la Guía, los Casos de Uso 2.0 complementan a lo expuesto por la Guía, de la manera en que cada porción de casos de uso, debe ser analizada a partir de la especificación el cual es la fase que recoge todo el trabajo de las fases anterior, medir su impacto, la claridad y alcance, validará muchas de las decisiones de diseño, por lo que se comprobará que nada se malentienda y proporcionará la trazabilidad para su uso posterior cuando se investigue el impacto de los defectos y cambios. Pues el paso inmediatamente siguiente es cuando el desarrollador procede a la codificación. A medida que el equipo entiende el sistema y su arquitectura, sus integrantes hallarán más y más fácil analizar las porciones.

En términos prácticos, la forma en que se propone analizar la porción de un caso de uso es hacer que el equipo se reúna para discutir como esta afectará a los distintos elementos del sistema. A medida que el equipo elabora diseños a partir de formas comunes, desde lo más básico como tablas simples, tableros de historias hasta construir diagramas de secuencias, diagramas de colaboración, diagramas de comunicación y diagramas de flujos de datos. A continuación se presenta la tabla 24 presentando el nivel de detalle sugerido.

Producto de Trabajo	Nivel de Detalle Propuesto	Descripción
Todos los demás productos de trabajo	N/A	Los productos de trabajo (Modelo de Casos de Uso, Narrativa de Casos de Uso, Información de Soporte y Casos de Prueba) solo se toman como referencia para construir la realización de casos de uso.

Producto de Trabajo	Nivel de Detalle Propuesto	Descripción
Realización de Casos de Uso	<ul style="list-style-type: none"> • Con responsabilidades asignadas • Con interacción definida 	<p>Para el nivel de detalle Opción 1 “Con Responsabilidades asignadas”, Para permitir que el equipo sea capaz de trabajar en paralelo o para soportar múltiples porciones, los desarrolladores necesitan entender las responsabilidades de los elementos individuales.</p> <p>Opción 2 “Con Interacción definida”: Para proporcionar una definición completa y sin ambigüedad de los cambios que requiere cada elemento del sistema involucrado en el caso de uso, la realización del caso de uso debe incluir detalles de todas las interfaces e interacciones involucradas en la ejecución del caso de uso.</p>

Tabla 26 Nivel de detalle por cada producto de trabajo en relación a fase de Validación de Requerimientos.

Cabe resaltar que el producto de trabajo “Realización de Casos de Uso” tiene como objetivo mostrar como los elementos del sistema, tales como componentes, programas, procedimientos almacenados, archivos de configuración y tablas de la base de datos, colaboran en conjunto para ejecutar un caso de uso, sin embargo, para efectos de un proceso de Ingeniería de Requerimientos, se restringe a evaluar escenarios donde las porciones de casos de uso interactúan, por ejemplo: la interfaz de usuario usan las realizaciones de los casos de uso (en forma de tableros de historias) para explorar el impacto de los casos de uso en la interfaz de usuario, Los arquitectos usan las realizaciones de los casos de uso para analizar los casos de uso significativos para la arquitectura y evaluar si la arquitectura cumple su propósito o no. Las realizaciones de los casos de uso se presentan en muchos formatos y son completamente dependiente de las prácticas de desarrollo del equipo.

Inspeccionar y adaptar los Casos de Uso

Como sucede en las metodologías ágiles, los Casos de Uso 2.0 ofrece también mediante “Cosas por hacer” la actividad “Inspeccionar y Adaptar los casos de uso”, en el que se afina y evalúa continuamente los casos de uso y sus porciones de los casos de uso, esto se realiza por los siguientes argumentos:

- Controlar los cambios
- Hacer seguimiento del progreso
- Ajustar el trabajo dentro del tiempo y presupuesto disponible
- Mantener el modelo de casos de uso al día
- Ajustar el tamaño de las porciones para aumentar el rendimiento

Partiendo de la literatura de Casos de Uso 2.0, a medida que el proyecto avanza, se apoya continuamente con el modelo de casos de uso, los casos de uso y las porciones de casos de uso, teniendo en cuenta que las prioridades cambian y se requieren cambios, esto puede tener un impacto en los casos de uso y en las porciones de casos de uso que se desarrollaron, así como los que están en proceso. Esta actividad conducirá al descubrimiento de nuevos casos de uso y a la refactorización de casos de uso y porciones de casos de uso existentes.

A partir del proceso de ingeniería de requerimientos definido en los puntos anteriores, se propone que cada porción de casos de uso cuente con un estado para generar trazabilidad y visibilidad para controlar los avances, ajustes, Estos estados son:

- **Alcance definido:** cuando se define el alcance y se clarifica la extensión de las historias cubiertas.
- **Preparada:** cuando se prepara la porción, ampliando la narrativa y los casos de prueba para definir claramente lo que significa la implementación exitosa de la porción.
- **Analizada:** cuando se analiza la porción de tal forma que se entiende su impacto en los componentes del sistema y las piezas afectadas están listas para codificación y pruebas del desarrollador (unitarias).
- **Implementada:** cuando se mejora el sistema de software para implementar la porción, que queda lista para pruebas.
- **Verificada:** finalmente, cuando se verifica la porción y queda lista para su inclusión en una versión.

Sin embargo, solo los tres primeros estados aplican para el proceso de ingeniería de requerimientos, los dos últimos aplican para porciones de casos de prueba que se encuentran en desarrollo y pruebas correspondientemente, a continuación, se relaciona el estado según fase del proceso de ingeniería de Requerimientos.

Fase de Proceso de Ingeniería de Requerimientos	Estado de porciones de Caso de uso
Obtención de Requerimientos	Alcance Definido
Análisis de Requerimientos	
Especificación de Requerimientos	Preparada
Validación de Requerimientos	Analizada

Tabla 27 Tabla de relación de estados porciones de casos de uso asociados a las fases del proceso de ingeniería de requerimientos.

Conclusiones

Una vez expuesto el análisis de la comparación de datos en el punto anterior, se procede a las conclusiones orientadas a los beneficios de la aplicación de la Guía que integra métodos formales de la ingeniería de requerimientos en metodologías ágiles, en este caso puntual, no solo para metodologías ágiles sino también metodologías híbridas en contextos reales de trabajo, en el sector de la industria del Software local.

A continuación, se presenta una tabla con los promedios y tasas²⁹ más representativas resultante de la comparación de los proyectos Control y Experimento, con lo que se soporta las conclusiones que se presentarán a continuación, cabe aclarar que las celdas resaltadas en amarillo son resultados no favorables para el proyecto Experimento y el color verde son las tasas favorables según resultados para el proyecto en el que se implanto la Guía:

Proyecto	HIPER por requerimiento	Tasa Promedio de casos de uso por Req.	Tasa de casos de prueba por un caso de uso	Promedio horas Desarrollo por Caso de uso	Promedio de horas Pruebas por Caso de uso	Tasa de defectos por cantidad de casos de prueba	Promedio de tiempo consumido en solución de un defecto	Promedio de tiempo consumido para validar un defecto
Proyecto Control	0,29 horas /REQ	1,15 CU / Req	7,26 casos de pruebas por un CU	13,8 horas / CU	10,13 horas / CU	1,36 casos Prueba / Defecto	40 minutos / Defecto	28 minutos / Defecto
Proyecto Experimento	0,48 horas /REQ	1,08 CU / Req	8,46 casos de pruebas por un CU	6,38 horas / CU	3,73 horas /CU	9,4 casos Prueba / Defecto	1 hora y 16 minutos / Defecto	27 minutos / Defecto

Tabla 28 Resultados de tasas o promedios más representativos del análisis comparativo entre proyecto control y proyecto experimento.

1. Como se puede observar del cuadro anterior, donde se representa un resumen de los resultados más relevantes del punto “*Análisis e interpretación de datos*”; las actividades ejecutadas relacionadas a un proceso de desarrollo de software, tiende a señalar mejores efectos para el proyecto experimento donde se implantó la Guía, no obstante, también es de resaltar que al implantar la Guía, propone un esfuerzo adicional para todas las fases del proceso de ingeniería de Requerimientos, en el que se manifiestan actividades complementarias en la consecución del cumplimiento de cada tarea de los estados de los alfas. A partir de esto, se genera un control que por el momento supone incertidumbre al extender el proyecto hasta dicho cumplimiento, pero al mismo tiempo garantiza la estabilidad del mismo en esta fase de definición, otorgando un nivel de calidad aceptable entre las partes.

Con la implantación de la guía se presume el cierre a la brecha que existe entre la expectativa del cliente y el alcance real del proyecto, si bien, el impacto de esto se refleja en la fase de definición, también es prudente, resaltar los resultados favorables en las fases definitivas del proyecto (desarrollo y pruebas), un ejemplo de esto, está en la planeación de casos de pruebas, pues en este aspecto, se evidencio un fortalecimiento en el diseño de casos de pruebas, fundamentadas en la especificación de los requerimientos, con esto se logró identificar casuísticas especiales propiciando en la ejecución de las pruebas resultados efectivos tanto en la tasa de defectos reportados como los diferentes escenarios de pruebas ejecutados.

²⁹ En el punto de Análisis e Interpretación de datos se presenta un análisis descriptivo por cada tasa o promedio que se presenta en la tabla 5.

2. Aunque el equipo del proyecto control cuenta con mayor experiencia por parte del Analista Funcional, aducen a esto para resolver los inconvenientes relacionados como se presentó en el levantamiento de información para el requerimiento de “Sugerencia de Abogados”, a diferencia de lo que se presentó para el proyecto Experimento donde el equipo no cuenta con la misma experiencia en el Analista funcional, sin embargo la guía ofreció pautas para identificar los avances que dejaba en evidencia los “síntomas” necesarios para generar alertas que permitan tomar acciones y aplicarlas oportunamente.

3. El uso de la guía que por su estructura se basa en los principios ágiles; la manera en que sugiere realizar las tareas, influye en emprender soluciones basadas en prácticas ágiles, situación que hace pensar en generar propuestas como mecanismos de mejora a los procesos de desarrollo de Software “*Software process Improvement*”, teniendo en cuenta que, sus recomendaciones van muy acompañadas del cómo se debe hacer facilitando la ejecución de estas mismas prácticas. Cabe aclarar, que independientemente la Guía genere una orientación a ejecutar prácticas ágiles, por la estructura que apropia de SEMAT, también dispone la flexibilidad de aplicarse a metodologías híbridas, e incluso podría aplicarse bajo metodologías tradicionales, pero para esto, deberá apropiarse un esquema diferente al planteado de “Way-Of-Thinking” valiéndose de las alfas de “Way of Working” de SEMAT.

4. En cuanto al uso de modelos, el diagrama de negocio bajo la notación BPMN y el modelo MON fueron los referentes durante la ejecución del proyecto experimento, el primero aportó en la definición del proceso cuando no se evidenciaba un alcance concreto para los requerimientos. En el caso del MON, impulsó al cliente a cuestionar la necesidad en términos de negocio (objetivo de este modelo), permitiendo orientar la solución en cuantificar sus resultados mediante indicadores, que permitan medir el aporte de la solución de Software al proceso de negocio dentro de una compañía; si sesgar la necesidad en un sentido individual.

5. Como parte del reto en el proyecto experimento, se resalta la complejidad en la definición de su alcance, pues a diferencia del proyecto control, el cliente no contemplaba una idea sólida de la necesidad como si se presentó para el proyecto experimento. Si bien los proyectos se plantean con características similares, al final de la ejecución, el consumo de horas resultó ser aproximado (1155,2 horas para Proyecto Experimento y 1054 para Proyecto Control), no obstante y como se presenta en el análisis en el punto “**Proyección de horas proyecto Experimento**”, probablemente de no haberse tenido en cuenta las consideraciones del guía en fases tempranas de definición de requerimientos, la proporción de crecimiento es considerable en cuanto a consumo de horas en fases definitivas de desarrollo y pruebas, arrojando un incremento de 976,1 horas adicionales.

6. Como se pudo evidenciar en la ejecución del diseño Quasi -Experimental, el proyecto Experimento en fases tempranas requiere un esfuerzo adicional para definir el alcance del proyecto en desarrollo, lo cual es inversamente proporcional para las fases definitivas del proceso de desarrollo de Software. Cuantificando estos resultados para la compañía desarrolladora de software, se considera favorable esta inversión puesto que el costo de un

ingeniero Analista Funcional es menor al costo de contar con los servicios de un Desarrollador de Software, aclarando que y en todo caso, el proceso de desarrollo partiendo de una continua aplicación de la guía supone madurar el proceso en las etapas correspondientes a la Ingeniería de requerimientos, en lo que el consumo de horas se vuelvan un recurso administrado eficientemente reduciendo los costos en relación para el proyecto de desarrollo de Software.

7. Como resultado a las observaciones generadas por el equipo de trabajo que participó en la implantación de la Guía, se decidió, que a diferencia de los modelos de organigrama, BPMN y MON, los demás modelos no fueron aplicados durante la ejecución de los proyectos por la complejidad y poca descripción que se referencia en la guía, pues la posibilidad de uso, generaba confusión y el aporte no era suficientemente claro en la práctica, teniendo en cuenta que para el equipo, el simple hecho de diseñarlos, tomaba más del tiempo que se estimaba para cualquier actividad relacionada a la definición de los requerimientos, aun considerando el tiempo adicional que ya representaba el tratar cada actividad de los estados de los alfas. Para el equipo, este tiempo que podría tomarse en el diseño de estos modelos, podría aprovecharse en otras actividades que para el equipo eran más significativas, ejemplo de esto, aplicar técnicas de obtención de requerimientos y el uso de diagramas de casos de uso. A raíz de lo anterior, para este equipo técnico, sugieren como trabajo futuro, hacer un énfasis de uso de estos modelos donde se represente mediante ejemplos el valor que puede aportar cada uno de estos, sin llegar a ser tan formales que le implique agregar un “peso” a un proceso que en principio se concibe como ágil como se propone en la Guía.

Trabajos Futuros

Adaptar alfas de TEAM y Way o Working: Partiendo de la experiencia de haber ejecutado la guía en proyectos reales, se considera la propuesta de trabajo futuro en ampliar el entorno de trabajo de desarrollo de Software agregando alfas de TEAM y Way of Working de SEMAT, con el objetivo, que la Guía pueda ampliar su metodología en diferentes espectros de metodologías de desarrollo de software incluyendo las tradicionales. Ejemplo de esto, se propone, se cree un paso previo antes de iniciar a la ejecución del proceso de ingeniería de requerimientos, para la construcción del equipo, partiendo del cumplimiento de los siguientes estados: “Seeded / Cultivado”, “Formed / Formado”. Para el caso en que la definición de requerimientos ya se esté ejecutando, se proponen los siguientes estados: “Colaborating / Colaborando” y “Performing/Ejecutando”. Para el aporte del alfa Way of Working en la guía se propone también incluir en esta primera iteración, la gestión de los siguientes estados: “Principles Established / Principios establecidos” y “Foundation Established / Bases establecidos”, una vez en ejecución se propone adelantar los siguientes estados: “In Use / En uso” y “Working Well / Trabajando Bien”.

Incluir métricas a los estados de las alfas en proceso de definición de requerimientos: Como propuesta de trabajo futuro, es diseñar un mecanismo de medición a los estados de los alfas que hacen parte del proceso de ingeniería de requerimientos, responsabilidad que se asignaría al gerente de proyectos / Scrum Master, con el objetivo de cuantificar la calidad con que se diseñan los requerimientos, esta iniciativa se origina, partiendo de la propuesta realizada por Carlos Mario Zapata, en su artículo “*On the Relationship of ISO/IEC 9126 Metrics and the Alpha States of the SEMAT Kernel*” [78] en el que propone la selección de las métricas apropiadas de la norma ISO / IEC 9126 para evaluar y validar Cada estado alfa con el fin de obtener un producto de software de alta calidad.

ANEXOS

ID	Nombre	Descripción	Ubicación
Anexo 1	Resumen de aplicaciones de SEMAT en Latinoamerica. docx	Resumen de aplicaciones de SEMAT.	/Entregables/Resumen Aplicaciones de SEMAT en LA/
Anexo 2	Bitácora_Horas_Consumidas_Proyecto_CONTROL.xlsx	Bitácora de horas consumidas en la ejecución del proyecto Control	/Entregables/Resultados ejecución proyectos/
Anexo 3	Bitácora_Horas_Consumidas_Proyecto_EXPERIMENTO.xlsx	Bitácora de horas consumidas en la ejecución del proyecto Experimento	/Entregables/Resultados ejecución proyectos/
Anexo 4	Bitácora_Horas_Consumidas_Proyecto_PILOTO.xlsx	Bitácora de horas consumidas en la ejecución del proyecto Piloto	/Entregables/Resultados ejecución proyectos/
Anexo 5	Resumen_Ejecución_Proyecto_CONTROL.doc	Resumen de la ejecución del proyecto Control	/Entregables/Resultados ejecución proyectos/
Anexo 6	Resumen_Ejecución_Proyecto_EXPERIMENTO.doc	Resumen de la ejecución del proyecto Experimento	/Entregables/Resultados ejecución proyectos/
Anexo 7	Resumen_Ejecución_Proyecto_PILOTO.doc	Resumen de la ejecución del proyecto Piloto	/Entregables/Resultados ejecución proyectos/
Anexo 8	Errores_y_Tiempos_Proyecto_CONTROL.xlsx	Bitácora de horas consumidas en el ajuste de errores del proyecto Control	/Entregables/Resultados ejecución proyectos/
Anexo 9	Errores_y_Tiempos_Proyecto_EXPERIMENTO.xlsx	Bitácora de horas consumidas en el ajuste de errores del proyecto Experimento	/Entregables/Resultados ejecución proyectos/
Anexo 10	Errores_y_Tiempos_Proyecto_PILOTO.xlsx	Bitácora de horas consumidas en el ajuste de errores del proyecto Piloto	/Entregables/Resultados ejecución proyectos/
Anexo 11	Perfiles_Involucrados_Proyectos.xlsx	Descripción de los perfiles de las personas involucradas en la ejecución de los proyectos	/Entregables/Otros Anexos/
Anexo 12	Definición_Variables_a_Medir.docx	Describe el proceso de medición de las variables seleccionadas	/Entregables/Variables a Medir/

ID	Nombre	Descripción	Ubicación
Anexo 13	Valor_Ganado_Verlocidad_Sprint_CONTROL.xlsx	Seguimiento y control a Sprints del proyecto Control	/Entregables/Resultados ejecución proyectos/
Anexo 14	Valor_Ganado_Verlocidad_Sprint_EXPERIMENTO.xlsx	Seguimiento y control a Sprints del proyecto Experimento	/Entregables/Resultados ejecución proyectos/
Anexo 15	Valor_Ganado_Velocidad_Sprint_PILOTO.xlsx	Seguimiento y control a Sprints del proyecto Piloto	/Entregables/Resultados ejecución proyectos/
Anexo 16	Actas de Seguimiento Ejecución del Proyecto.rar	Contiene las actas de los seguimientos de cada uno de los proyectos	/Entregables/Otros Anexos/
Anexo 17	FO-01-Aceptación	Encuesta Aceptación de la guía	/Entregables/Variables a Medir/Encuesta de Aceptación/
Anexo 18	FO-SF-070 EspecificacionFuncional.doc	Documento de Especificación	/Entregables/Otros Anexos/Formatos/
Anexo 19	FO-SF-71 CasosdeUso.doc	Documentos de casos de Uso	/Entregables/Otros Anexos/Formatos/
Anexo 20	dromero.zip	Producto de software diseñado para soportar los datos de métricas almacenados	/Entregables/Aplicación de Software Apoyo/Fuentes/
Anexo 21	Manual de instalación aplicación de apoyo.pdf	Manual de instalación de la aplicación de apoyo	/Entregables/Aplicación de Software Apoyo/Manual de Instalación/
Anexo 22	Manual de usuario aplicación de apoyo.pdf	Manual de usuario aplicación de apoyo	/Entregables/Aplicación de Software Apoyo/Manual de Usuario/
Anexo 23	FO-02- Historia de Usuarios.doc	Formato diseñado para registrar las necesidades de la aplicación soporte	/Entregables/Aplicación de Software Apoyo/Documentos de Obtención y Análisis/
Anexo 24	Diagrama_Clases.png	Diagrama de clases construido para la aplicación soporte al proyecto	/Entregables/Aplicación de Software Apoyo/Documentos de Obtención y Análisis/
Anexo 25	Diagrama_entidad_relacion_Imagen.png	Diagrama de entidad relación construido para la aplicación soporte al proyecto	/Entregables/Aplicación de Software Apoyo/Documentos de Obtención y Análisis/

ID	Nombre	Descripción	Ubicación
Anexo 26	Resultados Consolidados Encuesta Aceptación.xlsx	Resultados consolidados sobre las encuestas de aceptación realizadas en las diferentes instancias.	/Entregables/Variables a Medir/Encuesta de Aceptación/
Anexo 27	PlantillaEstimacion_PuntosCasosdeUso_Experimento.xlsx	Detalle de cálculo de puntos por casos de uso proyecto experimento y proyecto control	/Entregables/Otros Anexos/

REFERENCIAS

- [1] Mahsa Hasani Sadi and Eric Yu, “Analyzing the evolution of software development: From creative chaos to software ecosystems,” *Res. Chall. Inf. Sci. RCIS 2014 IEEE Eighth Int. Conf. On*, May 2014.
- [2] A. E. F. Maurer and F.E. Paetsch, “Requirements Engineering and Agile Software Development,” *12th IEEE Int'l Workshop Enabling Technol. Infrastruct. Collab. Enterp. Wetice 03*, p. 308, 2003.
- [3] Williams, L, “Agile Software Development Methodologies and Practices,” *Adv. Comput.*, vol. 80, pp. 1–44, 2010.
- [4] Kent Beck, “Embracing Change with Extreme Programming,” presented at the Computer, 1999, vol. 32, pp. 70–77.
- [5] K. Schwaber, “Scrum Development Process,” presented at the Workshop on Business Object Design and Implementation OOPSLA'95, 1995.
- [6] Jennifer Stapleton, *DSDM, Dynamic Systems Development Method: The Method in Practice*, 3rd ed. Ashford, Eng: Addison Wesley, 1997.
- [7] “Adaptive Software Development: A Collaborative Approach to Managing Complex ... - Jim Highsmith - Google Libros,” 27-Sep-2014. [Online]. Available: http://books.google.com.co/books?hl=es&lr=&id=CVcU-AAAAQBAJ&oi=fnd&pg=PA1&dq=Adaptive+Software+Development:+A+Collaborative+Approach+to+managing+complex+systems&ots=5rxCbvFvAM&sig=vgJ6hHrTHzJyG9NtaXDIdbr9UZg&redir_esc=y#v=onepage&q=Adaptive%20Software%20Development%3A%20A%20Collaborative%20Approach%20to%20managing%20complex%20systems&f=false. [Accessed: 27-Sep-2014].

- [8] Alistair Cockburn and Jim Highsmith, "Agile Software Development: The People Factor," *IEEE Computer Society*, vol. 34, pp. 31–133, 01-Nov-2001.
- [9] Kevin Vlaanderen, Slinger Jansen, Sjaak Brinkkemper, and Erik Jaspers, "The agile requirements refinery: Applying SCRUM principles to software product management," *Inf. Softw. Technol.*, vol. 53, no. 1, pp. 58–70, Jan. 2011.
- [10] M. Rizwan Jameel Qureshi, "Agile software development methodology for medium and large projects," *IET Softw.*, vol. 6, no. 4, pp. 358 – 363, Oct. 2012.
- [11] Maya Daneva *et al.*, "Agile requirements prioritization in large-scale outsourced system projects: An empirical study," *J. Syst. Softw.*, vol. 85, no. 5, pp. 1333–1353, May 2013.
- [12] Scott W. Ambler and Mark Lines, "Disciplined Agile Delivery," presented at the Disciplined Agile Consortium, 2013.
- [13] Juan Darío Murcia Torres, "Guía para la Integración de Métodos Formales de Ingeniería de Requerimientos en Procesos de Desarrollo Ágil." 17-Jun-2014.
- [14] Ivar Jacobson, Pan-Wei Ng, Paul E. McMahon, Ian Spence, and Svante Lidman, "The Essence of Software Engineering: The SEMAT Kernel," *Commun. ACM*, vol. 55, no. 12, pp. 42–49, Dec. 2012.
- [15] Claes Wohlin, Magnus C. Ohlsson, Per Runeson, Björn Regnell, Martin Host, and Anders Wesslén, *Experimentation in Software Engineering*. Springer, 2012.
- [16] Margarita André, María G. Baldoquín, and Silvia T. Acuña, "Formal model for assigning human resources to teams in software projects," *Elsevier Sci.*, vol. 53, no. 3, pp. 259–275, 2011.
- [17] Thomas D. Cook and Donald Thomas Campbell, *Quasi-Experimentation: Design & Analysis Issues for Field Settings*. Houghton Mifflin, 1979.
- [18] KEN PEFFERS, TUURE TUUNANEN, MARCUS A. ROTHENBERGER, and SAMIR CHATTERJEE, "A Design Science Research Methodology for Information Systems Research," *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2008.
- [19] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram, "DESIGN SCIENCE IN INFORMATION SYSTEMS RESEARCH," *MIS Q.*, vol. 28, pp. 75–105, Mar. 2004.
- [20] D. I. K. Sjoeborg *et al.*, "A survey of controlled experiments in software engineering," *IEEE Trans. Softw. Eng.*, vol. 31, no. 9, pp. 733–753, Sep. 2005.
- [21] William R., Jr. Shadish, Thomas D. Cook, and Donald T. Campbell, *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*, 2 International. 656: Houghton Mifflin, 2002.

- [22] Vigdis By Kampenes, Tore Dybå, Jo E. Hannay, and Dag I. K. Sjøberg, “A systematic review of quasi-experiments in software engineering,” *Inf. Softw. Technol.*, vol. 51, no. 1, p. Pages 71–82, Jan. 2009.
- [23] Asgari, S, Baptista Nunes, and J.M.B., “Experimental and quasi-experimental research in information systems,” *Int. Assoc. Dev. Inf. Soc. Iadis*, Jul. 2011.
- [24] A. Bryman, *Social research methods*. Oxford; New York: Oxford University Press, 2008.
- [25] A. M. Grant and T. D. Wall, “The Neglected Science and Art of Quasi-Experimentation Why-to, When-to, and How-to Advice for Organizational Researchers,” *Organ. Res. Methods*, vol. 12, no. 4, pp. 653–686, Oct. 2009.
- [26] L. Cohen, L. Manion, and K. Morrison, *Research methods in education*. London; New York: Routledge, 2007.
- [27] Herbert A. Simon, *The Sciences of the Artificial*, Third edition. London England, Cambridge Massachusetts: MIT Press, 1996.
- [28] M. Haran *et al.*, “Techniques for Classifying Executions of Deployed Software to Support Software Engineering Tasks,” *IEEE Trans. Softw. Eng.*, vol. 33, no. 5, pp. 287–304, May 2007.
- [29] P. Järvinen, “Action Research is Similar to Design Science,” *Qual. Quant.*, vol. 41, no. 1, pp. 37–54, Feb. 2007.
- [30] Ivar Jacobson, Pan-Wei Ng, Paul E. McMahon, Ian Spence, and Svante Lidman, “La Esencia de la Ingeniería de Software: El Núcleo de Semat,” *Rev. Latinoam. Ing. Softw.*, vol. 1, no. 3, pp. 71–78.
- [31] F. Daoudi and S. Nurcan, “A framework to evaluate methods’ capacity to design flexible business processes,” in *6th International Workshop on Business Process Modeling*, Porto, Portugal, 2006, vol. 12, pp. 1–8.
- [32] Anthony Chen and Joy Beatty, *Visual Models for Software Requirements (Developer Best Practices)*, 1st Edition. Microsoft Press, 2012.
- [33] Endres, A. and Rombach, D, *Empirical Software Engineering: A Handbook of Observations Laws and Theories*. Harlow, UK: Addison-Wesley, 2003.
- [34] Ian Sommerville, *SOFTWARE ENGINEERING*, 9th ed. Addison-Wesley, 2011.
- [35] IEEE, “Guide to the software engineering body of knowledge.” IEEE Computer Society, 2004.

- [36] R. Rana and M. Staron, "Machine learning approach for quality assessment and prediction in large software organizations," in *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2015, pp. 1098–1101.
- [37] IEEE Computer Society, "IEEE Standard Adoption of ISO/IEC 15939:2007— Systems and Software Engineering— Measurement Process." 10-Dec-2008.
- [38] Shahid Iqbal and Naeem Ahmed Khan, "Yet another Set of Requirement Metrics for Software Projects," *Int. J. Softw. Eng. Its Appl.*, vol. 6, no. 1, Jan. 2012.
- [39] Patrick Kua, "An Appropriate Use of Metrics," *Martin Fowler*, 19-Feb-2013.
- [40] L. Buglione and A. Abran, "Measurement Process: Improving the ISO 15939 Standard," in *2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement*, 2014, pp. 136–140.
- [41] *Software Engineering A Practitioner's Approach*, 8th ed. McGraw-Hill, 2014.
- [42] "Standard & Poor's | Americas." [Online]. Available: http://www.standardandpoors.com/en_US/web/guest/home. [Accessed: 03-May-2017].
- [43] "ISO 9001:2008 - Quality management systems -- Requirements." [Online]. Available: <https://www.iso.org/standard/46486.html>. [Accessed: 03-May-2017].
- [44] "ISO/IEC 27001:2013 - Information technology -- Security techniques -- Information security management systems -- Requirements." [Online]. Available: <https://www.iso.org/standard/54534.html>. [Accessed: 03-May-2017].
- [45] "Certifications | CMMI Institute." [Online]. Available: <http://cmmiinstitute.com/certifications>. [Accessed: 13-May-2017].
- [46] Comisión Europea and NU. CEPAL, *Desafíos y oportunidades de la industria del software en América Latina*. Bogotá: Mayol Ediciones, 2009.
- [47] Karla Cristina Palomino Zuluaga, "ESTUDIO DEL COMPORTAMIENTO DE LA INDUSTRIA DEL SOFTWARE EN COLOMBIA ANTE ESCENARIOS DE CAPACIDADES DE INNOVACION Y VENTAJAS COMPARATIVAS POR MEDIO DE DINAMICA DE SISTEMAS," Universidad Nacional de Colombia, Bogotá, 2011.
- [48] Andrew Rogers, "Colombia to train over 1,000 professionals in agile software development framework," 27-May-2013.
- [49] Patty Pedroza Barrios, "Choosing a Development Methodology from the Advantages of an Agile Methodology and a Robust Model like CMMI-DEV 1.3," *INGENIARE - Universidad Libre*, Barranquilla, pp. 113–122, May-2013.

- [50] Ken Schwaber and Jeff Sutherland, "The Scrum guide." Jul-2013.
- [51] Mike Cohn, *Agile Estimating and Planning*, 1st ed. Prentice Hall, 2005.
- [52] Marta N. Gómez and Silvia T. Acuña, "A replicated quasi-experimental study on the influence of personality and team climate in software development," *Springer Sci. Media*, Aug. 2013.
- [53] K. Kurtel, "Measuring and Monitoring Software Maintenance Services: An Industrial Experience," in *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement*, 2013, pp. 247–252.
- [54] Natalia Valeria Andriano, "Comparación del Proceso de Elicitación de Requerimientos en el desarrollo de Software a Medida y Empaquetado. Propuesta de métricas para la elicitación," Universidad Blas Pascal – Universidad Nacional de la Plata, La Plata, Argentina, 2006.
- [55] Davis, A, "Identifying and measuring quality in a software requirements Specification," *Software Metrics Symposium*, pp. 141–152, 1993.
- [56] J. Sutherland, S. Downey, and B. Granvik, "Shock Therapy: A Bootstrap for Hyper-Productive Scrum," 2009, pp. 69–73.
- [57] S. Downey and J. Sutherland, "Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft," 2013, pp. 4870–4878.
- [58] T. Sulaiman, B. Barton, and T. Blackburn, "AgileEVM - Earned Value Management in Scrum Projects," 2006, pp. 7–16.
- [59] E. Aranha and P. Borba, "ESTIMATING MANUAL TEST EXECUTION EFFORT AND CAPACITY BASED ON EXECUTION POINTS," *Int. J. Comput. Appl.*, vol. 31, no. 3, pp. 167–172, 2009.
- [60] Ming - Chang Lee and To Chang, "Software Measurement and Software Metrics in Software Quality," *Int. J. Softw. Eng. Its Appl.*, vol. 7, no. 4, Jul. 2013.
- [61] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis, "User acceptance of information technology: Toward a unified view," *MIS Q.*, vol. 27, no. 3, pp. 425–478, Sep. 2003.
- [62] Y. Varela, L. Antonio, R. Tovar, L. Arturo, and J. Chaparro, "Technological acceptance model (TAM): a study of the influence of the national culture and of the user profile in the use of ICTS," *Innovar*, vol. 20, no. 36, pp. 187–203, Jan. 2010.
- [63] R. Zhang, Z. Zhang, K. Liu, and J. Zhang, *LISS 2013: Proceedings of 3rd International Conference on Logistics, Informatics and Service Science*. Springer, 2014.

- [64] ISO, “Guía Técnica ISO / TR 10017: 2003.” 2003.
- [65] Jack R. Fraenkel and Norman E. Wallen, *How to Design and Evaluate Research in Education*. San Francisco State University: McGraw-Hill, 2008.
- [66] “Luis Antonio Salazar Caraballo | LinkedIn.” [Online]. Available: <https://www.linkedin.com/in/luchosalazar>. [Accessed: 30-May-2016].
- [67] “¿Cual es el impacto de semat en el futuro de la Ingeniería de Software?,” *Canal En VIVO*, 14-Oct-2011. .
- [68] C. M. Z. J. Claudia Elena Durango Vanegas, “Una representación basada en Semat y RUP para el Método de Desarrollo SIG del Instituto Geográfico Agustín Codazzi,” vol. 6, Jun-2015.
- [69] “PMI | Project Management Institute.” [Online]. Available: <https://www.pmi.org/>. [Accessed: 13-Mar-2017].
- [70] “What is Technology Acceptance Model (TAM) | IGI Global.” [Online]. Available: <http://www.igi-global.com/dictionary/technology-acceptance-model-tam/29485>. [Accessed: 10-May-2016].
- [71] “CakePHP - Build fast, grow solid | PHP Framework | Home,” *CakePHP - The rapid development php framework*. [Online]. Available: <https://cakephp.org/>. [Accessed: 13-May-2017].
- [72] “Welcome to The Apache Software Foundation!” [Online]. Available: <https://www.apache.org/>. [Accessed: 13-May-2017].
- [73] “MySQL :: Download MySQL Community Server.” [Online]. Available: <https://dev.mysql.com/downloads/mysql/>. [Accessed: 13-May-2017].
- [74] “PHP: Hypertext Preprocessor.” [Online]. Available: <http://php.net/>. [Accessed: 13-May-2017].
- [75] “The MIT License | Open Source Initiative.” [Online]. Available: <https://open-source.org/licenses/mit-license.php>. [Accessed: 13-May-2017].
- [76] Ivar Jacobson, Ian Spence, and Kurt Bittner, *USE-CASE 2.0*. IVAR JACOBSON INTERNATIONAL SA., 2013.
- [77] “Welcome To UML Web Site!” [Online]. Available: <http://www.uml.org/>. [Accessed: 25-Apr-2017].
- [78] Carlos Mario Zapata Jaramillo and Yury Montoya-Pérez, “On the Relationship of ISO/IEC 9126 Metrics and the Alpha States of the SEMAT Kernel,” *IEEE*, Apr. 2016.