

PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN  
PEQUEÑAS ORGANIZACIONES DE DESARROLLO  
SOFTWARE



KELLY YOHANNA ZUÑIGA SILVA

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE SISTEMAS  
Grupos IDIS  
Popayán  
2015

# PROCEDIMIENTO DE ANÁLISIS CAUSAL ENFOCADO EN PEQUEÑAS ORGANIZACIONES DE DESARROLLO SOFTWARE



KELLY YOHANNA ZUÑIGA SILVA

Monografía de trabajo de grado

Director: PhD. Francisco José Pino Correa  
Co-director: Mag. Carlos Alberto Ardila Albarracín

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE SISTEMAS  
Grupos IDIS  
Popayán  
2015

**Nota de aceptación:**

---

---

---

---

---

---

---

**Firma del Presidente del jurado**

---

**Firma del jurado**

---

**Firma del jurado**

**Popayán, 2015**

## **DEDICATORIA**

*Este trabajo está dedicado a mis padres que me han brindado todo su apoyo y confianza, también dedico este trabajo a mis amigos y compañeros que de una u otra manera han aportado en mi crecimiento profesional y personal.*

*Kelly Yohanna Zuñiga*

## **AGRADECIMIENTOS**

*Primeramente quiero agradecerle a Dios por ayudarme a alcanzar mis metas y darme la fortaleza para superar los obstáculos presentados. A mis padres y mi hermana por su apoyo incondicional. A mis amigos por sus palabras de aliento y el apoyo dado en los momentos difíciles. A mis docentes por su vocación, empeño y dedicación en mi formación profesional, especialmente a mi director Francisco José Pino Correa y mi Co-director Carlos Alberto Ardila Albarracín y finalmente a la universidad del Cauca por la formación brindada a nivel personal y profesional.*

## CONTENIDO

<b>Capítulo 1. Introducción</b> .....	11
<b>1.1. Planteamiento del problema</b> .....	12
<b>1.1. Justificación</b> .....	13
<b>1.2. Objetivos</b> .....	15
1.2.1. Objetivo General .....	15
1.2.2. Objetivos específicos .....	15
<b>1.3. Solución propuesta</b> .....	16
<b>1.4. Estrategia de investigación</b> .....	16
<b>1.5. Estructura del documento</b> .....	18
<b>Capítulo 2. Marco teórico y trabajos relacionados</b> .....	20
<b>2.1. Marco teórico</b> .....	20
2.1.1. Análisis causal.....	20
2.1.2. Esquema de clasificación ortogonal de defectos (ODC, Orthogonal Defect Classification) .....	21
2.1.3. Pequeñas organizaciones VSEs.....	23
2.1.3.1. Importancia de las pequeñas organizaciones.....	24
2.1.4. Business Process Modeling Notation (BPMN) [20] .....	25
<b>2.1.5. Capability Maturity Model Integration (CMMI-DEV 1.3)</b> .....	27
2.1.6. ISO/IEC 12207 –Ingeniería del software y sistemas- Procesos del ciclo de vida del software.....	27
2.1.7. ISO/IEC 15504-5 modelo de evaluación de procesos del ciclo de vida del software.....	28
<b>2.2. Trabajos relacionados</b> .....	28
2.2.1. Herramientas que apoyan el análisis causal .....	28
2.2.2. Métodos.....	29
2.2.3. Técnicas .....	32
<b>2.3. Discusión</b> .....	32
<b>2.4. Aporte</b> .....	36
<b>Capítulo 3. Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software</b> .....	37

<b>3.1. Comparación de referentes internacionales</b>	37
<b>3.2. Selección de técnicas para análisis causal</b>	44
<b>3.3. Definición del procedimiento</b>	48
3.3.1. Vista general de procedimiento	48
3.3.1.1. Propósito	48
3.3.1.2. Objetivo	48
3.3.1.3. Actividades y tareas	48
3.3.1.4. Roles	48
3.3.1.5. Productos de trabajo	49
3.3.1.6. Diagrama de actividades	49
3.3.2. Vista detallada del procedimiento	51
3.3.2.1. Descripción	51
<b>Capítulo 4. Caso de estudio</b>	66
<b>4.1. Antecedentes</b>	66
<b>4.2. Diseño del caso de estudio</b>	67
<b>4.3. Sujetos de investigación y unidad de análisis</b>	67
<b>4.4. Procedimiento de campo</b>	68
<b>4.5. Intervención</b>	69
4.5.1. Diagnóstico inicial	69
4.5.1.1. Adaptaciones al procedimiento	70
4.5.2. Asignación del proyecto piloto	70
4.5.3. Ejecución del procedimiento	73
<b>4.6. Recolección y análisis de datos</b>	74
4.6.1. Esfuerzo	74
4.6.2. Causas encontradas	81
4.6.3. Percepción de las personas involucradas en el proyecto	85
<b>Capítulo 5. Conclusiones, trabajos futuros y lecciones aprendidas</b>	89
<b>5.1. Conclusiones</b>	89
<b>5.2. Trabajos futuros</b>	91
<b>5.3. Lecciones aprendidas</b>	92

## LISTA DE TABLAS

Tabla 1. Atributos de la clasificación ODC .....	22
Tabla 2. Elementos de la notación BPMN.....	26
Tabla 3. Comparación de trabajos relacionados con análisis causal.....	34
Tabla 4. Estructura de los referentes internacionales.....	38
Tabla 5. Actividades y tareas para la resolución de problemas software-ISO/IEC 12207.	39
Tabla 6. Prácticas para la resolución de problemas software-ISO/IEC 15505-5.....	39
Tabla 7. Metas, prácticas y subprácticas para el análisis causal- CMMI-DEV.....	40
Tabla 8. Listado de las actividades de los referentes internacionales .....	42
Tabla 9. Listado de las actividades de los referentes internacionales .....	43
Tabla 10. Actividades del procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software .....	43
Tabla 11. Actividades adicionales del procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software .....	44
Tabla 12. Relación técnica-actividad.....	45
Tabla 13. Identificación de Roles involucrados .....	49
Tabla 14. Definición específica de la actividad Preparación.....	54
Tabla 15. Definición específica de la actividad Detección de defectos .....	58
Tabla 16. Diagrama de actividad Detección de causas fundamentales.....	62
Tabla 17. Diagrama de actividad Documentar .....	64
Tabla 18. Preguntas de investigación del caso de estudio .....	66
Tabla 19. Fases del procedimiento .....	73
Tabla 20. Esfuerzo involucrado en cada actividad Detección de defectos .....	75



Tabla 21. Esfuerzo involucrado en cada actividad Detección de causas fundamentales .	76
Tabla 22. Esfuerzo involucrado en cada actividad Documentar .....	77
Tabla 23. Esfuerzo involucrado en el procedimiento .....	77
Tabla 24. Esfuerzo total del líder de análisis causal.....	79
Tabla 25. Esfuerzo total del proyecto.....	80
Tabla 26. Lista de causas encontradas por fase .....	83
Tabla 27. Cantidad de causas encontradas por fase .....	83
Tabla 28. Percepción de los involucrados en el proyecto.....	86

## LISTA DE FIGURAS

Figura 1. Estrategia de investigación .....	17
Figura 2. Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software .....	50
Figura 3. Diagrama de actividad Preparación .....	54
Figura 4. Diagrama de actividad Detección de defectos .....	58
Figura 5. Detección de causas fundamentales.....	63
Figura 6. Diagrama de actividad Documentar .....	65
Figura 7. Estructura del proyecto “Sistemas de gestión de actividades diarias” .....	68
Figura 8. Procedimiento de campo que rige las actividades del caso de estudio .....	69
Figura 9. Relación de tipos de defectos .....	73
Figura 10. Esfuerzo involucrado en cada fase .....	78
Figura 11. Cantidad de causas encontradas por fase .....	83

# Capítulo 1. Introducción

---

Según Pino et al. [1] las pequeñas y medianas empresas representan el 99,2% de las empresas de desarrollo software. Por lo tanto, aportan de forma significativa a la economía, mediante el desarrollo de productos y servicios [2]. Sin embargo las pequeñas organizaciones<sup>1</sup> (VSEs, Very Small Entities) se enfrentan a desafíos relacionados con problemas de calidad y no cuentan con los recursos necesarios para afrontarlos. Esta es una de las principales diferencias con las grandes organizaciones, esto lleva a que modelos como CMMI no pueden ser adoptados por su complejidad y costo [3].

En este sentido, Czibula et al. [4] sostienen que la calidad del software está directamente relacionada con la ausencia de defectos y se considera importante en el campo de la ingeniería del software. Cuando estos defectos no son detectados o se detectan de forma tardía, en primer lugar acarrearán consecuencias como retrasos en las fechas de entrega, molestias en el cliente y aumento del costo y el esfuerzo involucrado; y en segundo lugar pueden necesitar un esfuerzo significativo para su corrección o hallazgo en etapas posteriores al desarrollo del software [5].

Teniendo en cuenta la problemática anteriormente expuesta, existen formas para identificar la causa de los defectos, entre ellas el análisis causal que permite mejorar los procesos, hallar las causas y prevenir los defectos en el desarrollo de productos software [6]. Así, se ve reflejada la importancia de implementar este análisis en las VSEs. En este sentido, es pertinente definir un procedimiento que integre actividades y técnicas apropiadas, para aplicar eficazmente el análisis causal en pequeñas organizaciones de desarrollo software, con el fin de mejorar la calidad, reducir sobrecostos y aportar en su desarrollo y evolución.

---

<sup>1</sup> Según la norma ISO/IEC 29110 una pequeña organización es una empresa, organización, departamento o proyecto de hasta 25 personas.

# Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

## 1.1. Planteamiento del problema

El análisis causal hace parte de modelos de calidad tales como CMMI DEV 1.3 [7], ISO/IEC 12207 [8], ISO/IEC 15504 [9] y Six Sigma [10]. Esta área ayuda a mejorar los procesos en las organizaciones desarrolladoras de software, debido a que contribuye en la identificación de causas que generan defectos durante el ciclo de vida del software. Por ejemplo Leszak et al. [11] realizan una clasificación de defectos y analizan sus posibles causas, entre las que se destacan: i) defectos de interfaz, causados por inesperadas interacciones, seguida de la funcionalidad y el diseño de datos, ii) defectos de implementación que pueden derivarse de una mala especificación de los componentes y de la arquitectura del sistema, iii) defectos humanos que pueden generarse por la falta de conocimiento en el sistema y por problemas de comunicación. Asimismo, el análisis causal permite hallar oportunidades de mejora e implementar acciones, para reducir la manifestación continua del mismo tipo de error<sup>2</sup> en proyectos futuros [12].

Una vez se reconoce el defecto, es importante hacer un análisis con el fin de determinar cuáles son los factores que lo causan. Pero identificar las causas de los problemas no es tarea fácil [13], además al no tener clara la causa principal, se toman decisiones erróneas que no dan solución a la problemática presentada o peor aún, se toman decisiones que agudizan la situación, provocando costos innecesarios a nivel económico y humano [14]. Así las organizaciones desarrolladoras de software buscan la manera de identificar las causas que generan problemas durante el desarrollo de proyectos software, pero esta labor es difícil de llevar a cabo, especialmente porque no se logra identificar la causa principal que está ocasionando resultados no deseados [15].

De igual modo, la forma de ejecutar análisis causal no es clara, porque cada persona, consultor u organización define el análisis de causa raíz de forma diferente, lo cual lleva a ejecutar actividades incorrectas que se creen necesarias. Esto representa un problema, puesto que las empresas desarrolladoras de software se limitan a determinadas sugerencias dadas dentro de métodos, guías o normas internacionales, sin aplicar eficazmente el análisis de causa raíz; esto puede provocar que el análisis causal se convierta en una causa de problemas [16]. Por otro lado, resolver defectos

---

<sup>2</sup>Según el estándar IEEE 610.12-1990 un error es una acción humana que produce un resultado incorrecto.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

producidos durante la ejecución de proyectos software es un trabajo difícil en muchas organizaciones [15], porque no se clasifican los tipos de defectos y aun cuando esta tarea se realiza, no se cuenta con las herramientas óptimas para aprender de los problemas y sus causas [14].

La búsqueda de evidencias sobre la aplicación del análisis causal en pequeñas organizaciones arroja pocos resultados, puesto que no se identifica un procedimiento para ejecutar análisis causal enfocado en el contexto de las pequeñas organizaciones. Tal vez las organizaciones no se han motivado para llevar a cabo análisis causal, al pensar que estas actividades no son necesarias, pues corresponden a un nivel de madurez que por lo general no se pretende alcanzar, sin embargo el análisis causal ayuda a encontrar áreas de mejora, permitiendo que la organización incremente su nivel de madurez sin importar el tipo de certificación buscado [17].

En relación a la problemática expuesta, y con la intención de ser fuente de motivación en futuras investigaciones, que mejoren las propuestas desarrolladas dentro del análisis causal, se hace necesario plantear la siguiente pregunta de investigación: ¿Cómo identificar de forma sistemática las causas de los problemas en el entorno de las pequeñas empresas desarrolladoras de software?

Para dar respuesta a esta pregunta de investigación se trabaja en este proyecto de grado, el cual pretende desarrollar un procedimiento que proporcione al personal encargado del análisis causal en esta clase de organizaciones, algunas pautas que permitan identificar los problemas de mayor incidencia, lo cual contribuye a tomar acciones correctivas y elegir la forma de tratar las causas de los problemas detectadas durante el desarrollo del software.

### **1.1. Justificación**

“El análisis causal se enfoca en comprender las relaciones causa efecto. Un sistema causal es un conjunto de causas que interactúan, produciendo efectos reconocibles” [14]. Es importante efectuar análisis causal en las pequeñas organizaciones, puesto que contribuye en la mejora de la calidad de los procesos, además, es probable prevenir defectos y detectarlos a tiempo, así es posible aprender de ellos y reducir la probabilidad de cometer errores cuando se ejecuta un nuevo proyecto.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

También el análisis causal es un método de bajo costo [17]. La inversión puede variar entre 0.5% y 1.5% del costo total del proyecto, por lo tanto es factible recuperar el dinero invertido y disminuir la tasa de error de los defectos producidos durante el desarrollo del software en un porcentaje superior al 50% [18]. Cabe resaltar que la detección temprana de los defectos es beneficiosa, ya que estos van a ser tratados a tiempo, evitando demoras en la ejecución del proyecto [11]. De las evidencias anteriores es posible decir que las pequeñas empresas desarrolladoras de software, necesitan un procedimiento que se adapte a sus características y les permita realizar análisis causal de forma sistemática.

En este sentido, las actividades relacionadas con análisis causal son tenidas en cuenta en diferentes modelos de mejora de procesos, por mencionar algunos de ellos CMMI DEV 1.3 [7], ISO/IEC 12207 [8] e ISO/IEC 15504 [9]. CMMI DEV define el análisis causal con el fin de optimizar el proceso, en primera instancia se centra en plantear o enumerar las prácticas y metas a lograr para identificar causas, para luego concentrarse en acciones específicas con el objeto de mejorar los resultados. ISO/IEC 15504-5 señala que la innovación del proceso es una medida del grado en que los cambios son identificados, a partir del análisis de las causas comunes que inducen la variación en el rendimiento, por lo cual se analizan los datos con el propósito de identificar oportunidades para las mejores prácticas y la innovación. Por otro lado ISO/IEC 12207 establece un marco para los procesos del ciclo de vida del software. Dentro de los procesos definidos en esta norma, se encuentra el proceso para la solución de problemas software, especificando dentro de sus actividades la resolución de problemas, lo cual implica identificar el problema y sus posibles causas. Cabe resaltar que los modelos mencionados anteriormente sólo plantean el qué se debe hacer, sin especificar el cómo se debe ejecutar el proceso.

Este proyecto pretende construir un procedimiento que se adapte a las características de las pequeñas organizaciones<sup>3</sup>, teniendo en cuenta los elementos que intervienen en el análisis causal y las herramientas que permiten establecer las causas de los problemas. Al mismo tiempo se busca aportar en la calidad del proceso software y mantener evidencia para proyectos futuros de la problemática presentada durante el ciclo de vida del software, además de proporcionar pautas para ejecutar análisis causal de modo que motive a las pequeñas organizaciones, ya que la mayoría de estudios en

---

<sup>3</sup> Según lo expresado en [13] las pequeñas organizaciones tienen sus propias características y limitaciones.

# Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

análisis de causa raíz se han realizado para las grandes compañías de software que poseen un alto nivel de madurez [6]. Por otra parte, referentes internacionales, métodos, técnicas y guías, entre otros, no son apropiados para éstas, debido a que las sugerencias propuestas pueden resultar muy costosas o no se cuenta con personal que tenga los conocimientos para implementar las indicaciones allí expuestas. Por estos motivos se busca construir un procedimiento que guíe a las pequeñas organizaciones en la ejecución de análisis causal.

## 1.2. Objetivos

### 1.2.1. Objetivo General

Proponer un procedimiento<sup>4</sup> que apoye la ejecución del análisis causal en pequeñas organizaciones desarrolladoras de software.

### 1.2.2. Objetivos específicos

- ❖ Determinar los elementos que intervienen en el análisis causal y las herramientas<sup>5</sup> que permiten establecer las causas del problema.
- ❖ Construir el procedimiento para guiar la ejecución del análisis causal a partir de los elementos identificados y modelar el procedimiento propuesto mediante el estándar de representación BPMN<sup>6</sup>.
- ❖ Evaluar el procedimiento propuesto mediante un método de evaluación por expertos o mediante un caso de estudio académico.

---

<sup>4</sup> Según la norma ISO/IEC 9000 un procedimiento es una forma especificada para llevar a cabo una actividad o un proceso.

<sup>5</sup> Para el desarrollo de este trabajo una herramienta es un objeto que facilita la ejecución de análisis causal

<sup>6</sup>Según [54] Business Process Modeling Notation (BPMN) es una notación gráfica para la elaboración de los procesos de negocio.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### **1.3. Solución propuesta**

Este trabajo de grado plantea un procedimiento de análisis causal, que integra actividades y técnicas orientadas a las necesidades y características de las pequeñas organizaciones de desarrollo de software. El aporte investigativo, está en hallar las mejores prácticas y las técnicas adecuadas para realizar análisis causal en las pequeñas organizaciones y lograr que éstas hagan parte del procedimiento a construir con el que se pretende guiar a las VSEs en cómo llevar el análisis causal de manera sistemática, guiando paso a paso la elaboración de los artefactos o productos de trabajo necesarios para este fin, y por último, se va a dar un aporte tecnológico mediante el modelado del procedimiento propuesto

### **1.4. Estrategia de investigación**

El proyecto utilizó el método cualitativo Investigación-Acción (AR, Action Research) según las adaptaciones realizadas por Pino et al. [19]. La propuesta planteada involucra un ciclo de investigación y uno de solución de problemas (Figura 1).



## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

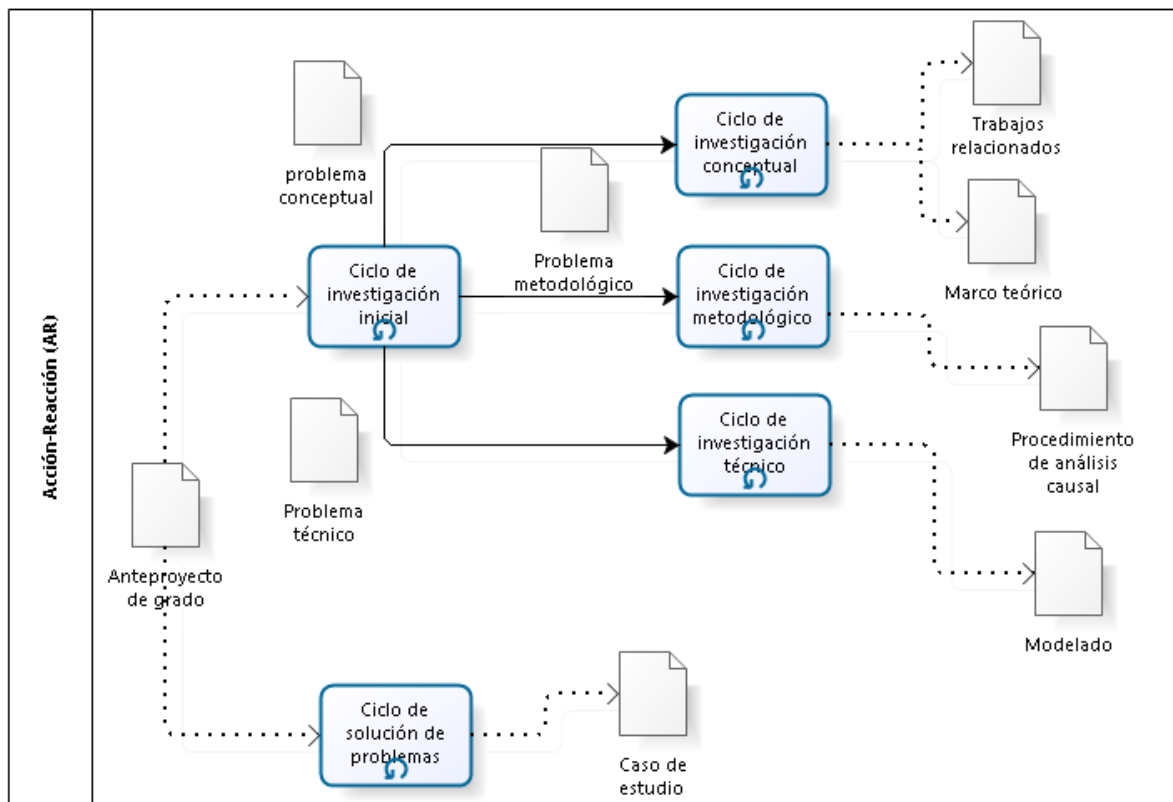


Figura 1. Estrategia de investigación

### 1. Ciclo de Investigación

Este ciclo parte de una investigación inicial, en el cual se abordan problemas de tipo conceptual, metodológico y técnico.

La investigación conceptual permite identificar el marco teórico y los trabajos relacionados con el análisis causal, mediante dos fases: identificación del problema y recopilación de datos. En la primera se determina los trabajos desarrollados a partir del análisis causal en organizaciones de desarrollo software y en la segunda se realiza una discusión sobre los trabajos encontrados.

En el ciclo de investigación metodológico se crea el procedimiento propuesto. Éste consta de tres fases: 1) identificación del problema, 2) plan de acción, 3) recopilación de los datos. En la primera se determinan las herramientas que permiten establecer las causas del problema y los artefactos que intervienen en análisis causal. En la segunda se recopilan métodos, modelos y estándares relacionados con la temática investigada, con el fin de establecer las actividades

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

óptimas para el análisis causal en las VSEs y en la última fase se describe las herramientas y elementos apropiados para crear el procedimiento que puede ser aplicado en las pequeñas organizaciones.

El ciclo de investigación técnico está relacionado con el modelado del procedimiento, el cual se realizará empleando el estándar de representación BPMN [20].

### 2. Ciclo de solución del problema

Este ciclo se realiza mediante un caso de estudio que permite validar y mejorar el procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software. Para la solución del problema se aplican tres actividades:

**Diagnóstico:** se realiza el diseño del caso de estudio y la preparación para la recolección de datos.

**Acción:** se recolectan las evidencias

**Reflexión:** se efectúa el análisis de los datos recolectados.

### 1.5. Estructura del documento

Este trabajo de grado se compone de 5 capítulos. A continuación se presenta una descripción general de cada uno de ellos.

En el capítulo 2 (Marco Teórico y Estado del Arte), se presenta las bases teóricas empleadas para el desarrollo de este trabajo de grado y los trabajos realizados en torno a la temática. Se muestra modelos y normas internacionales que proponen actividades para el análisis causal, así como las principales investigaciones realizadas en diferentes organizaciones y contextos.

En el capítulo 3 (Procedimiento de Análisis Causal), se presenta la definición y el desarrollo del procedimiento propuesto, para realizar análisis causal en las pequeñas organizaciones de desarrollos software. Se especifica con detalle las actividades, roles,

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

tareas y productos de trabajo involucrados y finalmente se muestra el modelado del procedimiento.

En el capítulo 4 (Caso de Estudio), se presenta la aplicación del procedimiento en un entorno académico, especificando las actividades de intervención realizadas, la recolección de datos y el análisis de los resultados obtenidos.

En el capítulo 5 (Conclusiones y Trabajos Futuros), se describe las conclusiones obtenidas al realizar este trabajo de grado y se propone algunas ideas para trabajos futuros.

# Capítulo 2. Marco teórico y trabajos relacionados

---

En este capítulo se presenta el marco teórico en el que se ubica esta investigación junto con las definiciones y los términos tratados, además del estado actual de propuestas similares a las que en este trabajo se proponen y finalmente se muestra una discusión sobre lo encontrado.

Se ha elegido como método de investigación el mapeo sistemático, ya que es importante tener en cuenta el incremento de los artículos, conferencias, libros y otros, relacionados con análisis causal. Este método permite evaluar un mayor número de documentos, mediante una visión general del área investigada, asimismo, es posible identificar los documentos publicados y los resultados obtenidos [21].

Atendiendo las recomendaciones dadas por Petersen et al. [21] se llevó a cabo el proceso de revisión de la literatura, donde inicialmente se hizo una selección de bibliotecas digitales con bases de datos electrónicas indexadas y después se definieron las preguntas de investigación, en las cuales se refleja los objetivos perseguidos en este trabajo. Mediante la revisión de las referencias presentadas en los artículos encontrados, fue posible identificar otros que proporcionaban información relevante sobre el tema.

## 2.1. Marco teórico

### 2.1.1. Análisis causal

El concepto clave en este trabajo de grado es el análisis causal, que se define como una práctica que permite a las organizaciones identificar las causas de los defectos y otros problemas, con el propósito de tomar medidas y evitar que se produzcan en el futuro [12]. Esta es la definición que está relacionada más fuertemente con el concepto de análisis causal que se tiene en este trabajo de grado.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Muchas de las personas intentan solucionar los problemas de forma rápida y dejan atrás los procesos y documentación establecida en las organizaciones [22], sin embargo seguir o definir procesos pueden evitar esfuerzos innecesarios, detectar problemas futuros y mejorar los diseños realizados. Ejecutar el análisis causal de forma sistemática contribuye en los siguientes aspectos:

- ❖ Detección temprana de defectos
- ❖ Reducción de defectos
- ❖ Mejora en la calidad
- ❖ Contribuye en la disminución del esfuerzo invertido en la corrección de defectos.

Otro de los conceptos importantes dentro del análisis causal es “defecto”, que se define como cualquier evento que provoca un incumplimiento en las necesidades del cliente [23]. Se ha adoptado esta definición, ya que esta investigación no solo va a tratar aquellos defectos que impiden el correcto funcionamiento del software, sino que también se va a tener en cuenta otro tipo de situación como problemas de comunicación, retrasos en las entregas, falta de capacitación, entre otros.

### **2.1.2. Esquema de clasificación ortogonal de defectos (ODC, Orthogonal Defect Classification)**

Es importante tener una taxonomía de defectos para conocer las características de éstos, debido a que pueden ocurrir confusiones durante la clasificación de los datos. Henry [24] plantea que la clasificación ODC proporciona información valiosa para los diferentes procesos en el ciclo del desarrollo del software.

Para este trabajo de grado se ha seleccionado la clasificación ortogonal de defectos, desarrollada inicialmente por IBM (International Business Machines Corp) [25]. Según Huang et al. [26] este tipo de clasificación tiene mayor influencia en comparación con otros tipos de taxonomías, es así como se ha implementado en diversos tipos de organizaciones industriales y gubernamentales. De igual manera, Gupta et al. [27] mencionan la existencia de dos principales esquemas de clasificación de defectos: el estándar de clasificación de la IEEE y el establecido por Hewlett-Packard (HP), sin embargo se caracterizan por su complejidad. Por ejemplo, la clasificación establecida por la IEEE cuenta con más de 140 atributos, lo cual dificulta su implementación. En este sentido, el esquema de clasificación ortogonal de defectos proporciona

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

retroalimentación durante el desarrollo del proyecto y contribuye en la mejora de la calidad del software, además es posible comprender el impacto de los defectos y detectar áreas problemáticas [26].

La clasificación ODC consta de dos etapas denominadas apertura y cierre, en donde cada una de ellas tiene unos atributos (Tabla 1). En la primera etapa se detecta el momento en que se incorporó el defecto, lo cual permite realizar un registro y hacer el seguimiento; la segunda etapa se ejecuta cuando el defecto es corregido.

Etapa de apertura		Etapa de cierre	
Atributo	Descripción	Atributo	Descripción
<b>Actividad</b>	Actividad que se estaba realizando al encontrar el defecto	<b>Origen</b>	se refiere a que entidad debe ser corregida (la ubicación del defecto)
<b>Disparador</b>	Se refiere a como se detecta el defecto	<b>Tipo de defecto</b>	Se refiere a que debe ser corregido
<b>Impacto para el cliente</b>	se refiere al impacto que tiene el defecto sobre el cliente	<b>Calificador</b>	Explica cómo debe ser corregido el defecto que se encontró
		<b>Fuente</b>	Captura el origen del código que tenía el defecto (Target)
		<b>Edad</b>	Se refiere al historial del origen

Tabla 1. Atributos de la clasificación ODC

IBM ha desarrollado una guía que contiene una explicación detallada de cada una de los atributos definidos en el esquema ODC [28], esta guía da ejemplos sobre los tipos de defectos de acuerdo a los diferentes atributos establecidos en este esquema. Asimismo, Gupta et al. [27] sugieren ejemplos claros sobre algunos tipos de defectos presentados en el desarrollo de proyectos software, tal como se muestra a continuación:

- ❖ Implementación
  - algoritmo,
  - funcionalidad
  - rendimiento,
  - dificultades con el lenguaje

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- ❖ Defectos de interfaz
  - protocolo de comunicación
  - coordinación de procesos
  - interacciones inesperadas
  - coordinación en los cambios
  
- ❖ Defectos externos
  - entorno de desarrollo
  - trabajo concurrente).

### 2.1.3. Pequeñas organizaciones VSEs

Según la norma ISO/IEC 29110 [29] una pequeña organización es una empresa, organización o departamento que cuenta con un máximo de 25 empleados. A continuación se muestra las características e importancia de éstas.

#### 2.1.3.1. Características de las pequeñas organizaciones

Por lo que se refiere a la norma ISO/IEC 29110 [29], las pequeñas organizaciones tienen dificultad para adaptarse y acoger normas internacionales, como la ISO/IEC 15504 e ISO/IEC 12207. Esto se debe a que la mayoría no cuentan con el tiempo, los recursos económicos y humanos para implementar los procesos descritos en la reglamentación. Sin embargo, existen estándares apropiados como la norma ISO/IEC 29110 que ha sido desarrollada de acuerdo al contexto VSEs [30], teniendo en cuenta sus limitaciones y características.

La característica principal de las pequeñas organizaciones es su tamaño, pero existen otras diferencias, entre ellas: 1) una estructura organizacional plana, pocos recursos económicos, 2) se espera retorno de la inversión a corto plazo, 3) pequeño número de empleados, 4) orientado al desarrollo a la medida y 5) poca planificación en los proyectos ejecutados [29].

Asimismo, Habra et al. [3] establecen una serie de características para las pequeñas organizaciones, las cuales son mostradas a continuación:

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- ❖ Algunas disciplinas del ciclo de vida del software no están completamente formalizadas, lo que genera que este disminuya de forma significativa. En ocasiones, se da mayor importancia al desarrollo y al ensayo, sin tener en cuenta el valor que tienen las pruebas en la elaboración del software. Lo anterior, se podría explicar por la falta de recursos y la pretensión de cumplir con el tiempo de entrega establecido.
- ❖ Los niveles de madurez, dentro de los procesos de la organización se combinan, es decir, buenos procesos pueden mezclarse con prácticas de bajo nivel.
- ❖ Los procedimientos de control de calidad tienen poca formalidad.
- ❖ Los recursos destinados a la formación del personal son por lo general muy limitados, debido a restricciones económicas. Las actividades de capacitación se llevan a cabo a menudo en respuesta a deficiencias técnicas específicas.
- ❖ Las pequeñas organizaciones por lo general no implementan prácticas de aprendizaje y gestión del conocimiento.
- ❖ Debido a su tamaño, las VSEs tienen dificultades para imponer un enfoque metodológico.
- ❖ Los problemas de calidad no se abordan de forma explícita, mediante la gestión de calidad.

En este sentido, las pequeñas organizaciones de software deben tener en cuenta la calidad de sus productos, dada la competitividad en el mercado, que las obliga a mejorar sus procesos y a mantener su organización en constante crecimiento [31].

### **2.1.3.1. Importancia de las pequeñas organizaciones**

La industria del software es una actividad relevante para la economía mundial [32], por lo tanto, es importante desarrollar propuestas con el objetivo de mejorar los procesos llevados a cabo y aumentar su competitividad a nivel nacional e internacional. En este sentido, se puede ver como la industria del software colombiana ha realizado esfuerzos para mejorar y crear nuevas empresas, sin embargo son pocos los estándares de calidad y las metodologías de desarrollo implementadas [33]. De esto se puede ver



## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

como procedimientos como el propuesto en este trabajo de grado pueden ser útiles para el aumento de la calidad de los productos.

Ahora bien, la industria del software colombiana según el número de empleados está conformada en un 44% por microempresas, esto es de 1 a 9 empleados, un 44% pequeñas empresas, es decir de 10 a 49 empleados y en un 12% medianas y grandes empresas con más de 200 empleados [34]. Teniendo en cuenta que la industria de software colombiana se caracteriza por estar conformada por micro y pequeñas empresas (88%) y considerando los serios problemas de madurez que tienen en sus procesos, lo cual conduce a problemas que afectan a toda la empresa [32], se hace necesario definir un procedimiento que guíe de forma sistemática el hallazgo de las causas de los defectos, con el fin de que éstas lo puedan aplicar, sin hacer grandes inversiones de dinero, tiempo y recursos.

Asimismo, estudios realizados por la Federación Colombiana de la Industria de Software y Tecnologías Informáticas Asociadas – FEDESOFTE [35] afirman que la industria de software colombiana creció un 27,27%, por lo cual se llegó a facturar 4,2 billones de pesos en ingresos operacionales para el año 2012, también hubo un aumento del 34,2% en exportaciones entre el 2011 y el 2012, lo cual representa 139,7 millones de dólares. Estas cifras permiten evidenciar el desarrollo de la industria del software como una posibilidad de mejorar la economía nacional, brindando trabajo y bienestar a la sociedad.

En primer lugar, el estudio realizado [35] demuestra que el sector software requiere aproximadamente 44.829 personas calificadas, lo cual demuestra el empleo generado y su contribución en el desarrollo del país. En segundo lugar, se muestra el potencial del país en el sector informático y su crecimiento en un corto plazo, dando la oportunidad a Colombia de convertirse en un competidor a nivel internacional.

### **2.1.4. Business Process Modeling Notation (BPMN) [20]**

Es un estándar desarrollado por Object Management Group (OMG), con el fin de proporcionar una notación comprensible para todos los usuarios del negocio: los analistas de negocio encargados del diseño inicial de los procesos, los desarrolladores técnicos responsables de aplicar la tecnología con la que se llevará a cabo el proceso y los gerentes de negocio. Por lo tanto, BPMN crea un puente estandarizado para la brecha entre el diseño de procesos de negocio y la implementación de procesos [20].

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

El aporte tecnológico de este trabajo de grado consiste en el modelado del procedimiento propuesto, para esta labor se hizo uso del estándar BPMN, los elementos empleados para la construcción del modelo y su significado son mostrados a continuación:








Elemento	Descripción	Gráfica
Evento	Un evento es algo que sucede en el curso de un proceso de negocio. Hay tres tipos de eventos según el momento en que afecten el flujo del proceso: inicio, intermedio y fin.	 <p>Inicio Intermedio Fin</p>
Actividad	Es un término genérico para representar el trabajo realizado en una empresa. Hay dos tipos de actividades: tareas (unidades atómicas) y sub-procesos (unidades compuestas).	 <p>tarea Sub-proceso</p>
Gateway	Se usan para controlar la interacción de los flujos de secuencia dependiendo de su divergencia o convergencia dentro del proceso. Define las decisiones para mezclar o unir caminos.	
Secuencia	Indica el orden en que las actividades se realizaran en el proceso.	
Asociación	Se utiliza para indicar entradas y salidas de las actividades.	
Pool	Es la representación de un participante en un proceso, también puede ser empleado para dividir un conjunto de actividades.	
Lane	Se emplea para organizar y categorizar actividades.	

Tabla 2. Elementos de la notación BPMN

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### **2.1.5. Capability Maturity Model Integration (CMMI-DEV 1.3)**

Capability Maturity Model Integration [7], es un modelo que contiene prácticas recomendadas y que ayuda a las organizaciones a mejorar sus procesos. Dentro de este modelo se encuentra CMMI-DEV que proporciona un conjunto de guías para desarrollar productos y servicios. También incluye 22 áreas entre las que se tiene Análisis Causal y Resolución (CAR), que en la representación escalonada de este modelo, se encuentra en el nivel 5 dentro de la categoría soporte del proceso.

El área de análisis causal y resolución tiene como propósito identificar las causas de los resultados seleccionados y actuar para mejorar el rendimiento del proceso. Dentro de esta área se definen las actividades mostradas a continuación: 1) Identificar y analizar las causas de los resultados seleccionados, 2) eliminar las causas y prevenirlas, 3) analizar los datos para identificar problemas, 4) incorporar las causas de éxitos con el fin de mejorar el proceso.

A pesar de que este modelo da ejemplos de productos de trabajo y subprácticas que se deben llevar a cabo en cada actividad, no proporciona un nivel de detalle profundo que oriente a las pequeñas organizaciones en la ejecución de análisis causal.

### **2.1.6. ISO/IEC 12207 –Ingeniería del software y sistemas- Procesos del ciclo de vida del software**

La norma ISO/IEC 12207 [8] es el estándar definido para el proceso del ciclo de vida del software. Tiene como objetivo proporcionar un modelo común para los compradores, proveedores, desarrolladores, personal de mantenimiento, operadores, gestores y técnicos. Esta norma estructura los procesos de la organización por actividades, que a su vez se implementan a través de tareas que pueden ser aplicadas durante el suministro, desarrollo y mantenimiento del software.

Dentro de los procesos definidos en esta norma se tiene la resolución de problemas, cuyo propósito es asegurar que todos los problemas descubiertos sean analizados, gestionados y controlados. Dentro de este proceso se definen las actividades mostradas a continuación: 1) Implementación del proceso y 2) resolución del problema.

La primera actividad se compone de cuatro tareas 1) detectar los problemas, notificar su existencia a las partes interesadas, identificar las causas, analizarlas y eliminarlas, 2)

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

clasificar y priorizar los problemas, 3) detectar tendencias en los problemas detectados, 4) identificar problemas que han sido resueltos, los cambios aplicados y determinar si se ha introducido problemas adicionales. En la segunda actividad se debe realizar un informe para describir cada problema detectado.

### **2.1.7. ISO/IEC 15504-5 modelo de evaluación de procesos del ciclo de vida del software**

La ISO/IEC 15504-5 [9] es un modelo para la mejora y evaluación de los procesos de desarrollo, mantenimiento de sistemas de información y productos software. Proporciona una serie de prácticas genéricas que pertenecen a unos indicadores de capacidad del software. Esta norma también brinda un conjunto de características con las que deben cumplir las guías y productos de trabajo.

Dentro de los procesos definidos en esta norma se tiene la resolución de problemas software, cuyo propósito es identificar, analizar, gestionar y controlar los problemas detectados. Este proceso define las prácticas mostradas a continuación: 1) desarrollar una estrategia para la resolución de problemas, 2) identificar y registrar el problema, 3) clasificar los problemas, 4) Investigar y diagnosticar la causa de los problemas, 5) evaluar el impacto del problema y determinar su solución, 6) ejecutar las soluciones propuestas de mayor importancia, 7) levantar las notificaciones de alerta, 8) implementar las acciones de resolución de problemas, 9) Iniciar la solicitud de cambio, 10) seguimiento de cierre al estado de los problemas.

## **2.2. Trabajos relacionados**

### **2.2.1. Herramientas que apoyan el análisis causal**

En la revisión sistemática presentada por Arreche and Matalonga [14] se puede apreciar una recopilación de herramientas, que han sido usadas en las organizaciones desarrolladoras de software para ejecutar análisis causal. Éstas han sido clasificadas en la siguiente taxonomía: tradicionales, no tradicionales y solución de problemas. Las primeras son aquellas técnicas que tienen una amplia aceptación en las organizaciones de software, las segundas, son presentadas como una alternativa a las técnicas no tradicionales y las últimas son empleadas en otras áreas; sin embargo, permiten realizar análisis causal.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

A partir de la clasificación mencionada, surge un total de 14 técnicas óptimas para realizar análisis causal de defectos. De estas 14, 5 son utilizadas para la mejora de procesos software, 6 son utilizadas en el campo de la Inteligencia artificial y las restantes son aplicadas en un área diferente a la ingeniería del software. Finalmente el documento muestra las técnicas empleadas en la mejora de procesos software: Diagramas Causa-Efecto, Mapas mentales, pensamiento sistemático, análisis causa raíz y análisis de grafico radiales.

Ahora bien, Lehtinen et al. [36] se enfocan en analizar herramientas que soportan el análisis causal en grupos distribuidos de desarrollo software, debido a que esta modalidad cada día coge mayor fuerza y se ha convertido en una manera de trabajar con personas dispersas geográficamente. Por consiguiente, se hizo un estudio sobre las herramientas empleadas para el análisis causal en grupos que trabajan cara a cara, dando como resultado que éstas no son apropiadas para equipos distribuidos. A pesar que se ha afirmado que herramientas como correos electrónicos, hojas de cálculo y puentes de audio facilitan la retroalimentación en esta modalidad de trabajo, éstos no permiten organizar e interpretar un gran número de causas. Por esta razón, el trabajo desarrolla una herramienta basada en la nube que se caracteriza por tener una vista actualizada en tiempo real sobre la retroalimentación que se da en el equipo, además de proporcionar funciones para la creación de diagramas de Ishikawa, el desarrollo de ideas de mejora, obtenidas a partir de votos de los participantes y el apoyo del aprendizaje organizacional.

### 2.2.2. Métodos

El trabajo presentado por Card [17] muestra un método de análisis causal de defectos, diseñado para las organizaciones desarrolladoras de software. En éste se expone un proceso para ejecutarlo, el cual está conformado por seis fases: seleccionar problemas de muestra, clasificar los problemas seleccionados, identificar errores sistemáticos, determinar la causa principal, elaborar propuestas de acción y documentar los resultados obtenidos. En este sentido, se realizó una evaluación de los beneficios del método propuesto, durante un período de dos años en dos organizaciones diferentes: IBM y Computer Sciences Corporation. Al cabo de este tiempo se determinó que la tasa de defectos disminuyó en promedio en un 50% en ambas organizaciones. Es importante resaltar que durante estos dos años se presentó la tasa más baja de

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

defectos entre proyectos similares ejecutados anteriormente, asimismo fue posible eliminar los errores graves en el proyecto que implementó análisis causal de defectos.

Por otro lado, el autor menciona que existen otros enfoques relacionados con análisis causal; sin embargo, éstos efectúan esquemas de clasificación de defectos exhaustivos o modelos matemáticos complejos.

Dentro de esta perspectiva, el trabajo presentado por Lehtinen et al. [6] muestra un método ligero de Análisis de causa raíz, enfocado en las organizaciones productoras de software de tamaño medio, éste consta de cuatro pasos: detección del problema objetivo, detección de la causa raíz, acciones correctivas y documentación de los resultados. Entre las propuestas innovadoras de este método se encuentra una reunión de focus group que permite tratar un mayor número de problemas simultáneamente, disminuyendo el esfuerzo requerido en la etapa inicial de análisis causal, debido a que no es necesario definir esquemas de clasificación de defectos y/o causas. Además el trabajo propone un diagrama causa-efecto en forma de grafo dirigido, lo cual hace que una causa de problemas no se repita varias veces en el diagrama.

Gracias a los estudios realizados en cuatro organizaciones finlandesas, se demostró la facilidad, utilidad y calidad del método. Dentro del conjunto de evidencias expuestas en el artículo [6] se encuentran entrevistas y formularios en donde se refleja la satisfacción del personal encargado del análisis causal. Asimismo se comprobó que el esfuerzo requerido es adecuado en relación a los resultados obtenidos.

Dentro de este orden de ideas, Pankaj [37] muestra un método enfocado en prevenir defectos en proyectos de desarrollo software, que manejan un modelo iterativo e incremental. Se analiza los defectos en una iteración con la intención de que estos no se sigan presentando. Este análisis consta de los siguientes seis pasos: 1) clasificar los defectos encontrados, 2) identificar los tipos más comunes de defectos, 3) realizar un análisis causal y priorizar las causas fundamentales, 4) identificar y desarrollar soluciones para las causas fundamentales, 5) implementar la solución 6) revisar los beneficios de la prevención de defectos al final de la próxima iteración. En relación a la elaboración de análisis de defectos y prevención, se sugiere que en la fase de planificación del proyecto también se tenga en cuenta actividades relacionadas con la prevención de defectos tales como: 1) seleccionar el equipo encargado de la prevención de defectos, 2) tener una reunión inicial e identificar las soluciones

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

existentes, 3) establecer los objetivos de prevención de defectos, 4) conseguir un equipo entrenado en análisis causal y prevención de defectos.

El método ha sido implementado en varios proyectos, pero en el documento solo se expresa con detalle los beneficios obtenidos en uno de ellos. Según los autores la tasa de defectos del proyecto en que se aplicó el método, disminuyó notablemente. Asimismo fue posible reducir el esfuerzo invertido en la corrección de errores, y la retroalimentación realizada en las diferentes iteraciones ayudó a mejorar la calidad del proyecto.

Por otro lado, Nelms [16] muestra un método de análisis causal que se enfoca en comprender por qué el análisis de causa raíz se ha convertido en un problema. Según lo planteado no se tiene una definición unificada sobre lo que es análisis de causa raíz, por este motivo se maneja diferentes significados en las industrias e incluso puede haber múltiples puntos de vista dentro de la misma organización. También se considera que las personas son la principal causa de problemas, es por ello que el autor orienta su método de investigación hacia éstas, con el fin de que puedan identificar su papel dentro de las cosas que no están marchando bien. El método propuesto incluye preguntas que ayuda a los Stakeholders (personas cuyo comportamiento debe cambiar como resultado de un incidente) a darse cuenta de las acciones que han realizado y que han llevado a resultados no deseados.

Finalmente, El artículo presentado por Jabrouni et al. [38] muestra un enfoque de experiencias de retroalimentación para la formalización de procesos en la industria del transporte. Lo anterior utiliza los principios de análisis de causa raíz para identificar las causas de los problemas, también hace uso lecciones aprendidas en proyectos anteriores. Lo innovador en esta propuesta es el planteamiento de un modelo que consta de tres capas: operacional, semántica y de creencias.

En este trabajo se ha diseñado una capa operacional que usa análisis de causa raíz para identificar, corregir y eliminar la causa de los problemas. Se dan ejemplos sobre herramientas, entre las cuales se ha seleccionado la técnica de los 5 por qué, debido a su eficiencia y facilidad de uso, permitiendo llegar al origen del problema. La investigación emplea un gráfico en forma de árbol con las causas que están generando fallas, al mismo tiempo que se realizan 5 iteraciones a partir de la técnica de los 5 por qué.

# Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

## 2.2.3. Técnicas

El trabajo presentado por Honda [39] muestra una técnica de análisis de causa raíz y procedimiento 1+n, la cual permite encontrar defectos relacionados con otros encontrados previamente. Esta técnica hace parte de Quality Accounting System, que fue propuesta por el grupo NEC alrededor de 1982.

Para el análisis de la causa raíz de defectos y procedimiento 1+n, el “why why analysis” ha sido incluido y mejorado con el fin de obtener buenos resultados en la detección de defectos. Esta técnica es diseñada bajo la suposición de procesos de desarrollo en forma de V para establecer una relación entre las actividades y los tipos de causas considerados en el artículo. Adicionalmente el trabajo presenta un ejemplo de la aplicación del método. Sin embargo, esta técnica no es adecuada para resolver un gran número de problemas y puede tomar bastante tiempo, por lo que se recomienda aplicarla solo en hechos importantes.

En este trabajo se ha diseñado una capa operacional que usa análisis de causa raíz para identificar, corregir y eliminar la causa de los problemas. Se dan ejemplos sobre herramientas, entre las cuales se ha seleccionado la técnica de los 5 por qué, debido a su eficiencia y facilidad de uso, permitiendo llegar al origen del problema. La investigación emplea un gráfico en forma de árbol con las causas que están generando fallas, al mismo tiempo que se realizan 5 iteraciones a partir de la técnica de los 5 por qué.

## 2.3. Discusión

En la Tabla 3 se muestra la comparación de los documentos revisados previamente. En esta comparación también se incluye las características básicas del procedimiento a desarrollar en esta propuesta. Para este propósito se va a comparar cada una de las fases planteadas en el análisis causal. Según lo revisado en la literatura, la mayoría de los métodos propuestos incluyen las siguientes fases de desarrollo: 1) detección del problema objetivo, 2) Detección de causas fundamental, 3) acciones correctivas, 4) documentación de los resultados. De acuerdo a estas fases se va a realizar la tabla comparativa, también se va a tener en cuenta el tipo de propuesta, el nivel de descripción, si ha sido aplicado en pequeñas organizaciones y si da sugerencias sobre herramientas para realizar análisis causal.



## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Los artículos presentados en la Tabla 3 se enfocan en las grandes o medianas organizaciones, sin tener en cuenta las necesidades y características de las pequeñas. Por otra parte, las normas internacionales y recomendaciones expuestas en este trabajo, especifican qué se debe hacer, pero no cómo se debe hacer. Es importante saber qué se debe hacer, debido a que las pequeñas organizaciones no tienen suficientes recursos para aplicar análisis causal. Como se puede ver en la Tabla 3 la mayoría de documentos presentan un nivel de descripción bajo, lo cual dificulta aplicar análisis causal en las pequeñas organizaciones. El establecer un procedimiento que permita describir el “cómo” debe llevarse a cabo, cada una de las prácticas requeridas para ejecutar análisis causal, es de ayuda para las pequeñas organizaciones, ya que esto puede suplir parcialmente la necesidad de personal especializado, cabe resaltar que el procedimiento que se construyó en este trabajo de grado, está descrito de forma detallada y permite que las pequeñas organizaciones puedan aplicarlo de manera sistemática. Dentro de esta perspectiva el procedimiento se va a crear en el contexto de las VSEs en Colombia

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Tabla 3. Comparación de trabajos relacionados con análisis causal

Propuesta para análisis causal	Tipo de Propuesta	Nivel de descripción	Contexto en el que ha sido creada (Geográfico /Organizacional)	Sugerencias sobre herramientas <sup>7</sup> para el análisis causal		Adecuado para VSEs	Prácticas de análisis causal				
							Detección de problema objetivo	Detección de causas fundamentales	Acciones correctivas	Documentar resultados	Otras prácticas
[14]	Revisión sistemática	Bajo	Uruguay/	SI	Diagrama Causa-Efecto, mapas mentales, pensamiento sistemático, análisis causa raíz, análisis de gráficos radiales	NO	NO	NO	NO	NO	NO
[36]	Estudio de herramientas	Bajo	Finlandia/ empresas de tamaño medio	SI	Herramienta ARCA	NO	NO	NO	NO	NO	NO
[17]	Método	Bajo	/Grandes empresas	SI	Diagrama de Pareto, diagrama Causa-Efecto, entrevista	NO	SI	SI	SI	SI	NO
[6]	Método	Medio	Finlandia/ empresas de tamaño medio	SI	Reuniones de focus group, diagrama Causa-Efecto, talleres, técnica brainstorming	NO	SI	SI	SI	SI	NO
[16]	Método	Bajo	Estados Unidos/	SI	Entrevista	NO	NO	NO	NO	NO	SI
[37]	Método	Bajo	India/	SI	Diagrama de Pareto, gráfico de barras, gráfico de líneas, diagrama Causa-Efecto, entrevista, talleres	NO	SI	SI	SI	NO	NO
[11]	Caso de estudio	Bajo	N.D <sup>8</sup>	SI	Talleres	NO	SI	SI	SI	NO	NO
[39]	Técnica	Bajo	Japón/	SI	Why why analysis	NO	SI	SI	NO	NO	NO

<sup>7</sup> En este contexto se hace referencia a las herramientas usadas para facilitar la realización de análisis causal

<sup>8</sup> No se presentan datos

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

[38]	Método	Bajo	Francia/	NO	cinco (5) por qués	NO	NO	SI	NO	NO	NO
propues- ta de tesis	Procedimiento	Alto	Colombia/	SI	Diagrama de Ishikawa, diagrama de Pareto, mapas causales, gráficos radiales, diagrama de afinidad, reuniones y observación no participante.	SI	SI	SI	SI	SI	SI

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### **2.4. Aporte**

De los documentos analizados para elaborar este trabajo de grado, se puede inferir que desde nuestros conocimientos no se encuentra disponible un procedimiento de análisis causal, adaptado a las necesidades y características de las pequeñas organizaciones de desarrollo de software. Es cierto que existen diferentes maneras para realizar análisis causal, sin embargo los métodos propuestos no cuentan con el nivel de detalle suficiente para que las VSEs puedan aplicarlo. Por lo general las propuestas de análisis causal son desarrolladas teniendo en cuenta las características de grandes organizaciones.

El aporte investigativo que da este trabajo, está en hallar las mejores prácticas y las herramientas adecuadas para realizar análisis causal en las pequeñas organizaciones. Asimismo, el procedimiento que se construyó pretende guiar a las VSEs en cómo llevar el análisis causal de manera sistemática, guiando paso a paso la elaboración de los artefactos o productos de trabajo necesarios para este fin. Además se dio un aporte tecnológico mediante el modelado del procedimiento propuesto.

# **Capítulo 3. Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software**

---

Este capítulo contiene el procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software junto con los pasos seguidos para su construcción. En primer lugar, se muestra la comparación entre referentes internacionales CMMI DEV 1.3 [7], ISO/IEC 12207 [8] e ISO/IEC 15504 [9] que tienen en cuenta actividades relacionadas con el análisis causal. En segundo lugar, se analizan y comparan diferentes técnicas empleadas dentro del análisis causal, esto con el fin de seleccionar aquellas de mayor utilidad para el procedimiento propuesto y finalmente se expone el procedimiento con sus actividades, tareas, roles y productos de trabajo. En <http://artemisa.unicauca.edu.co/~kyzuniga/index.html> se muestra el procedimiento modelado con su respectiva documentación.

## **3.1. Comparación de referentes internacionales**

Se llevó a cabo una comparación con el fin de obtener las actividades del procedimiento propuesto en este trabajo de grado. Los referentes que se tuvieron en cuenta fueron: CMMI DEV 1.3 [7], ISO/IEC 12207 [8], e ISO/IEC 15504 [9], los cuales incluyen actividades o prácticas relacionadas con el análisis causal. Se realizó un listado teniendo en cuenta cada una de las actividades o prácticas definidas en los referentes internacionales. A partir de este listado se definió las actividades en común y relevantes para este trabajo.

A continuación se muestra los pasos ejecutados.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Primero se analizó la estructura de cada uno de los referentes internacionales, esto con el propósito de conocer su funcionamiento e identificar entidades de proceso en el mismo nivel de abstracción para ejecutar la comparación.

Referentes internacionales	Estructura
ISO/IEC 12207	<ul style="list-style-type: none"> <li>❖ Proceso                             <ul style="list-style-type: none"> <li>▪ Propósito                                     <ul style="list-style-type: none"> <li>▪ Outcomes   <ul style="list-style-type: none"> <li>▪ Actividades</li> <li>▪ Tareas</li> </ul> </li> </ul> </li> </ul> </li> </ul>
ISO/IEC 15504	<ul style="list-style-type: none"> <li>❖ Proceso                             <ul style="list-style-type: none"> <li>▪ Propósito                                     <ul style="list-style-type: none"> <li>▪ Outcome   <ul style="list-style-type: none"> <li>▪ Practicas</li> </ul> </li> </ul> </li> </ul> </li> </ul>
CMMI-DEV	<ul style="list-style-type: none"> <li>❖ Áreas de proceso                             <ul style="list-style-type: none"> <li>▪ Propósito                                     <ul style="list-style-type: none"> <li>▪ Metas específicas   <ul style="list-style-type: none"> <li>▪ Prácticas específicas   <ul style="list-style-type: none"> <li>▪ Ejemplo de productos de trabajo   <ul style="list-style-type: none"> <li>▪ Subprácticas   <ul style="list-style-type: none"> <li>▪ Ejemplos de técnica para ejecutar la subpráctica</li> </ul> </li> </ul> </li> </ul> </li> </ul> </li> </ul> </li> </ul> </li> </ul>

**Tabla 4. Estructura de los referentes internacionales**

En segundo lugar, se analizó las actividades y tareas definidas en la norma ISO/IEC 12207 (Tabla 5), así como las prácticas para la resolución de problemas software de la norma ISO/IEC 15504-5 (Tabla 6), además se consideró las metas, prácticas y subprácticas establecidas para el análisis causal en CMMI-DEV (Tabla 7).

ISO/IEC 12207	
Actividad	Tareas
Implementación de procesos	<p>Asegurarse de que todos los problemas detectados se notifiquen rápidamente y entrar en el proceso de resolución de problemas. Las partes interesadas son informadas de la existencia del problema, según corresponda; causas se identifican, analizan y, cuando sea posible se eliminan y los registros de los problemas se mantienen según lo estipulado en el contrato.</p> <p>El proceso debe contener un esquema para categorizar y priorizar los problemas. Cada problema debe ser clasificado por la categoría y prioridad para facilitar el análisis de tendencias y la resolución de problemas.</p>

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	Un análisis se realiza para detectar tendencias en los problemas reportados.
	Evaluar que los problemas se han resuelto, tendencias adversas se han invertido, y los cambios se han aplicado correctamente en los productos y actividades de software apropiados; y para determinar si se han introducido problemas adicionales.
Resolución de problemas	Cuando se han detectado problemas (incluyendo las no conformidades) en un producto de software o una actividad, se preparará un informe de problemas para describir cada problema detectado. El informe de problemas se utiliza a partir de la detección del problema a través de la investigación, el análisis y la resolución del problema y sus causas y en la detección de tendencias a través de problemas

**Tabla 5. Actividades y tareas para la resolución de problemas software-ISO/IEC 12207**

ISO/IEC 15504-5
Prácticas
Desarrollar una estrategia de solución de problemas
Identificar y registrar el problema
Brindar apoyo inicial y clasificar los problemas
Investigar y diagnosticar la causa del problema
Evaluar el impacto del problema y determinar la solución
Ejecutar acción de resolución urgente, en caso necesario
Elevar las notificaciones de alerta, cuando sea necesario
Implementar la resolución de problemas
Iniciar solicitud de cambio
seguimiento del estado de los problemas

**Tabla 6. Prácticas para la resolución de problemas software-ISO/IEC 15505-5**

CMMI-DEV		
Metas específicas	Prácticas específicas	Subprácticas
Determinar la causa de los defectos	Seleccionar los defectos y otros problemas para el análisis	Recoger los datos relevantes del defecto o del problema
		Determinar qué defectos y otros problemas serán analizados posteriormente
	Analizar las causas	Llevar a cabo el análisis causal con las personas responsables de realizar la tarea

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

		Analizar los defectos y otros problemas seleccionados para determinar sus causas raíz.
		Agrupar los defectos y otros problemas seleccionados basándose en sus causas raíz.
		Proponer y documentar las acciones que necesitan tomarse para prevenir la ocurrencia futura de defectos o de otros problemas similares
Tratar las causas de los defectos	Implementar las propuestas de acción	Analizar las propuestas de acción y determinar sus prioridades.
		Seleccionar las propuestas de acción que serán implementadas
		Crear elementos de acción para implementar las propuestas de acción.
		Identificar y eliminar defectos similares que pueden existir en otros procesos y productos de trabajo.
		Identificar y documentar las propuestas de mejora para el conjunto de procesos estándar de la organización
	Evaluar el efecto de los cambios sobre el rendimiento del proceso.	Medir el cambio en el rendimiento del proceso definido del proyecto según sea apropiado.
		Medir la capacidad del proceso definido del proyecto según sea apropiado
	Registrar los datos	Registrar los datos

**Tabla 7. Metas, prácticas y subprácticas para el análisis causal- CMMI-DEV**

Una vez analizados los referentes internacionales se elaboró un listado con las tareas definidas en la norma ISO/IEC 12207, las prácticas de la ISO/IEC 1504-5 y las subprácticas definidas en CMMI-DEV. Estas comparaciones se realizaron teniendo en cuenta la estructura de cada referente internacional y la relación existente entre ellos.



## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Pino et al. [40] plantean que las áreas de proceso de CMMI-DEV están estrechamente relacionadas con los procesos descritos en la norma ISO/IEC 15504-5 y la norma ISO/IEC 12207. Según el trabajo realizado [40] el análisis causal y resolución definido en CMMI-DEV y el proceso de resolución de problemas establecido en la ISO/IEC 12207 tienen una relación del 51% al 85%, lo cual indica que la mayoría de las prácticas específicas del análisis causal y resolución (CAR) están conectadas al proceso de resolución de problemas definido en la norma ISO/IEC 12207.

Actividad	ISO/IEC 12207	ISO/IEC 15504	CMMI DEV
Desarrollar una estrategia de solución de problemas		X	
Identificar y registrar el problema	X	X	
Recoger los datos relevantes del defecto o del problema			X
Informar a los interesados sobre la existencia del problema	X		
Determinar qué defectos y otros problemas serán analizados posteriormente	X		X
Clasificar los problemas	X	X	
Realizar un análisis para identificar tendencias en los problemas detectados	X		
Investigar y diagnosticar la causa del problema	X	X	
Llevar a cabo el análisis causal con las personas responsables de realizar la tarea			X
Analizar los defectos y otros problemas seleccionados para determinar sus causas raíz			X
Agrupar los defectos y otros problemas seleccionados basándose en sus causas raíz			X
Analizar las causas	X		X
Evaluar el impacto del problema y determinar la solución		X	
Proponer y documentar las acciones que necesitan tomarse para prevenir la ocurrencia futura de defectos o de otros problemas similares			X
Ejecutar acción de resolución urgente, en caso necesario		X	
Implementar la resolución de problemas	X	X	
evaluar que los problemas se han resuelto	X		
Elevar las notificaciones de alerta, cuando sea necesario		X	

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Iniciar solicitud de cambio		X	
Verificar los cambios introducidos en el software			
Evaluar el efecto de los cambios			X
Seguimiento del estado de los problemas		X	
Medir el cambio en el rendimiento del proceso definido del proyecto según sea apropiado			X
Medir la capacidad del proceso definido del proyecto según sea apropiado			X
Determinar si se han introducido problemas adicionales al hacer los cambios especificados	X		
Registrar los datos			X
Realizar un informe de problemas para describir cada problema detectado	X		

**Tabla 8. Listado de las actividades de los referentes internacionales**

Finalmente en la Tabla 9 se unificaron aquellas actividades comunes en los referentes internacionales y se obtuvo el siguiente resultado.

Actividad	ISO/IEC 12207	ISO/IEC 15504	CMMI DEV
Desarrollar una estrategia de solución de problemas		X	
Identificar y registrar los datos relevantes del defecto o del problema	X	X	X
Informar a los interesados sobre la existencia del problema	X		
Determinar qué defectos y otros problemas serán analizados posteriormente	X		X
Clasificar los problemas	X	X	
Realizar un análisis para identificar tendencias en los problemas detectados	X		
Determinar la causa de los problemas	X	X	X
Agrupar los defectos y otros problemas seleccionados basándose en sus causas raíz			X
Analizar las causas	X		X

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Evaluar el impacto del problema y determinar la solución	X		
Implementar las propuestas de acción	X	X	X
Elevar las notificaciones de alerta, cuando sea necesario		X	
Iniciar solicitud de cambio		X	
Verificar los cambios introducidos en el software		X	
Seguimiento del estado de los problemas	X	X	
determinar si se han introducido problemas adicionales al hacer los cambios especificados	X		
Realizar un informe de problemas para describir cada problema detectado	X		

**Tabla 9. Listado de las actividades de los referentes internacionales**

Teniendo en cuenta las actividades mostradas en la Tabla 9 y teniendo en cuenta aquellas actividades que son tenidas en cuenta en dos o más referentes y de acuerdo a su relevancia para este trabajo de grado se han seleccionado las actividades mostradas en la Tabla 10.

Actividad	Tareas
Detección de defectos	<ul style="list-style-type: none"> <li>❖ Identificar los defectos</li> <li>❖ Determinar muestra de los defectos</li> <li>❖ Clasificar los defectos</li> </ul>
Detección de causas fundamentales	<ul style="list-style-type: none"> <li>❖ Identificar las causas</li> <li>❖ Analizar las causas</li> <li>❖ Desarrollar recomendaciones</li> <li>❖ Iniciar solicitud de cambio</li> <li>❖ Divulgar al personal correspondiente</li> <li>❖ Hacer seguimiento a las recomendaciones</li> </ul>

Tabla 10. Actividades del procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

También se considera importante las actividades relacionadas con la capacitación del procedimiento y las técnicas recomendadas para apoyar las tareas, así como documentar las lecciones aprendidas durante el desarrollo del proyecto y almacenar los

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

documentos resultantes en cada actividad. Por estos motivos la actividad Preparación y Documentación han sido incluidas, debido a que capacitar al personal permite emplear el potencial del procedimiento y ejecutarlo de forma eficiente. Se debe agregar que para elegir un líder o a un grupo encargado de realizar determinado trabajo, éstos deben cumplir con unas habilidades específicas, es por ello que vale la pena adicionar tareas relacionadas con esta labor. Las tareas seleccionadas son mostradas en la Tabla 11.

Actividad	Tareas
Preparación	<ul style="list-style-type: none"><li>❖ Elección del líder de análisis causal</li><li>❖ Elección del líder del grupo de análisis causal</li><li>❖ Capacitación sobre la estructura y componentes del procedimiento</li><li>❖ Capacitación de técnicas</li></ul>
Documentar	<ul style="list-style-type: none"><li>❖ Documentar lecciones aprendidas</li><li>❖ Almacenar información obtenida en el procedimiento</li></ul>

Tabla 11. Actividades adicionales del procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### 3.2. Selección de técnicas para análisis causal

Para seleccionar las técnicas que apoyan el procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software, se hizo una revisión de la literatura con el fin de obtener aquellas que han sido implementadas en la ingeniería del software y en el análisis causal. Para cada una de la técnicas seleccionadas se analizó el esquema implementado, las herramientas software disponibles para elaborar los esquemas, una breve descripción, las ventajas, desventajas y el procedimiento para su elaboración.

Las técnicas analizadas se muestran a continuación: diagrama de Ishikawa, diagrama de Pareto, mapas mentales, pensamiento sistémico, mapas causales, mapa conceptual, modelo de ecuaciones estructurales, modelos gráficos, análisis de trayectoria o ruta, análisis de grafico radial, modelo basado en reglas, teoría relacional para mapas causales, diagrama de dispersión o de correlación, diagrama de afinidad, análisis modal de fallo y efectos.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Al mismo tiempo, la elección de las técnicas óptimas para el procedimiento propuesto se realizó mediante la evaluación de seis criterios: 1) Referencias encontradas de su aplicación dentro de la ingeniería del software, 2) Nivel de conocimientos requeridos para el uso de la técnicas, 3) Nivel de complejidad en la implementación del procedimiento, 4) Nivel de complejidad en la interpretación de esquema, 5) Herramientas de software libre disponibles y 6) nivel de conocimientos estadísticos. Éstos fueron seleccionados teniendo en cuenta las características de las pequeñas organizaciones, debido a que éstas no cuentan con los recursos suficientes para invertir en la compra de costosas licencias para programas software o en capacitación de personal. Además, es importante que las técnicas que se van a emplear en las pequeñas organizaciones sean de fácil implementación. Asimismo, se definió una escala que permitió evaluar de forma equitativa las técnicas encontradas y otras que se puedan considerar en el futuro.

El Anexo 2 muestra en detalle las fichas de las técnicas analizadas junto con los criterios de evaluación y la ponderación realizada, de la cual se obtuvo las técnicas que apoyan el procedimiento de análisis causal propuesto en este trabajo de grado. Cabe mencionar que éstas fueron seleccionadas teniendo en cuenta el estudio realizado para cada una de ellas y sugeridas en las actividades que se consideró apropiadas de acuerdo con las características de cada técnica (Tabla 12).

Técnica	Actividad
Diagramas de afinidad	Detección de defectos
Diagrama de Pareto	Detección de defectos
Mapas causales	Detección de causas fundamentales
Diagrama de Ishikawa	Detección de causas fundamentales
Análisis de grafico radial	Detección de causas fundamentales

Tabla 12. Relación técnica-actividad

A continuación se muestra una breve descripción de las herramientas seleccionadas:

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### ❖ Diagrama de Ishikawa

Esta técnica, además de ser ampliamente utilizada en la ejecución de análisis causal es de fácil implementación [41]. A continuación se presentan algunos trabajos que reconocen su utilidad: [17] [37] [42] [43] [44] [36] [41], además Arreche and Matalonga [14] aseguran que el diagrama de Ishikawa es la técnica con mayor uso en el análisis causal de defectos.

En este sentido, el procedimiento para realizar este diagrama es sencillo lo cual facilita su implementación en las VSEs, esto es importante debido a que éstas no cuentan con los recursos económicos y humanos suficientes para el uso de herramientas complejas. Además, al revisar el esquema planteado para el diagrama de Ishikawa, se puede notar que su representación visual es intuitiva, lo cual permite reconocer otras posibles causas o visualizar relaciones entre elementos que no se habían detectado en un principio. Como si fuera poco, para dibujar el diagrama de Ishikawa existen programas software que facilitan esta tarea y algunos de éstos son gratuitos, razón por la cual las pequeñas organizaciones desarrolladoras de software no tienen que invertir recursos en la compra de licencias.

### ❖ Diagrama de Pareto

El diagrama de Pareto al igual que el de Ishikawa ha sido una de las técnicas con mayor aceptación dentro del análisis causal. Esta ha sido implementada en varios trabajos, algunos son mostrados a continuación: [17][37] [43] [44][11] [18] [45].

La selección de este diagrama se ha dado porque muestra visualmente los principales tipos de defectos, identificándolos en un 80-85% [37]. Lo anterior, es de ayuda para el encargado del análisis causal, debido a que le permite centrarse en los defectos de mayor importancia. Asimismo el procedimiento para realizar el diagrama de Pareto es sencillo y existen programas software que contribuyen en esta tarea.

### ❖ Mapas causales

Según Bjørnson et al. [46] los mapas causales son adecuados en los grupos pequeños de desarrollo software, debido a que no se cuenta con personal experto en el uso de herramientas complejas. Además, con éstos es posible obtener un

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

mayor número de causas, de las cuales se seleccionan las principales creando largas cadenas de búsqueda (mayor profundidad), lo que ayuda a hacer caso omiso de aspectos irrelevantes. De esta manera se identifica posibles acciones de mejora.

En este sentido, el método desarrollado por Lehtinen et al. [6] destaca la importancia de esta técnica, ya que a diferencia de otras no permite la duplicación de las causas, debido a que un nodo puede ser el origen de otros nodos y solo se debe expresar la relación entre éstos.

### ❖ Diagramas de afinidad

Los diagramas de afinidad han sido considerados como una técnica aplicable a las VSEs, ya que mediante ésta las personas involucradas en el proyecto pueden interactuar de forma directa. Como se puede ver en el procedimiento planteado para esta técnica (Anexo 2) cada persona debe ayudar a formar el diagrama, dando como resultado ideas compartidas y discutidas entre todos los miembros del equipo. Además permite obtener conocimientos con pocos gastos económicos, lo cual es muy importante para las VSEs.

### ❖ Análisis de gráfico radial

El trabajo realizado por Henderson [47] muestra la utilidad de esta técnica dentro del análisis causal de defectos y la novedosa forma de uso, donde se logra representar gráficamente la relación entre la causa y los defectos producidos en las diferentes disciplinas del software.

Finalmente, el procedimiento propuesto también hace uso de técnicas como reuniones grupales y la observación, las cuales fueron empleadas por sugerencias de trabajos relacionados con el análisis causal y la utilidad observada en ellos [17] [18] [6] [48] [49][50] [12].

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### **3.3. Definición del procedimiento**

#### **3.3.1. Vista general de procedimiento**

La versión final del procedimiento de análisis causal para pequeñas organizaciones de desarrollo software, planteado en este trabajo de grado está conformado por una descripción del propósito, objetivo, diagrama de actividad, actividades, tareas, roles y productos de trabajo, como se muestra a continuación.

##### **3.3.1.1. Propósito**

El propósito del procedimiento de análisis causal para pequeñas organizaciones de desarrollo software, es detectar las causas de los defectos, con el fin de mejorar la calidad del producto y reducir sobrecostos.

##### **3.3.1.2. Objetivo**

Proponer un procedimiento de análisis causal que guie a las pequeñas organizaciones en la identificación de las causas que generan los defectos, estableciendo las técnicas adecuadas para aplicar análisis causal y teniendo como referente normas y modelos internacionales.

##### **3.3.1.3. Actividades y tareas**

El procedimiento planteado está conformado por las siguientes actividades: Preparación, Detección de defectos, Detección de causas fundamentales y Documentar. Para cada una de estas actividades se ha definido un conjunto de tareas, donde se especifica cómo llevar a cabo la actividad.

##### **3.3.1.4. Roles**

Para cada una de las tareas se ha definido un responsable, de acuerdo a sus competencias. Los roles involucrados en el procedimiento son mostrados a continuación.



## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Abreviatura	Rol
LA	Líder de análisis causal
GA	Grupo de análisis causal
GP	Gerente del proyecto
ME	Todos los miembros del equipo
EC	Encargado de la capacitación
SC	Solicitante del cambio

Tabla 13. Identificación de Roles involucrados

### 3.3.1.5. Productos de trabajo

Los productos de trabajo son el resultado de un conjunto de actividades relacionadas entre sí, las cuales se transforman en entradas o salidas. Éstos son mostrados a continuación: perfil del encargado de aplicar el procedimiento, perfiles de los miembros del grupo de análisis causal, plantilla de recolección de defectos diligenciada, plantilla de defectos de mayor impacto diligenciada, plantilla para clasificar los defectos diligenciada, plantilla de documento de defectos diligenciada, plantilla de causas encontradas diligenciada, plantilla de causas fundamentales diligenciada, plantilla de recomendaciones diligenciada, plantilla de solicitud de cambios diligenciada, Plantilla de seguimiento de recomendaciones, plantilla de documento de causas diligenciada y plantilla de lecciones aprendidas diligenciada. Las plantillas para cada uno de los productos de trabajo mencionados anteriormente se encuentran en el anexo 1.

### 3.3.1.6. Diagrama de actividades

La Figura 2 muestra todas las actividades que hacen parte del procedimiento propuesto en este trabajo de grado.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

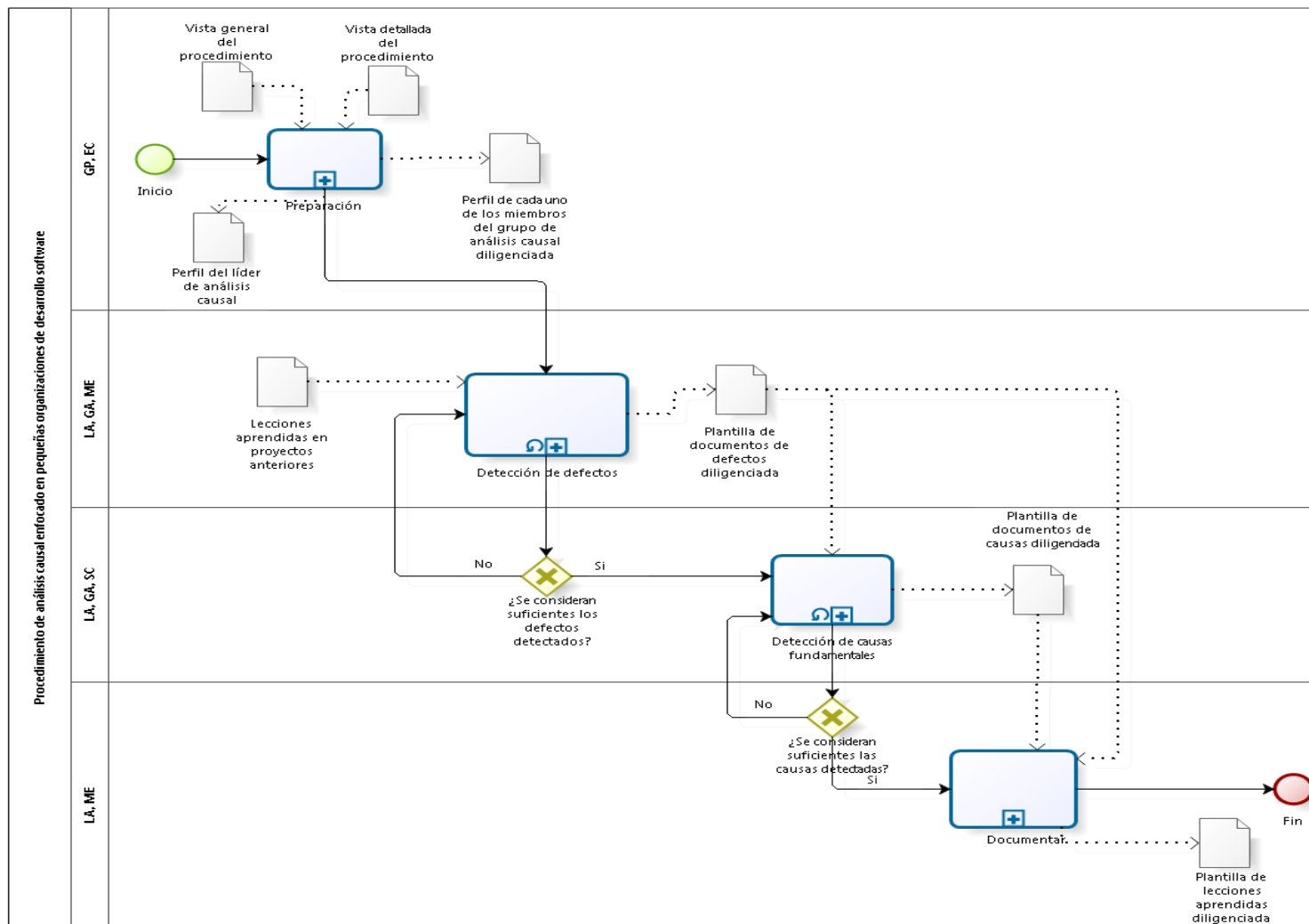


Figura 2. Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### 3.3.2. Vista detallada del procedimiento

#### 3.3.2.1. Descripción

El procedimiento que se plantea en este trabajo de grado, está conformado por actividades, tareas, roles y productos de trabajo. A continuación se presenta la vista detallada del procedimiento propuesto. Para este propósito se tomó como referencia el formato presentado en el proyecto COMPETISOFT [51].

<b>Actividad</b>	Preparación
<b>Propósito</b>	El propósito de la actividad Preparación es conformar el grupo responsable de ejecutar el procedimiento para realizar análisis causal. En primer lugar se debe seleccionar la persona que ocupará el rol de líder (LA) dentro del grupo. En segundo lugar se procede a seleccionar las personas que harán parte del grupo de análisis causal (No más de 8 personas) y finalmente se debe realizar una capacitación previa sobre el procedimiento (en caso de que el procedimiento se esté aplicando por primera vez), donde se deje claro el objetivo del procedimiento y su estructura, esto con el fin de obtener los resultados esperados.
<b>Descripción</b>	<p>La actividad Preparación se compone de las tareas mencionadas a continuación:</p> <ol style="list-style-type: none"><li>1. Elección del líder de análisis causal encargado de aplicar el procedimiento: para la elección del líder de análisis causal se debe tener en cuenta la cercanía del candidato con el proyecto. También es importante que la persona seleccionada sea un respetado miembro del equipo.</li><li>2. Elección del grupo de análisis causal: El grupo de análisis causal debe estar conformado con aquellos que entienden el resultado de los defectos a tratar, por este motivo se recomienda que este grupo este conformado por personas con conocimientos de las diferentes fases del proyecto.</li><li>3. Capacitación sobre la estructura y componentes del procedimiento: se debe dar un encuentro entre el líder de análisis causal y el gerente del proyecto, donde este último brinda la información necesaria al encargado de aplicar el método, para que comprenda para que se hace y como se debe hacer, asimismo se debe especificar los beneficios que obtendrá la organización mediante el procedimiento. El líder de análisis causal debe tener acceso a toda la documentación asociada al procedimiento (vista general del método y la información de aplicación). Asimismo es necesario motivar al grupo de análisis causal.</li><li>4. Capacitación de técnicas: formación en las técnicas que se van a utilizar para llevar a cabo el análisis causal. Los miembros del grupo de análisis causal deben capacitarse sobre el funcionamiento de la técnica de Ishikawa, diagramas de afinidad, mapas causales, gráficos radiales y diagrama de Pareto.</li></ol>

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<b>Objetivos</b>	<p>O1. Identificar y seleccionar la persona encargada de aplicar el procedimiento (Líder de análisis causal).</p> <p>O2. Identificar y seleccionar las personas que harán parte del grupo de análisis causal.</p> <p>O3. Capacitar al líder de análisis causal para la ejecución del procedimiento.</p> <p>O4. Capacitar al grupo de análisis causal en el funcionamiento de las técnicas empleadas para llevar a cabo el análisis causal.</p>	
<b>Responsable</b>	Gerente del proyecto	
<b>Entradas</b>		<b>Fuente</b>
Vista general del procedimiento		Gerente del proyecto
Vista detallada del procedimiento		Gerente del proyecto
<b>Salidas</b>		<b>Destino</b>
Perfil del encargado de aplicar el procedimiento		Documentar
Perfiles de los miembros del grupo de análisis causal		Documentar
<b>Productos Internos</b>		
No aplica		
<b>Tareas</b>		
<b>T1. Elección del líder de análisis causal encargado de aplicar el procedimiento</b>		
<b>Entradas</b>	Plantilla del perfil del líder de análisis causal	
<b>Rol</b>	<b>Descripción</b>	
GP	<p>T1.1 Elección del líder de análisis causal</p> <ul style="list-style-type: none"> <li>• Si la organización cuenta con un experto en la aplicación del procedimiento, éste asumirá el rol de líder de análisis causal</li> <li>• Si la organización no cuenta con un experto en la aplicación del procedimiento, se recomienda seleccionar a una persona que no ocupe puestos de gerencia o aseguramiento de la calidad, ya que estas personas pueden perseguir fines que sesgan las discusiones producidas en las reuniones. Algunas de las competencias que debe tener este rol son: <ul style="list-style-type: none"> <li>• Habilidad para construir reportes</li> <li>• Habilidad para identificar defectos a partir de entrevistas</li> <li>• Habilidad para escuchar atentamente</li> <li>• Dispuesto a aprender</li> <li>• Capacidad de comunicación con los miembros del equipo.</li> <li>• Conocimientos en los procesos de la organización y en el desarrollo del</li> </ul> </li> </ul>	

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<p>proyecto.</p> <ul style="list-style-type: none"> <li>• Escuchar activamente.</li> <li>• Conocimiento en técnicas de análisis causal</li> </ul>
<b>Salidas</b>	Plantilla del perfil del líder de análisis causal de la persona elegida para la aplicación del procedimiento.
<b>T2. Elección del grupo de análisis causal</b>	
<b>Entradas</b>	Plantilla del perfil del aspirante al grupo de análisis causal
<b>Rol</b>	<b>Descripción</b>
GP	<p>T2.1 Elección del grupo de análisis causal</p> <p>El GP se encarga de seleccionar a las personas que harán parte del grupo de análisis causal. Se debe seleccionar las personas que cuenten con la mayor cantidad de las siguientes competencias:</p> <ul style="list-style-type: none"> <li>• Conocimiento sobre el proceso y el producto</li> <li>• Habilidad para relacionarse con otras personas</li> <li>• Habilidad para expresar dificultades</li> <li>• Capacidad de comunicación con los miembros del equipo</li> <li>• Conocimiento en técnicas de análisis causal</li> </ul> <p>Por otra parte, se recomienda que el grupo esté formado por personas que trabajen en las diferentes disciplinas del ciclo del vida del software. Es importante que se tenga en cuenta la participación de los programadores y las personas encargadas del mantenimiento del software, ya que estos pueden hacer valiosos aportes en la detección de los defectos.</p>
<b>Salidas</b>	Plantilla diligenciada con el perfil de cada uno de los miembros del equipo seleccionados.
<b>T3. Capacitación sobre la estructura y componentes del procedimiento</b>	
<b>Entradas</b>	Vista general de procedimiento Vista específica del procedimiento
<b>Rol</b>	<b>Descripción</b>
GP	<p>T3.1 Charla</p> <p>Con el fin de motivar al encargado de ejecutar el procedimiento de análisis causal y al equipo, el gerente del proyecto dará una explicación breve de los beneficios obtenidos y los resultados esperados.</p>
GP	<p>T3.2 Entregar documentación para la capacitación</p> <p>EL GP debe entregar al LA toda la documentación relacionada con el procedimiento: vista general y vista específica del procedimiento. La primera vista debe contener el propósito, actividades, productos asociados, entre otros. La segunda vista debe contener todos los pasos, componentes y artefactos necesarios para la aplicación del procedimiento.</p>
Salidas	No aplica
<b>T4. Capacitación de técnicas</b>	
<b>Entradas</b>	Material de apoyo sobre las técnicas en estudio
<b>Rol</b>	<b>Descripción</b>

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<b>EC</b>	T4.1 Capacitación El encargado de la capacitación se debe organizar jornadas en las cuales el grupo de análisis causal recibe asesoría sobre el funcionamiento de las técnicas empleadas en el análisis causal: la técnica de Ishikawa, diagramas de afinidad, mapas causales, gráficos radiales y diagrama de Pareto.
<b>Salidas</b>	No aplica

Tabla 14. Definición específica de la actividad Preparación

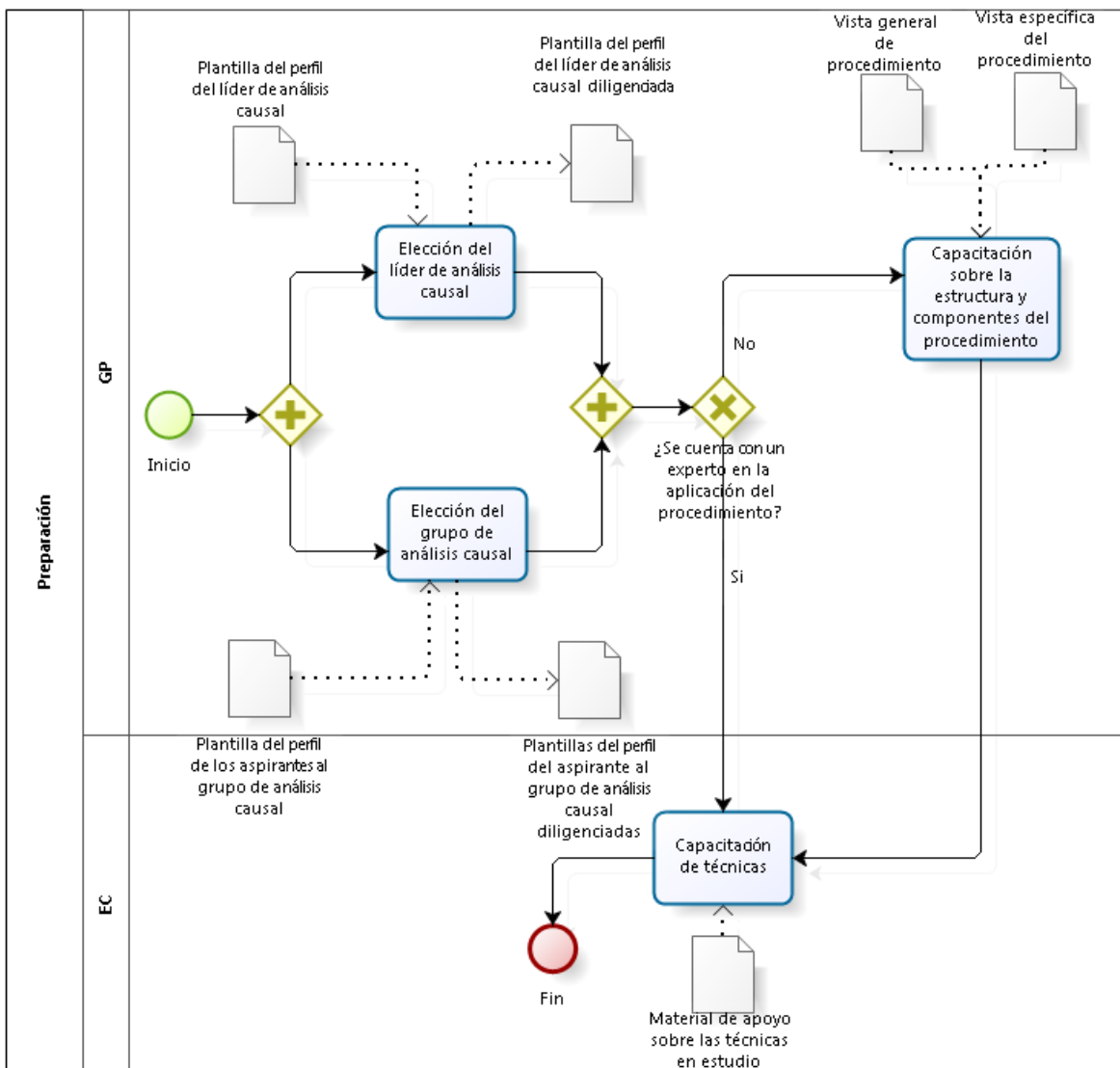


Figura 3. Diagrama de actividad Preparación

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<b>Actividad</b>	Detección de defectos	
<b>Propósito</b>	El propósito de la actividad <i>Detección de defectos</i> es discutir, analizar y clasificar los defectos.	
<b>Descripción</b>	<p>La actividad <i>Detección de defectos</i> se compone de las tareas mencionadas a continuación:</p> <ol style="list-style-type: none"> <li>1. Identificar los defectos: el grupo de análisis causal se debe encargar de discutir los defectos que han surgido en el desarrollo del proyecto. Para realizar esta tarea se deben apoyar en una serie de técnicas que facilitaran esta labor. Entre las técnicas recomendadas se tiene: reuniones grupales y diagramas de afinidad.</li> <li>2. Determinar muestra de los defectos: el grupo de análisis causal se encarga de seleccionar los defectos que causan mayor impacto en el desarrollo del proyecto. En una reunión grupal se debe priorizar los defectos empleando técnicas como diagramas de Pareto o Análisis modal de fallo y efectos (FMEA).</li> <li>3. Clasificar los defectos: el líder de análisis causal se encarga de analizar los defectos e identificar cuales están relacionados directamente con el producto software y cuales tienen mayor relación con el desarrollo del proyecto. De acuerdo a los tipos de defectos identificados se emplea el esquema de clasificación ortogonal de defectos (ODC, Orthogonal Defect Classification) o el esquema de clasificación de causas, formulado por Kaoru Ishikawa (Diagrama de Ishikawa o Diagrama Causa-Efecto).</li> <li>4. Preparar documento de defectos: se crea el documento que contiene la información obtenida en las tareas realizadas anteriormente.</li> </ol>	
<b>Objetivos</b>	<p>O1. Determinar los defectos que causan mayor impacto en el desarrollo del proyecto.</p> <p>O2. Clasificar los defectos, con el fin de facilitar la detección de las causas.</p>	
<b>Responsables</b>	Grupo de análisis causal	
<b>Entradas</b>		<b>Fuente</b>
Reporte de lecciones aprendidas en proyectos anteriores		Gerente del proyecto
<b>Salidas</b>		<b>Destino</b>
Plantilla de documento de defectos diligenciada		Detección de causas fundamentales, Documentar
<b>Productos Internos</b>		
Plantilla de recolección de defectos diligenciada		
Plantilla de defectos de mayor impacto diligenciada		
Plantilla para clasificar los defectos diligenciada		
<b>Tareas</b>		
<b>T1. Identificar los defectos</b>		
<b>Entradas</b>	Plantilla de recolección de defectos	

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Rol	Descripción
GA	<p><b>T1.1 Reuniones grupales</b></p> <p>El grupo de análisis causal debe reunirse para discutir los defectos evidenciados en el desarrollo del proyecto. Por ejemplo, defectos notificados por el cliente, defectos encontrados en las pruebas, medidas de productividad que son menores de lo esperado, informe de problemas de gestión de proyectos que requieren acción correctiva, problemas de capacitación de proceso, problemas de cumplimiento o de satisfacción del servicio, bajo rendimiento de los recursos, defectos evidenciados por el gestor del proyecto mediante observación. Estas reuniones deben ser rápidas con una duración máxima de 40 minutos.</p> <p>Para el desarrollo de esta reunión los miembros del grupo deben discutir los defectos evidenciados, utilizando la técnica diagramas de afinidad y siguiendo el procedimiento planteado en su respectiva ficha técnica (Anexo 2). Además el LA debe dar a conocer la lista de defectos que fueron detectados por otros miembros del equipo, mediante los correos y las entrevistas realizadas (T1.2, T1.3).</p> <p>En esta reunión se deben registrar los defectos detectados, el líder de análisis causal se encargará de anotar esta información (Ver plantilla).</p> <p>Se debe hacer dos reunión de inicio, en la primera se identifica los defectos que han surgido durante el desarrollo del proyecto y en la segunda se analiza sus causas y se decide las medidas que se van a implementar.</p> <p>Luego de las reuniones de inicio, estas deben ser repetidas al final de cada iteración, es decir cuando los defectos han sido eliminados. Adicionalmente en la primera reunión se debe informar al personal correspondiente las acciones implementadas para eliminar las causas de los defectos</p>
ME	<p><b>T1.2 Enviar correo electrónico</b></p> <p>Previo a la reunión grupal cada uno de los participantes del proyecto debe enviar un correo al líder de análisis causal. Este correo debe contener los defectos que ha evidenciado el remitente.</p>
GA	<p><b>T1.3 Realizar entrevistas:</b> algunos miembros del grupo de análisis causal se encargaran de diseñar y ejecutar entrevistas a los usuarios del producto, esto con el fin de detectar posibles defectos. El encargado de realizar estas preguntas debe tener en cuenta aspectos como:</p> <ul style="list-style-type: none"> <li>• Conformidad del cliente con el producto</li> <li>• Inconvenientes presentados con la empresa encargada del proyecto</li> <li>• Cumplimiento de plazos establecidos</li> <li>• Objetivos no cumplidos</li> <li>• Funcionalidad del software</li> </ul> <p>Las personas encargadas de realizar estas entrevistas, deben realizar un listado con los defectos encontrados, para luego enviarlos al líder de análisis causal.</p>
<b>salidas</b>	<p>Plantilla de recolección de defectos diligenciada</p> <p>Entrevistas diligenciada</p> <p>correos</p>



## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<b>T2. Determinar muestra de los defectos</b>	
<b>Entradas</b>	Plantilla de recolección de defectos diligenciada Plantilla de defectos de mayor impacto
<b>Rol</b>	<b>Descripción</b>
LA	<p>T2.1 Determinar defectos a ser analizados</p> <p>La organización puede emplear las siguientes técnicas para determinar la muestra de los defectos: Diagrama de Pareto o análisis modal de falla y efectos (FMEA). Los diagramas de Pareto permiten determinar los defectos presentados con mayor ocurrencia, mientras que FMEA permite realizar un análisis más riguroso, debido a que tiene en cuenta tres atributos: ocurrencia, gravedad y detección.</p> <p>T2.1.1 Realizar diagrama de Pareto</p> <p>A partir del reporte de los defectos detectados inicialmente, el líder de análisis causal debe realizar un diagrama de Pareto, con el objeto de identificar los problemas presentados con mayor frecuencia, lo cual da una idea de los problemas que deben ser abordados primero. El LA debe realizar este diagrama teniendo como referente los defectos detectados en las reuniones grupales (T1.1).</p> <p>T2.1.2 Análisis modal de falla y efectos</p> <p>A partir del reporte de los defectos detectados inicialmente, el líder de análisis causal debe realizar el análisis modal de fallas y efectos. Para realizar esta tarea debe tener en cuenta las escalas establecidas para cada criterio (Anexo 3)</p>
<b>Salidas</b>	Plantilla de defectos de mayor impacto diligenciada
<b>T3. Clasificar los defectos</b>	
<b>Entradas</b>	Plantilla de documento de defectos Plantilla de recolección de defectos diligenciada Plantilla de defectos de mayor impacto diligenciada Plantilla para clasificar los defectos diligenciada
<b>Rol</b>	<b>Descripción</b>
LA	<p>T3.1 Clasificar los defectos</p> <p>A partir del reporte que contiene los defectos que van a ser analizados, el LA debe identificar si el defecto está relacionado directamente con el producto software o con el proceso definido para el desarrollo del proyecto. A continuación se muestra lo que se debe realizar en cada situación.</p> <ul style="list-style-type: none"> <li>❖ Si el defecto está relacionado directamente con el producto software (el defecto es un cambio que se debe realizar para el correcto funcionamiento del software), se procede a realizar la clasificación de defectos mediante el esquema ODC (Orthogonal Defect Classification). La plantilla para clasificar los defectos tiene dos etapas: apertura y cierre. La etapa de apertura permite identificar el defecto, las circunstancias que lo provocaron y el impacto para el cliente. Los atributos que pertenecen a la etapa de cierre deben ser llenados cuando el defecto ha sido reparado y se deba anotar los</li> </ul>

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<p>cambios a realizar (por ejemplo: requisitos, diseño, código, entre otros) y el tipo de cambio que se debe hacer para solucionar el defecto (por ejemplo asignación, comprobación, algoritmo, entre otros) y otros atributos mostrados en la plantilla.</p> <p>El anexo 4, contiene una traducción libre de las partes fundamentales de la guía realizada por IBM, donde se encuentra una explicación detallada del funcionamiento del esquema ODC.</p> <ul style="list-style-type: none"> <li>❖ Si el defecto está relacionado con la ejecución del proyecto (por ejemplo retrasos en entregas, falta de trazabilidad en los artefactos generados durante el desarrollo del proyecto, ausencia de especificaciones técnicas, requerimientos no especificados, entre otros), se procede a realizar un diagrama de Ishikawa, ya que éste permite identificar las causas de cualquier tipo de defecto (Ver actividad identificar causas, técnica Diagrama de Ishikawa).</li> </ul>
<b>Salidas</b>	Plantilla para clasificar los defectos diligenciada.
<b>T4. Preparar documento de defectos</b>	
<b>Entradas</b>	Plantilla de documento de defectos
<b>Rol</b>	<b>Descripción</b>
LA	T4.1 Crear documento
	Elaborar un documento que contenga la información recolectada en las anteriores tareas
<b>Salidas</b>	Plantilla de documento de defectos diligenciada

Tabla 15. Definición específica de la actividad Detección de defectos

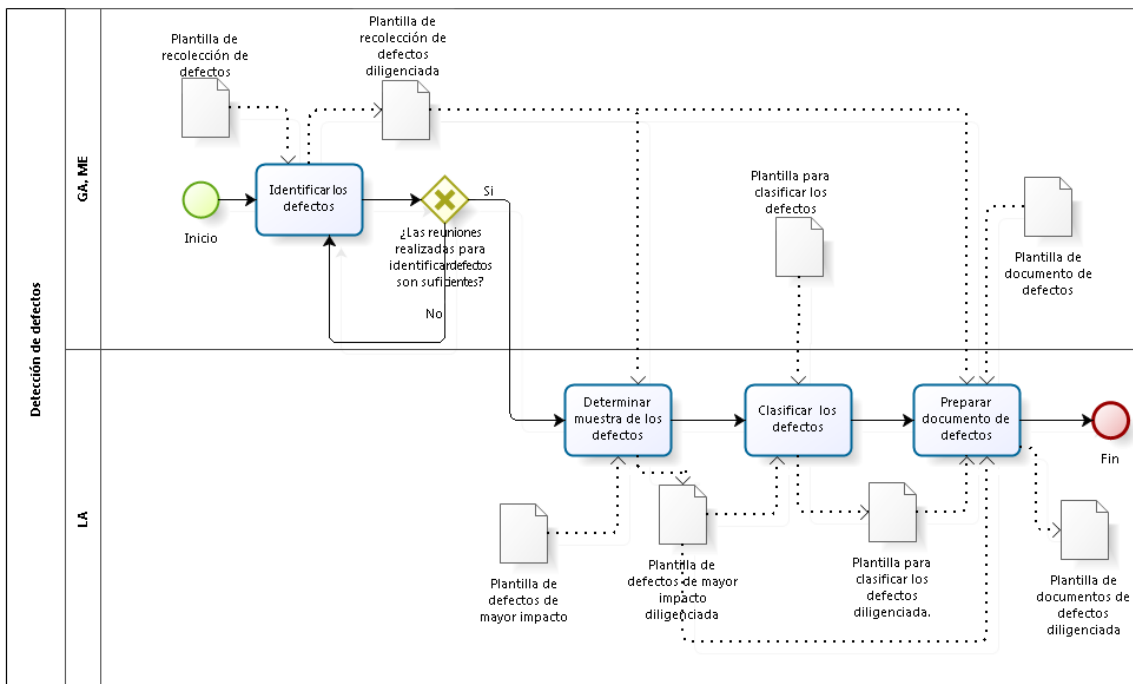


Figura 4. Diagrama de actividad Detección de defectos

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<b>Actividad</b>	Detección de causas fundamentales	
<b>Propósito</b>	El propósito de la actividad <i>Detección de causas fundamentales</i> es identificar, analizar y clasificar las causas de los problemas detectados. Así como dar lineamientos para eliminar las causas de mayor impacto.	
<b>Descripción</b>	<p>La actividad <i>Detección de causas fundamentales</i> se compone de las tareas mencionadas a continuación:</p> <ol style="list-style-type: none"> <li>1. Identificar las causas: el grupo de análisis causal se encarga de determinar las causas que están ocasionando los defectos seleccionados. Para realizar esta tarea se debe realizar un reunión con todos los miembros del grupo de análisis causal, donde cada causa debe estar clasificada de acuerdo a las categorías planteados por Kaoru Ishikawa.</li> <li>2. Analizar las causas: el grupo de análisis causal debe determinar las causas que están provocando la mayor cantidad de defectos y aquellas que causan mayor impacto en el desarrollo del proyecto. Esto se debe determinar mediante una reunión grupal.</li> <li>3. Desarrollar recomendaciones: el Líder de análisis causal se encarga de redactar un informe, el cual contiene propuestas de acción para eliminar las causas de mayor impacto en el desarrollo del proyecto.</li> <li>4. Iniciar solicitud de cambio: el encargado de realizar un cambio debe notificar los cambios a realizar.</li> <li>5. Divulgar al personal correspondiente: El líder de análisis causal se encarga de enviar un correo, al personal encargado de implementar los cambios sugeridos en las recomendaciones.</li> <li>6. Hacer seguimiento a las recomendaciones: el Líder de análisis causal se debe encargar de verificar que las recomendaciones realizadas se implementen.</li> <li>7. Preparar documento de causas: se crea el documento que contiene la información obtenida en las tareas realizadas anteriormente.</li> </ol>	
<b>Objetivos</b>	<p>O1. Identificar las causas de los defectos, mediante el uso de técnicas propuestas en este trabajo de grado.</p> <p>O2. Desarrollar las recomendaciones que se deben implementar para eliminar las causas que tienen mayor impacto en el desarrollo del proyecto.</p> <p>O3. Notificar al personal involucrado, lo cambios sugeridos para eliminar las causas.</p>	
<b>Responsables</b>	Grupo de análisis causal, líder de análisis causal	
<b>Entradas</b>		<b>Fuente</b>
Plantilla de documento de defectos diligenciada		Detección de defectos

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Salidas		Destino
Plantilla de documento de causas diligenciada		Documentar
Productos Internos		
Plantilla de causas encontradas diligenciada		
Plantilla de reporte de causas fundamentales diligenciada		
Plantilla de recomendaciones diligenciada		
Plantilla de solicitud de cambios diligenciada		
Plantilla de seguimiento de recomendaciones diligenciada		
Tareas		
T1. 1 Identificar las causas		
Entradas	Plantilla para clasificar los defectos diligenciada. Plantilla de causas encontradas	
Rol	Descripción	
GA, LA	<p>T1.1 Reunión grupal</p> <p>Mediante una reunión los integrantes del grupo de análisis causal realizan una sesión de lluvia de ideas para determinar las causas de los defectos. El líder de análisis causal debe determinar cuál de las siguientes técnicas emplear en la detección de las causas.</p> <ul style="list-style-type: none"> <li>❖ Diagrama de Ishikawa: esta técnica puede ser empleada cuando se desea visualizar gráficamente las causas y motivar al personal para la construcción de este diagrama.</li> <li>❖ Mapas causales: los mapas causales son una buena elección en aquellos casos que existen relaciones complejas, asimismo es una buena forma para representar defectos que han surgido por múltiples causas en donde se arma una red compleja. Esta técnica evita duplicar las causas, ya que cada causa se coloca una sola vez en el diagrama y de ésta se pueden derivar múltiples causas.</li> </ul> <p>Los diagramas anteriormente mencionados deben ser dibujados por el LA, con el apoyo del todos los miembros del grupo. Esta Reunión debe tener una duración mínima de una hora.</p> <p>Para realizar el diagrama (Ishikawa o mapa causal) se recomienda tener como base las categorías propuestas por Kaoru Ishikawa: herramientas, entradas, personas y métodos.</p> <p>A continuación se dan ejemplos de posibles causas que generan defectos en el desarrollo de un proyecto software:</p> <ul style="list-style-type: none"> <li>❖ Falta de comunicación</li> <li>❖ Deficiencia del proceso</li> <li>❖ Formación y habilidades inadecuadas</li> <li>❖ Estimaciones inadecuadas para el desarrollo de las tareas</li> <li>❖ Poco entendimiento de los requisitos</li> <li>❖ Uso inconsistente del entorno de desarrollo</li> </ul>	

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<ul style="list-style-type: none"> <li>❖ Uso inconsistente del paquete de interfaz de usuario</li> <li>❖ Uso de herramientas inadecuadas</li> </ul> <p>Finalmente el Líder de análisis causal se debe encargar de realizar un reporte de las causas encontradas (Anexo 1, Plantilla de causas encontradas).</p>
LA	<p>T1.2 Observación no participante</p> <p>El LA debe observar al equipo que desarrolla el proyecto, con el fin de identificar comportamientos o conductas que puedan ser la causa de los defectos evidenciados. Esta tarea debe realizarse una semana antes de la reunión propuesta para identificar causas (T1.1)</p>
<b>Salidas</b>	<p>Diagramas realizados</p> <p>Plantilla de causas encontradas diligenciada</p>
<b>T2. Analizar las causas</b>	
<b>Entradas</b>	<p>Plantilla de causas encontradas diligenciada</p> <p>Diagramas realizados</p> <p>Plantilla de causas fundamentales</p>
<b>Rol</b>	<b>Descripción</b>
GA, LA	<p>T2.1 Reunión grupal</p> <p>En la reunión realizada para identificar las causas, el GA debe determinar las causas que deben ser eliminadas. Para ello cada integrante del equipo expone en voz alta las causas que considera tienen mayor impacto y necesitan ser tratadas con acciones correctivas. Posterior a esto se debe llegar a un consenso y minimizar el conjunto de causas que se tenía inicialmente.</p>
LA	<p>T2.2 Realizar Gráfico radial</p> <p>El análisis de gráfico radial ayuda al LA a identificar la relación entre los defectos y las causas introducidas en las diferentes disciplinas del desarrollo del software. Mediante esta técnica es posible establecer los defectos introducidos en fases críticas para el proyecto (Anexo 2).</p>
<b>Salidas</b>	<p>Plantilla de causas fundamentales diligenciada</p>
<b>T3. Desarrollar recomendaciones</b>	
<b>Entradas</b>	<p>Plantilla de causas fundamentales diligenciada</p> <p>Plantilla de recomendaciones</p>
<b>Rol</b>	<b>Descripción</b>
LA	<p>T3.1 Realizar plan de recomendaciones</p> <p>El LA debe diligenciar la plantilla de recomendaciones. Las recomendaciones especifican cambios necesarios para eliminar o reducir las causas que están generando los defectos. Es posible que se necesiten cambios en:</p> <ul style="list-style-type: none"> <li>• La definición del proceso</li> <li>• Capacitación</li> <li>• Herramientas</li> <li>• Métodos</li> <li>• Productos de trabajo</li> <li>• Roles asignados</li> <li>• Metodología implementada</li> </ul> <p>Por otro lado, el LA debe asignar las personas encargadas de implementar los cambios y</p>

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	mantener informado al personal correspondiente sobre estos cambios.
<b>Salidas</b>	Plantilla de recomendaciones diligenciada
<b>T4. Iniciar solicitud de cambio</b>	
<b>Entradas</b>	Plantilla de solicitud de cambio Plantilla de recomendaciones diligenciada
<b>Rol</b>	<b>Descripción</b>
SC	T4.1 Solicitar cambio Para solicitar un cambio se debe llenar la plantilla de solicitud de cambio, justificando el cambio y especificando los artefactos que se verán afectados. Para su respectiva aprobación el gerente del proyecto debe registrar la solicitud de cambio y evaluar el impacto de éste sobre el proyecto.
<b>Salidas</b>	Plantilla de solicitud de cambio diligenciada
<b>T5. Divulgar al personal correspondiente</b>	
<b>Entradas</b>	Plantilla de solicitud de cambio diligenciada
<b>Rol</b>	<b>Descripción</b>
LA	T5.1 Envío de correos electrónicos El LA debe dar a conocer a las personas involucradas en los defectos encontrados, el documento de recomendaciones. Para ello se debe notificar a cada persona los cambios a implementar, esto mediante un correo electrónico.
<b>Salidas</b>	Correos enviados
<b>T6. Hacer seguimiento a las recomendaciones</b>	
<b>Entradas</b>	Plantilla de recomendaciones diligenciada Plantilla de solicitud de cambio diligenciada Plantilla de seguimiento de recomendaciones
<b>Rol</b>	<b>Descripción</b>
LA	T6.1 Hacer seguimiento a las recomendaciones El LA se encarga de verificar que los cambios se implementen antes de la fecha límite y no se introduzcan nuevos defectos. También se debe verificar que los cambios implementados ayudaron a reducir o eliminar las causas de los defectos detectados.
<b>Salidas</b>	Plantilla de seguimiento de recomendaciones diligenciada
<b>T7. Preparar documento de causas</b>	
<b>Entradas</b>	Plantilla del documento de causas Plantilla de causas encontradas diligenciada Plantilla de reporte de causas fundamentales diligenciada Plantilla de recomendaciones diligenciada Plantilla de solicitud de cambios diligenciada Plantilla de seguimiento de recomendaciones diligenciada
<b>Rol</b>	<b>Descripción</b>
LA	T7.1 Crear documento Elaborar un documento que contenga la información recolectada en las anteriores tareas
<b>Salidas</b>	Plantilla del documento de causas diligenciado

Tabla 16. Diagrama de actividad Detección de causas fundamentales

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

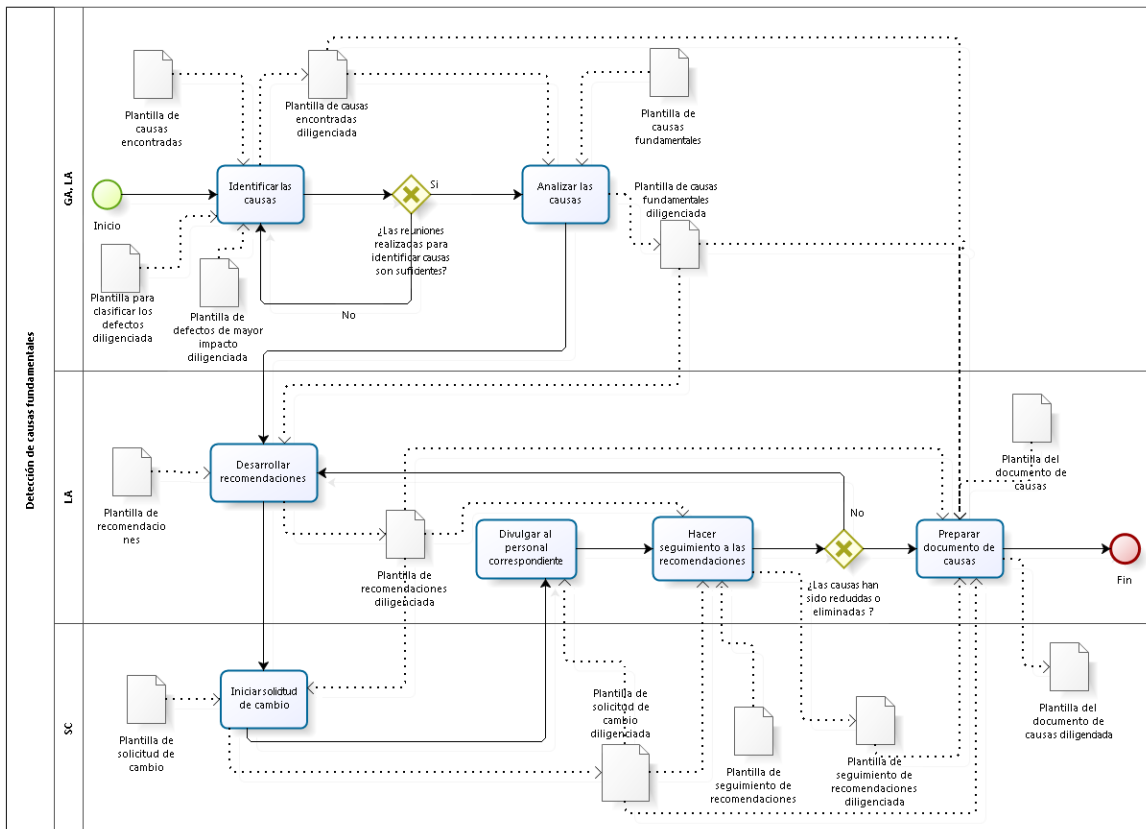


Figura 5. Detección de causas fundamentales

<b>Actividad</b>	Documentar
<b>Propósito</b>	El propósito de la actividad <i>Documentar</i> es registrar las lecciones aprendidas en el desarrollo del proyecto y almacenar en el repositorio de la organización los documentos obtenidos al aplicar el procedimiento de análisis causal.
<b>Descripción</b>	<p>La actividad <i>Documentar</i> se compone de las tareas mencionadas a continuación:</p> <ol style="list-style-type: none"> <li>1. Documentar lecciones aprendidas: el grupo encargada de desarrollar el proyecto se reúne para tratar temas relacionados con los defectos encontrados y sus causas. Además se debe dar a conocer a todo el grupo las experiencias (buenas y malas) vividas durante el desarrollo del proyecto. El líder de análisis causal se encarga de registrar los aspectos fundamentales tratados en la reunión.</li> <li>2. Almacenar información obtenida en el procedimiento: El LA debe guardar todos los documentos obtenidos en la aplicación del procedimiento en el repositorio de la organización. Esto con el fin de mantener información que permite mejoras en proyectos futuros.</li> </ol>
<b>Objetivos</b>	O1. Obtener conocimiento sobre las diferentes situaciones encontradas durante el

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	desarrollo del proyecto.  O2. Mantener información histórica a partir de los documentos obtenidos al aplicar el procedimiento.
<b>Responsables</b>	Líder de análisis causal
<b>Entradas</b>	
Perfil del encargado de aplicar el procedimiento	Preparación
Perfiles de los miembros del grupo de análisis causal	Preparación
Plantilla para clasificar los defectos diligenciada	Detección de defectos
Plantilla de seguimiento de recomendaciones diligenciada	Detección de causas fundamentales
<b>Salidas</b>	
plantilla de lecciones aprendidas diligenciada	
<b>Productos Internos</b>	
<b>Tareas</b>	
<b>T1. Documentar lecciones aprendidas</b>	
<b>Entradas</b>	Plantilla de lecciones aprendidas
<b>Rol</b>	Descripción
LA, ME	T1.1 Reuniones grupales El líder de análisis causal debe convocar una reunión donde se invite a todos los miembros involucrados en el proyecto. En ésta se debe tratar los defectos que han surgieron durante el proyecto y las causas que han sido encontradas. Los miembros de grupo deben cuestionarse sobre cada una de las causas y lo mecanismos empleados para eliminarlas. De esta reunión deben salir las lecciones aprendidas a nivel grupal e individual. El documento de lecciones aprendidas contiene todos los defectos trabados durante las diferentes iteraciones, las acciones implementadas y los resultados obtenidos.  Esta reunión debe realizarse al finalizar el proyecto
<b>Salidas</b>	plantilla de lecciones aprendidas diligenciada
<b>T2. Almacenar información obtenida en el procedimiento</b>	
<b>Entradas</b>	Documentos obtenidos al aplicar el procedimiento
<b>Rol</b>	<b>Descripción</b>
LA	T2.1 Almacenar información obtenida en el procedimiento El LA debe guardar todos los documentos obtenidos en la aplicación del procedimiento, para ello se hace uso del repositorio con el que cuente la organización. Esto con el fin de mantener información que permite mejoras en proyectos futuros.
<b>Salidas</b>	No aplica

Tabla 17. Diagrama de actividad Documentar



## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

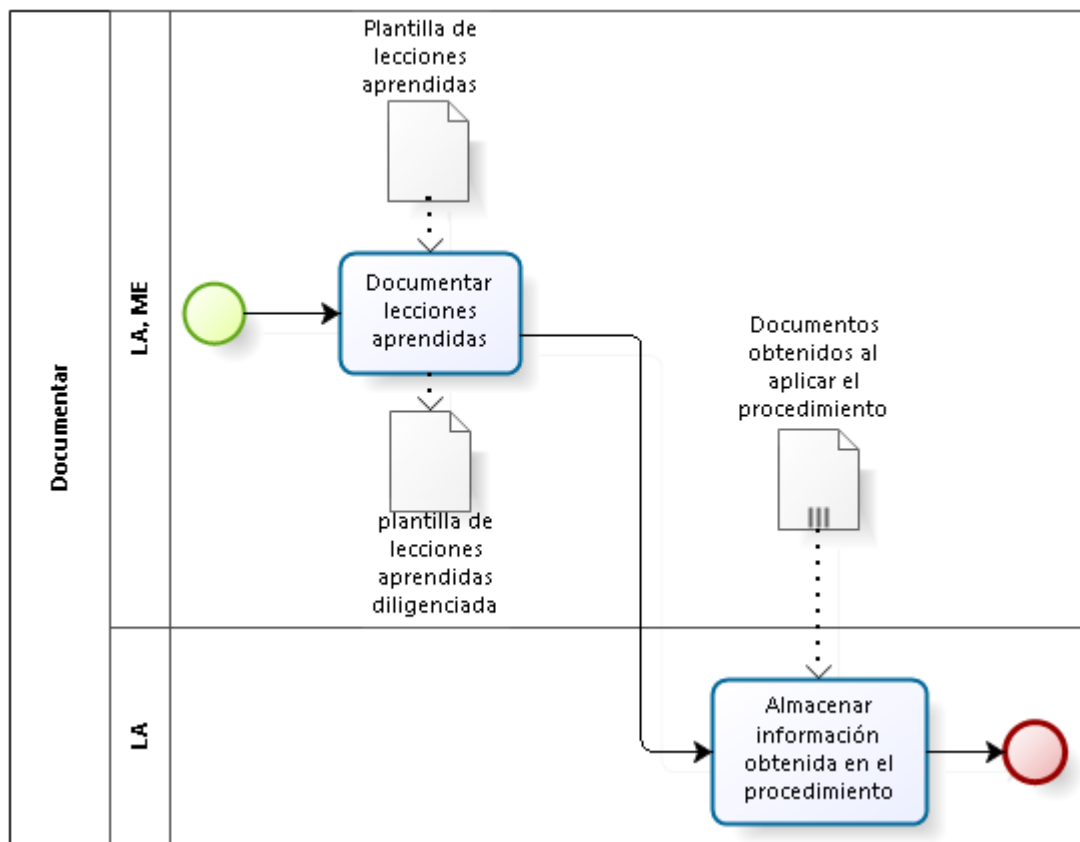


Figura 6. Diagrama de actividad Documentar

## Capítulo 4. Caso de estudio

---

Este capítulo presenta el caso de estudio, el cual fue empleado como método de evaluación del procedimiento propuesto, éste fue aplicado a un grupo encargado de desarrollar una aplicación software para la asignatura proyecto II, perteneciente al Programa de Ingeniería de Sistemas de la Universidad del Cauca. Los componentes de ese caso de estudio que tomó como referente el protocolo de planeación propuesto por Brereton et al. [52], se muestran a continuación: diseño, sujetos de investigación, unidad de análisis, procedimiento de campo, intervención y análisis.

### 4.1. Antecedentes

Los estudiantes que cursan la asignatura proyecto II desarrollan un trabajo en grupos, en los cuales a cada miembro se le asigna un rol específico. Algunos se encargan de los roles de gestión de proyectos, del aseguramiento de la calidad o del desarrollo del software. Los participantes deben tener en cuenta conocimientos relacionados con gestión de proyectos informáticos, procesos de desarrollo y calidad del software.

En la Tabla 18 se encuentra la pregunta de investigación principal (PP) y las preguntas adicionales (PA) que guían el caso de estudio académico.

Preguntas de investigación	
<b>PP</b>	¿El procedimiento planteado para realizar análisis causal, es adecuado para las pequeñas organizaciones de desarrollo software?
<b>PA1</b>	¿Las técnicas sugeridas son adecuadas para realizar análisis causal en las pequeñas organizaciones de desarrollo software?
<b>PA2</b>	¿El procedimiento planteado para realizar análisis causal, permite evidenciar las causas de los defectos presentados durante la ejecución de proyectos software?

Tabla 18. Preguntas de investigación del caso de estudio

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### **4.2. Diseño del caso de estudio**

El diseño de caso de estudio se realizó tomando como referencia el trabajo propuesto por Yin et al. [53]. El diseño que se empleó en este estudio es de tipo simple holístico, ya que el procedimiento planteado para realizar análisis causal en las pequeñas organizaciones de desarrollo software, fue aplicado en un solo grupo que tiene características similares a las VSEs. Para este propósito se empleó como unidad de análisis un proyecto de desarrollo denominado “Sistemas de gestión de actividades diarias”. El objeto de estudio fue el procedimiento planteado y las medidas usadas para dar respuesta a las preguntas de investigación propuestas fueron: 1) el esfuerzo realizado para llevar a cabo las actividades propuestas en el procedimiento, 2) el número de causas encontradas, 3) percepción de las personas involucradas en el proyecto.

Las características mostradas a continuación son propias de las VSEs y también se pudieron evidenciar en el grupo Proyecto II.

- ❖ Poco número de empleados
- ❖ Pocos recursos económicos
- ❖ Poca planificación en sus proyectos

Una de las características en la que difiere el grupo de Proyecto con las VSEs es el nivel de experiencia, ya que el grupo proyecto solo contaba con una personal que estaba o había laborando en un entorno empresarial. Es de anotar que en las VSEs por lo general hay personas que ya cuentan con experiencia laboral certificada.

### **4.3. Sujetos de investigación y unidad de análisis**

El caso de estudio se realizó en un entorno académico y los sujetos para esta investigación fueron un grupo de estudiantes de la asignatura Proyecto II de la Universidad del Cauca. El grupo estaba conformado por siete estudiantes de noveno semestre de Ingeniería de Sistemas, que desempeñaron roles específicos dentro del proceso definido para desarrollar el proyecto. Estas funciones fueron asignadas de acuerdo a las capacidades y habilidades de los miembros del equipo. Los responsables de cada una de las prácticas definidas acorde al modelo de procesos para la industria del software MoProSoft fueron:

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### Sub-grupo de Gestión

- ❖ Wilian Fernando Pantoja, (líder).
- ❖ Cesar Eduardo Restrepo.

### Sub-grupo de Aseguramiento de Calidad:

- ❖ María Del Mar Granda, (líder).
- ❖ Cesar Eduardo Restrepo.

### Sub-grupo de Desarrollo:

- ❖ Karen Julieth Bolaños, (líder de desarrollo).
- ❖ Hernán Guillermo Dulcey.
- ❖ Cristian Daniel Yanza.
- ❖ Andrea Cristina Zambrano.

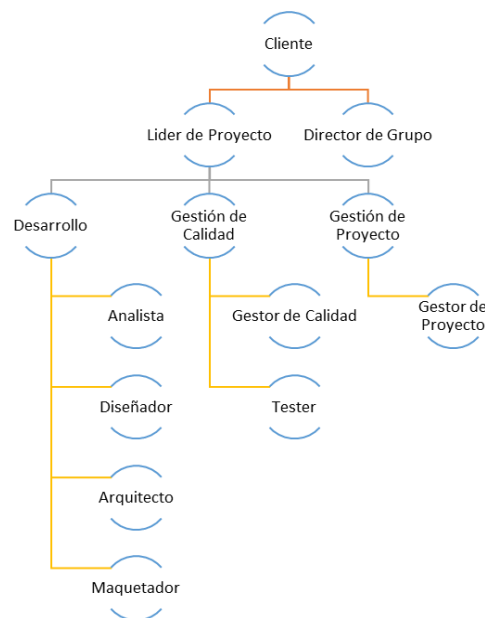


Figura 7. Estructura del proyecto “Sistemas de gestión de actividades diarias”  
Fuente: Project Charter del proyecto “Sistemas de gestión de actividades diarias”

## 4.4. Procedimiento de campo

El procedimiento de campo a seguir es fundamental para ejecutar e implementar la propuesta de este trabajo de grado, lo cual permite conocer el grupo y obtener los resultados esperados. Esto nos lleva a adaptar las actividades y tareas según las condiciones y características del proyecto. El procedimiento de campo consta de cuatro

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

actividades (Figura 8): Asignación del proyecto piloto, Detección de defectos, Detección de causas fundamentales y Documentar. Los productos de trabajo involucrados en las actividades mencionadas anteriormente son todos los especificados en el procedimiento propuesto.

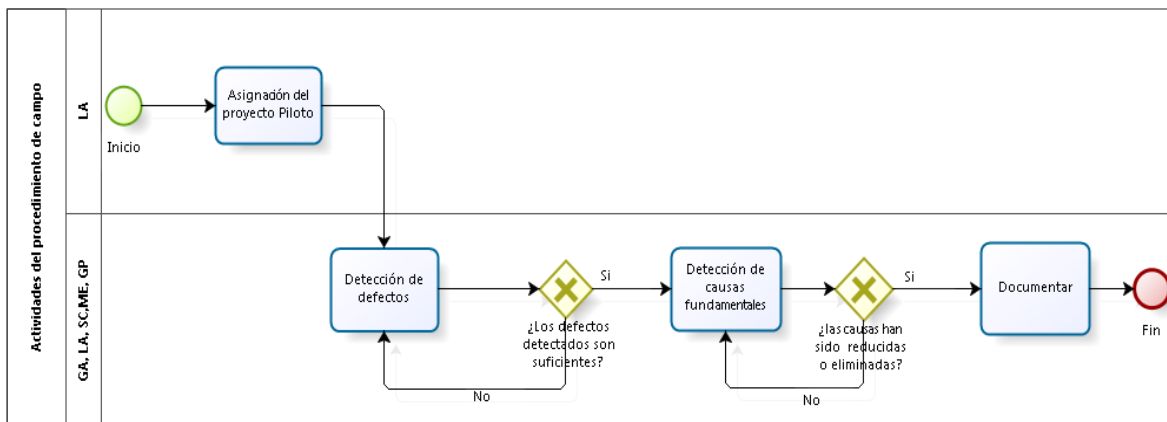


Figura 8. Procedimiento de campo que rige las actividades del caso de estudio

### 4.5. Intervención

Para conocer el grupo objeto de estudio, se observaron sus actividades y comportamiento en el desarrollo del proyecto. Para ello fue necesario asistir a las reuniones realizadas durante la semana (4 días por semana). Inicialmente se analizó el grupo de trabajo, de acuerdo con los roles establecidos para la ejecución del proyecto. Asimismo, se preparó al grupo para acoger el procedimiento de análisis causal y posteriormente se procedió a aplicarlo y asignar un proyecto piloto, el cual se ejecutó dentro del grupo académico que cursa la asignatura Proyecto II.

#### 4.5.1. Diagnóstico inicial

Inicialmente se procedió a realizar una valoración para conocer el proceso que sigue el grupo, lo que permitió comprender sus actividades, interpretarlas y analizarlas, para abstraer información relevante para el éxito del procedimiento planteado. En un principio el grupo no había considerado el análisis causal entre sus actividades, por esta razón fue necesario motivarlos y explicarles las ventajas de su aplicación en la resolución de problemas y la mejora de la calidad. Para ello se llevó a cabo una charla

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

inicial, en la que se trataron estos temas. Adicionalmente fue necesario asistir a las reuniones establecidas por el equipo y observar las actividades que estaban desarrollando.

### 4.5.1.1. Adaptaciones al procedimiento

Para realizar el caso de estudio y dadas las tareas planteadas en la actividad *Preparación* (elección del líder de análisis causal encargado de aplicar el procedimiento, elección del grupo de análisis causal, capacitación sobre la estructura y componentes del procedimiento) se decidió suprimir esta actividad, debido a que la autora de este trabajo de grado asumió el rol de Líder (LA) e involucró a todos los miembros del proyecto como parte del equipo de análisis causal (GA), puesto que se consideró importante los aportes que puedan dar desde los diferentes roles definidos. De otro lado, no fue necesario ejecutar la tarea *Capacitación sobre la estructura y componentes del procedimiento*, teniendo en cuenta que el líder ya conocía el funcionamiento de éste.

Se debe agregar que el envío de correo electrónicos como soporte a la tarea de identificar los defectos (actividad detectar defectos) no se realizó, ya que todos los miembros hacían parte del grupo de análisis causal y éstos trataban los defectos durante las reuniones grupales.

De otra parte, es importante aclarar que la tarea *Iniciar solicitud de cambio* no estaba definida inicialmente en el procedimiento, ya que esta surgió al finalizar el caso de estudio como resultado de las observaciones realizadas. Asimismo la técnica FMEA no se había considerado inicialmente, pero fue agregada al procedimiento para aquellas situaciones en que se quiera tener mayor rigurosidad en el momento de priorizar los defectos (tarea Determinar muestra de los defectos).

### 4.5.2. Asignación del proyecto piloto

El procedimiento planteado en este trabajo de grado fue evaluado mediante un proyecto piloto denominado “Sistemas de gestión de actividades diarias”, desarrollado por los estudiantes que cursaban la asignatura Proyecto II. El cual se realizó en el lenguaje de programación PHP, con base de datos en MySQL.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

De acuerdo a la documentación y lo firmado con el cliente, el objetivo era realizar un aplicativo web que permitiera gestionar, planificar y registrar actividades, así como generar información relacionada al esfuerzo y productividad empleada para cada día. Para este propósito el sistema debía permitir a los usuarios organizar actividades y tareas, las cuales pueden tener definida previamente la fecha y hora límite de su ejecución, pero se puede dar el caso en que estos atributos no se encuentren definidos. De este modo, el sistema registra el momento en el que se inicia una tarea, teniendo en cuenta las pausas que se hacen durante su ejecución y el momento de finalización.

Los requerimientos del producto se resumen en los siguientes aspectos:

- ❖ El sistema permitirá gestionar (crear, editar, eliminar, visualizar) y organizar categorías (una categoría es un conjunto de actividades )del quehacer diario del usuario.
- ❖ El sistema permitirá gestionar (crear, editar, eliminar, visualizar) y organizar actividades (una actividad es un conjunto de tareas) del quehacer diario del usuario.
- ❖ El sistema permitirá gestionar (crear, editar, eliminar, visualizar) y organizar tareas (Una tarea es la unidad atómica que refleja las acciones que realiza el usuario en su quehacer diario) del quehacer diario del usuario.
- ❖ El sistema permitirá al usuario planificar las tareas y las fechas en que se van a ejecutar.
- ❖ El sistema permitirá la visualización de un calendario mensual, en donde se distingan los días donde el usuario tiene tareas planificadas pendientes.
- ❖ El sistema permitirá listar las tareas diarias de un usuario y las pendientes (tareas no terminadas o no comenzadas en la fecha planeada)
- ❖ El sistema permitirá adicionar a un día específico tareas que no se hayan terminado o comenzado en la fecha planificada, siempre y cuando sean de tipo movibles, para su previa ejecución, pausa y terminación.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- ❖ El sistema permitirá al usuario llevar el tiempo real que dedica a cada tarea planeada. Cada vez que el usuario comienza una tarea se creará un nuevo registro de tiempo, el último registro será cuando el usuario finalice la tarea. Cada registro tiene una fecha de inicio, hora de inicio y duración, campos que pueden ser modificados manualmente por el usuario.
- ❖ El sistema permitirá al usuario seleccionar el nivel de sensación de productividad (alta, media, baja), para cada día.
- ❖ El sistema permitirá la visualización de una barra de progreso por actividad donde se muestra el avance de tareas terminadas (número de tareas terminadas vs el número de tareas totales hasta ese momento).
- ❖ El sistema permitirá al usuario visualizar mediante una barra de progreso, el avance que lleva de las tareas programadas para cada día (número de tareas terminadas vs el número de tareas totales hasta ese momento).
- ❖ El sistema generará los siguientes reportes:
  1. El sistema muestra mediante un gráfico circular el porcentaje de tareas terminadas y pendientes, esto para un rango de fechas seleccionado por el usuario. También se muestra una tabla donde se muestra las tareas completadas y las pendientes.
  2. El sistema debe mostrar la sensación de productividad mediante un gráfico de barras que indica por cada día si la productividad fue alta, media o baja, esto para un rango de fechas seleccionado por el usuario. También se muestra una tabla con la sensación de productividad del usuario y el tiempo invertido para cada día.
  3. El sistema debe mostrar el tiempo invertido por cada tipo de tarea (investigación, docencia, gestión, extensión), mediante un gráfico de barras. , esto para un rango de fechas seleccionado por el usuario. También se muestra una tabla donde por cada tipo de tarea se muestra el tiempo y las tareas pertenecientes a ésta.



## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### 4.5.3. Ejecución del procedimiento

El procedimiento fue ejecutado al final de cada iteración donde se hacía entrega de módulos acordados con el cliente. En esa fecha se realizan reuniones grupales con el fin de tratar los defectos evidenciados, detectar su causa y aplicar las acciones necesarias para reducirlas o eliminarlas, así como documentar lo encontrado. Inicialmente se ejecutó la primera versión del procedimiento propuesto y según las observaciones realizadas en cada iteración se plantearon mejoras con el propósito de obtener un procedimiento óptimo para las VSEs.

Uno de los aspectos observados durante la ejecución del procedimiento fue que a pesar de que éste inicialmente estaba enfocado en defectos directamente relacionados con el producto software (el defecto es un cambio que se debe realizar para obtener el correcto funcionamiento del software) se evidenció la necesidad de tratar los defectos relacionados con la ejecución del proyecto, tales como retrasos en entregas, falta de trazabilidad en los artefactos generados durante el desarrollo del proyecto, ausencia de especificaciones técnicas, requerimientos no especificados, entre otros. Al respecto conviene decir que esto se tuvo en cuenta, debido a que se consideró que pueden ser la fuente para ocasionar los defectos en el producto software.

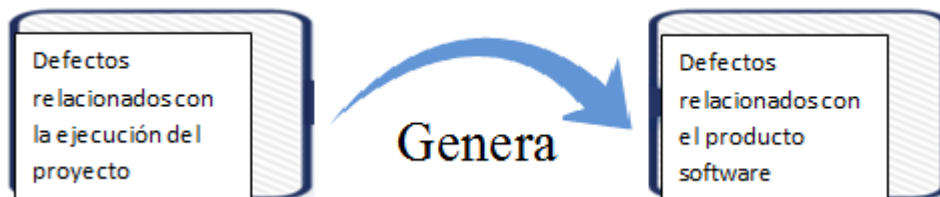


Figura 9. Relación de tipos de defectos

La ejecución del procedimiento se llevó a cabo por fases, como se muestra en la Tabla 19.

<b>Fase</b>	<b>Ejecución de las actividades</b>
Fase 1	15 de septiembre del 2014
Fase 2	06 de octubre del 2014
Fase 3	27 de octubre del 2014
Fase 4	24 de noviembre del 2014

Tabla 19. Fases del procedimiento

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### 4.6. Recolección y análisis de datos

En esta sección se presenta los datos recolectados durante el caso de estudio, los cuales son presentados en tablas, seguidas del respectivo análisis sobre los datos encontrados.

El procedimiento fue ejecutado en cuatro fases, en las cuales se detectaron defectos y sus causas fundamentales. En el anexo 7 se muestra en detalle las dos primeras fases, junto con su respectiva documentación.

La recolección de datos se hizo mediante las plantillas propuestas para apoyar el procedimiento y el registro de la información requerida. Para todas las actividades se registró el tiempo invertido en cada una de ellas, asimismo se anotó las causas halladas en cada fase y las recomendaciones establecidas, esto para determinar el esfuerzo requerido en las actividades propuestas y el número de causas encontradas. Para conocer la percepción de las personas involucradas en el proyecto se diseñó una entrevista que permitió conocer su opinión sobre éste (Anexo 5).

#### 4.6.1. Esfuerzo

Para cada fase se realizaron reuniones en las que participaron los miembros del grupo de análisis causal (GA), dirigidos por el líder. En la Tabla 20, Tabla 21 y Tabla 22 se presenta el esfuerzo en horas-persona de las actividades planteadas en el procedimiento. En la segunda columna se muestra el tiempo asociado en minutos que cada miembro del equipo empleó durante las reuniones de trabajo. En la quinta columna se presenta el esfuerzo (Horas-persona) del líder, el cual equivale al esfuerzo invertido durante las reuniones más el esfuerzo que implica el registro de la información y las tareas adicionales asociadas a este rol. Finalmente en la sexta columna se obtiene el esfuerzo total, resultante de sumar el esfuerzo del grupo de análisis causal y el esfuerzo del líder.

A continuación se muestra las fórmulas empleadas para obtener los resultados mostrados en la Tabla 20, Tabla 21 y Tabla 22.

$$Total.GA = Tiempo * cantidad de personas$$

Ecuación 1. Esfuerzo del GA

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

$$Total = Esf.GA + Esf.Líder$$

Ecuación 2. Esfuerzo por fases de las actividades propuestas en el procedimiento

<b>Actividad 1: Detección de defectos</b>					
<b>Tarea 1: Identificar los defectos</b>					
<b>Fase</b>	<b>Tiempo (minutos)</b>	<b>Cantidad personas</b>	<b>Tot. GA (Horas-persona)</b>	<b>Esf. Líder (Horas-persona)</b>	<b>Total (Horas-persona)</b>
Fase 1	20 (0,33 horas)	6	1,98	0,5	<b>2,48</b>
Fase 2	25 (0,42 horas)	6	2,52	0,59	<b>3,11</b>
Fase 3	25 (0,42 horas)	4	1,68	0,5	<b>2,18</b>
Fase 4	40 (0,67)	6	4,02	0,84	<b>4,86</b>
<b>Tarea 2: Determinar muestra de los defectos</b>					
Fase 1	0	6	0	0,17	<b>0,17</b>
Fase 2	0	6	0	0,25	<b>0,25</b>
Fase 3	0	4	0	0,17	<b>0,17</b>
Fase 4	0	6	0	0,42	<b>0,42</b>
<b>Tarea 3: Clasificar los defectos</b>					
Fase 1	0	6	0	0	<b>0</b>
Fase 2	0	6	0	0,58	<b>0,58</b>
Fase 3	0	4	0	0,5	<b>0,5</b>
Fase 4	0	6	0	0,67	<b>0,67</b>

Tabla 20. Esfuerzo involucrado en cada actividad Detección de defectos

Como se puede ver en la Tabla 20 la cantidad de personas que participaron en las reuniones durante la tercera fase fueron 4, debido a que para esta fecha dos miembros del grupo de análisis causal no pudieron asistir.

Por otra parte, la tarea *Determinar muestra de los defectos* y *Clasificar los defectos* no tiene registrado el esfuerzo para el grupo de análisis causal, dado que esta tarea está a cargo del líder. Asimismo se puede ver que en la fase 1 de la tarea 3, tanto el esfuerzo grupal como el esfuerzo del líder es cero (0), esta situación se presentó porque durante esa fase los defectos tratados estaban relacionados con la ejecución del proyecto, por lo cual se continuó con la tarea *Identificar las causas* (siguiendo lo establecido en el procedimiento).

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<b>Actividad2 :Detección de causas fundamentales</b>					
<b>Tarea 1: Identificar las causas</b>					
<b>Fase</b>	<b>Tiempo (minutos)</b>	<b>Cantidad personas</b>	<b>Tot. GA (Horas-persona)</b>	<b>Esf. Líder (Horas-persona)</b>	<b>Total (Horas-persona)</b>
Fase 1	60 (1 hora)	6	6	1,17	<b>7,17</b>
Fase 2	35 (0,58 horas )	6	3,48	0,75	<b>4,23</b>
Fase 3	25 (0,42 horas)	6	1,68	0,5	<b>3,2</b>
Fase 4	40 (0,67 horas)	6	4,02	0,84	<b>4,86</b>
<b>Tarea 2: Analizar las causas</b>					
Fase 1	45 (0,75 horas)	6	4,5	1	<b>5,5</b>
Fase 2	30 (0,5 horas)	6	3	0,67	<b>3,67</b>
Fase 3	25 (0,42 horas)	6	1,68	0,5	<b>3,02</b>
Fase 4	40 (0,67 horas)	6	4,02	0,75	<b>4,77</b>
<b>Tarea 3: Desarrollar recomendaciones</b>					
Fase 1	0	6	0	0,5	<b>0,5</b>
Fase 2	0	6	0	0,58	<b>0,58</b>
Fase 3	0	6	0	0,5	<b>0,5</b>
Fase 4	0	6	0	0,58	<b>0,58</b>
<b>Tarea 4: Hacer seguimiento a las recomendaciones</b>					
Fase 1	0	6	0	0,25	<b>0,25</b>
Fase 2	0	6	0	0,28	<b>0,28</b>
Fase 3	0	6	0	0,22	<b>0,22</b>
Fase 4	0	6	0	0,33	<b>0,33</b>

Tabla 21. Esfuerzo involucrado en cada actividad Detección de causas fundamentales

En la Tabla 21 se puede ver que las tareas 3 y 4 no tienen registrado esfuerzo grupal, debido a que estas tareas están a cargo del líder.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<b>Actividad 3: Documentar</b>					
<b>Documentar lecciones aprendidas</b>					
<b>Fase</b>	<b>Tiempo(minutos)</b>	<b>Cantidad</b>	<b>Tot. GA (Horas- personas)</b>	<b>Esf. Líder (Horas- persona)</b>	<b>Total (Horas- persona)</b>
Fase 1	0	6	0	0	<b>0</b>
Fase 2	0	6	0	0	<b>0</b>
Fase 3	0	6	0	0	<b>0</b>
Fase 4	0	6	0	1,5	<b>1,5</b>
<b>Almacenar información</b>					
Fase 1	0	6	0	0	<b>0</b>
Fase 2	0	6	0	0	<b>0</b>
Fase 3	0	6	0	0	<b>0</b>
Fase 4	0	6	0	0	<b>0</b>

Tabla 22. Esfuerzo involucrado en cada actividad Documentar

De la Tabla 22 se puede ver que la tarea *Documentar lecciones aprendidas* en la fase 1,2 y 3 no tienen registrado el esfuerzo, sin embargo esto no significa que no se haya realizado. Esta situación se presentó debido a que el proyecto tuvo un retraso del 75% en la primera fase, en consecuencia no se tenía el tiempo suficiente, por lo cual se decidió que la reunión de lecciones aprendidas no se iba a realizar durante cada fase, pero si al finalizar el proyecto. Esto no quiere decir que no se hayan tratada las lecciones aprendidas durante todo el proyecto, ya que en la fase 4 se trató de condensar todo lo aprendido durante los cuatro meses de trabajo.

En la Tabla 23 se muestra el esfuerzo total, involucrado en cada fase.

<b>Horas-persona</b>				
<b>Actividad</b>	<b>Fase1</b>	<b>Fase 2</b>	<b>Fase 3</b>	<b>Fase 4</b>
Actividad 1	2,65	3,94	2,85	5,95
Actividad 2	13,42	8,76	6,17	10,54
Actividad 3	0	0	0	1,15
<b>Total</b>	<b>16,02</b>	<b>12,7</b>	<b>9,02</b>	<b>17,64</b>
<b>Esfuerzo total del procedimiento: 55,38</b>				

Tabla 23. Esfuerzo involucrado en el procedimiento

A partir de los datos consolidados en la Tabla 23 se obtiene la gráfica del esfuerzo involucrado en cada fase.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

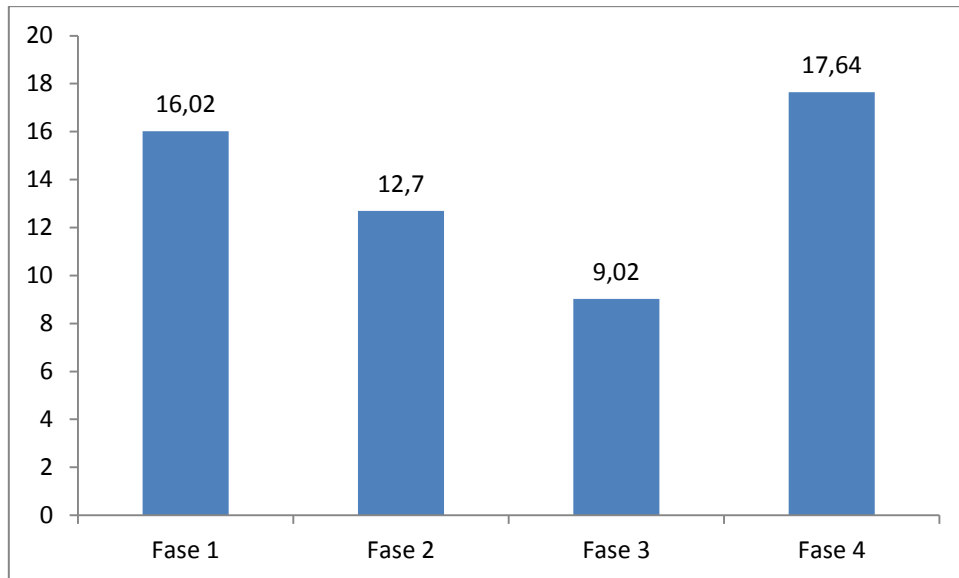


Figura 10. Esfuerzo involucrado en cada fase

La información relacionada con el esfuerzo, se obtuvo al sintetizar el tiempo empleado en las reuniones grupales y el tiempo que invirtió el líder de análisis causal en analizar y registrar la información en sus respectivas plantillas.

El esfuerzo requerido en la actividad *Preparación* no está incluido en la ejecución del proyecto, debido a que esta actividad no se realizó porque ya se tenía el líder (Autora del procedimiento) y todos los miembros del equipo formaban el grupo de análisis causal. Por estos motivos, no fue necesario hacer una selección para conformar este grupo. Cabe aclarar que para ese momento no se había definido la tarea capacitación de técnicas, dado que esta tarea surgió al finalizar el caso de estudio como mejora al procedimiento.

De la Tabla 23 se puede extraer el esfuerzo involucrado (horas-persona) para llevar a cabo cada fase en el proyecto desarrollado por el grupo de la asignatura Proyecto II. Como se puede observar para la segunda y tercera fase, el esfuerzo disminuyó considerando que el grupo y el líder de análisis causal adquirieron experiencia en las técnicas empleadas y el modo de ejecutar la reunión. Sin embargo en la cuarta fase el esfuerzo requerido aumentó, esto se debe a que el proyecto en ese momento tenía un retaso del 50%, por lo cual fue necesario colocarse al día con las funcionalidades retrasadas, lo que implicó trabajar bajo presión. En esta última fase se observó un incremento en la cantidad de defectos lo cual puede explicar el aumento del esfuerzo,

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

debido al trabajo adicional que se tuvo que hacer para recuperar el retraso llevado hasta la fecha. Además durante la cuarta fase se ejecutó la actividad Documentar lo que implicó 1,50 horas adicionales.

Ahora bien, para establecer la relación entre el esfuerzo requerido para ejecutar las actividades del procedimiento y el esfuerzo invertido en todo el proyecto, en primera instancia se procedió a calcular el esfuerzo total realizado por el líder de análisis causal durante cada una de las actividades, tal y como se muestra en la Tabla 24. Cabe mencionar que las fases 2 y 3 se hicieron en el mes de octubre, dados los retrasos presentados en la fase 1, lo que llevo a retrasos en la segunda fase. Para la fase 4, en noviembre se realizó las actividades *Detección de defectos* y *Detección de causas fundamentales* y en diciembre, mes en el cual se finalizó el proyecto y se hizo entrega de la aplicación web al cliente, se llevó a cabo la reunión de lecciones aprendidas.

Los valores mostrados en la Tabla 24 fueron obtenidos al sumar por fases el esfuerzo que invirtió el líder en las tareas propuestas en cada actividad, para ello se tomó como referencia la columna Esf. Líder (Horas-persona) de la Tabla 20 (Actividad 1), Tabla 21 (Actividad 2) y Tabla 22 (Actividad 3).

Horas-persona					
Esfuerzo total del líder					
Mes	Septiembre	Octubre		Noviembre	Diciembre
	fase 1	fase 2	fase 3	fase 4	
Actividad 1	0,67	1,42	1,17	1,93	
Actividad 2	2,92	2,28	1,72	2,5	
Actividad 3	0	0	0		1,5
<b>Total</b>	<b>3,59</b>	<b>3,7</b>	<b>2,89</b>	<b>4,43</b>	<b>1,5</b>

Tabla 24. Esfuerzo total del líder de análisis causal

En segunda lugar se procedió a calcular el esfuerzo total del proyecto, para ello se tuvo en cuenta los siguientes aspectos:

- ❖ El grupo dedicaba 9 horas semanales al proyecto.
- ❖ El caso de estudio inicio el 18 de agosto del 2014 y finalizó el 19 de diciembre del 2014.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- ❖ Los valores presentados en la columna Esf. Líder de la Tabla 25 son obtenidos a partir de los resultados obtenidos en la Tabla 24 (Esfuerzo total del líder de análisis causal). Para el mes de octubre fue necesario sumar el esfuerzo total invertido en la fase 2 y 3, ya que éstas se realizaron en el mismo mes.
- ❖ A pesar de que en la Tabla 24 el líder no tiene registrado ningún esfuerzo para el mes de agosto, en la Tabla 25 se registra un esfuerzo de 5 horas-persona, que el líder de análisis causal empleo para conocer el proyecto y la metodología de trabajo del grupo. Cabe aclarar que esto no estaba incluido en las actividades del procedimiento propuesto, sin embargo se consideró apropiado invertir este esfuerzo adicional para el éxito del caso de estudio.

Para obtener el total de horas-persona que invirtió el grupo durante todo el proyecto se empleó las siguientes fórmulas:

$$Total\ grupo\ proyecto\ II = cantidad\ de\ semanas * tiempo\ individual * cantidad\ de\ personas$$

Ecuación 3. Esfuerzo por mes de grupo proyecto II

$$Total = Total\ grupo\ proyecto\ II + Esf.\ Lider$$

Ecuación 4. Esfuerzo por periodo mensual

Mes	Cantidad de semanas	Tiempo individual invertido por semana (horas)	Cantidad de personas	Total GA (horas-persona)	Esf.Lider (horas persona)	Total (horas persona)
Agosto	2	9	7	126	5	131
Septiembre	4	9	7	252	3,59	255,59
Octubre	4	9	7	252	6,59	258,59
Noviembre	4	9	7	252	4,43	256,43
Diciembre	3	9	7	189	1,5	190,5
<b>Esfuerzo total del proyecto</b>						<b>1092,11</b>

Tabla 25. Esfuerzo total del proyecto

A partir de las evidencias mostradas anteriormente se puede concluir que el esfuerzo requerido para realizar las actividades del procedimiento fue de 55,30 horas-persona (para un proyecto que duro 4 meses), en relación a 1092,11 horas-personas empleadas para todo el proyecto. Esto indica que el esfuerzo requerido para la ejecución del



## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

procedimiento es adecuado para las pequeñas organizaciones, ya que requiere el 5,07% del esfuerzo total empleado en el proyecto.

Los datos mostrados anteriormente contestan de manera positiva la pregunta de investigación principal (PP) y adicional (PA1), dado que los tiempos empleados y los beneficios obtenidos al ejecutar las actividades y usar las técnicas sugeridas permitió identificar las causas de los defectos en el producto software. Al mismo tiempo, los recursos humanos y el tiempo empleado son óptimos, ya que las VSEs no cuentan con recursos económicos y humanos para estas actividades.

### 4.6.2. Causas encontradas

La ejecución del procedimiento permitió evidenciar las siguientes causas.

Fase	Causas encontradas
Fase 1	<ol style="list-style-type: none"> <li>1. No se ha establecido un medio para acordar las fechas de las reuniones</li> <li>2. Falta de motivación</li> <li>3. Falta de tiempo</li> <li>4. Carga académica (Algunos miembros del equipo están cursando hasta 8 materias en el semestre)</li> <li>5. No se ha realizado el plan de comunicaciones</li> <li>6. No se ha establecido fechas límites para las entregas</li> <li>7. Cultura (para los estudiantes es común llegar tarde a las reuniones establecidas)</li> <li>8. Falta de responsabilidad</li> <li>9. No se ha establecido fechas límites para las entregas</li> <li>10. No se ha definido los mecanismos de comunicación</li> <li>11. Falta de costumbre</li> <li>12. No se ha realizado el plan de comunicaciones</li> <li>13. No se sigue el procedimiento</li> <li>14. El gerente del proyecto no ha establecido un mecanismo para asignar tareas</li> <li>15. No se reporta la finalización de las tareas</li> <li>16. Los integrantes del grupo no cuentan con un medio para verificar las tareas que se les ha asignado por cada semana</li> <li>17. Los documentos no son actualizados en el repositorio (al hacer cambios no se actualizan en el repositorio)</li> <li>18. Inexperiencia del analista</li> <li>19. No se informa a los interesados los cambios realizados</li> <li>20. La línea de aprendizaje en el framework ha sido grande</li> <li>21. Falta de capacitación (por ejemplo en la gerencia de proyectos y en el levantamiento de requisitos)</li> <li>22. No se dedica las horas estipuladas a la asignatura</li> <li>23. No se conoce el funcionamiento de la tecnología empleada</li> <li>24. El grupo no logra adaptarse a la metodología</li> </ol>

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<p>25. No se realizó un diseño previo de las entrevistas</p> <p>26. No se tiene claro lo que desea el cliente, hay diferentes interpretaciones entre los miembros del equipo</p> <p>27. Falta de experiencia en el levantamiento de requisitos</p>
Fase 2	<p>1. No se han definido entregas al final de cada reunión</p> <p>2. Falta de motivación</p> <p>3. Falta de responsabilidad</p> <p>4. Se hicieron cambios en el diseño de la base de datos</p> <p>5. Falta de conocimiento en el framework</p> <p>6. Se sube al repositorio código que no está funcionando y afecta otras funcionalidades de la aplicación</p> <p>7. Se cuenta con un solo desarrollador, además del desarrollador Front end</p> <p>8. Falta de coordinación entre los desarrolladores</p> <p>9. Presión en las entregas (plazo muy cortos respecto a los productos que se comprometieron a entregar)</p> <p>10. Los programadores no están borrando funciones que no son útiles para el sistema</p> <p>11. Los desarrolladores no entregaron el primer módulo en la fecha establecida</p> <p>12. No se integró en el repositorio las diferentes funcionalidades</p> <p>13. Las pruebas se realizaron antes de hacer la integración</p>
Fase 3	<p>1. Falta de conocimiento en AJAX</p> <p>2. Los desarrolladores no habían definido estándares para programar</p> <p>3. Además de modificaciones previas a la base de datos fue necesario realizar nuevamente cambios</p> <p>4. Uno de los programadores tuvo problemas de salud</p> <p>5. El documento de requisitos estaba especificado a muy alto nivel</p> <p>6. Solo se manejó documento de requisitos, no se realizó casos de uso detallados</p> <p>7. Presión en las entregas</p> <p>8. No se empleó Ajax para el llamado de los calendarios, por este motivo se muestran en inglés</p> <p>9. Falta de conocimiento en las actividades definidas en MoProSoft</p>
Fase 4	<p>1. Falta de tiempo para hacer pruebas de caja negra</p> <p>2. Los desarrolladores no prueban la funcionalidad que tienen a cargo</p> <p>3. Fue necesario hacer nuevamente cambios a la base de datos</p> <p>4. Inconsistencia en la ortografía</p> <p>5. Incorrecta estimación en el tiempo requerido por cada tarea</p> <p>6. Dos de los programadores tuvieron percances de salud</p> <p>7. Se tenía un retraso del 50% en la ejecución del proyecto</p> <p>8. No se tuvo en cuenta validaciones requeridas en los campo contraseña</p> <p>9. Los programadores no diferenciaron en la aplicación las actividades de las tareas</p> <p>10. Los mensajes mostrados al usuario son confusos</p> <p>11. En algunos campos al editar, eliminar o guardar no se presentan mensajes de confirmación</p> <p>12. No se siguieron estándares, por ejemplo colocar * en los campos obligatorios</p> <p>13. Falta de mensajes que ayude al usuario a identificar los campos que no han sido llamados y que son obligatorios</p> <p>14. Falta de mensajes que indiquen brevemente lo que se va hacer al dar click sobre un botón</p> <p>15. Falta un mensaje que indique al usuario el ingreso incorrecto de correo o contraseña</p>

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

	<p>16. No se estableció un estándar para editar las actividades, tareas y categorías (por ejemplo para editar una actividad es necesario dar click sobre el nombre, lo que representa confusión y des-ubicación para el usuario)</p> <p>17. La información no se muestra de manera consistente (por ejemplo el formulario de autenticación se titula "Inicio de sesión", y en el extremo superior derecho dice "login", asimismo al ingresar dice "logout" )</p>
--	--

Tabla 26. Lista de causas encontradas por fase

En la Tabla 27 se muestra el total de causas encontradas en el procedimiento.

Fase	Cantidad de causas encontradas
Fase 1	27
Fase 2	13
Fase 3	9
Fase 4	17
Total	65

Tabla 27. Cantidad de causas encontradas por fase

A partir de los datos consolidados en la Tabla 27 se obtiene la gráfica de la cantidad de causas encontradas por fase.

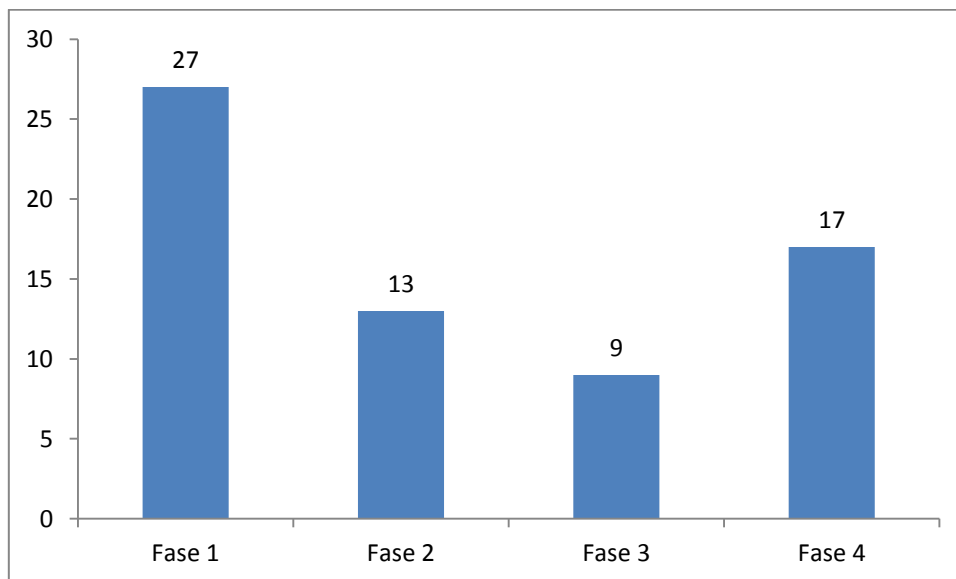


Figura 11. Cantidad de causas encontradas por fase

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Después de la reunión de análisis causal empleada para hallar las causas de los defectos, el líder realizaba las recomendaciones y establecía las personas responsables de los cambios, asimismo fijaba fechas límites para la entrega de éstos. De esta manera se verificaba si los cambios solicitados se habían implementado en las fechas acordadas. A pesar de que se hicieron recomendaciones para eliminar las causas que provocaban los defectos y se establecieron fechas límites para realizar los cambios, algunos de éstos no fueron realizados en las fechas acordadas, razón por la cual ciertas causas persistieron durante el proyecto, provocando retrasos durante las diferentes fases.

Como se puede ver en la Tabla 26, causas tales como la falta de motivación y responsabilidad fueron recurrentes en cada fase, esto podría haberse dado debido a que este proyecto estaba relacionado con un tema académico, por lo cual no existía una presión laboral, además los miembros del equipo no tenían experiencia en la ejecución de este tipo de proyectos.

De otra parte, es de resaltar los beneficios obtenidos gracias a las reuniones que permitían discutir las posibles causas de los defectos. Éstas ayudaron a que los miembros del equipo se concientizaran sobre algunos aspectos y se comprometieran a mejorarlos. Durante este tiempo de retroalimentación los participantes manifestaron sus inconformidades y ayudaron a realizar el diagrama de Ishikawa. Como resultado de estas reuniones fue posible eliminar algunas causas, por ejemplo las relacionadas con la comunicación, para las cuales se estableció los mecanismos recomendados, asimismo se hicieron aclaraciones sobre los medios válidos y como éstos debían ser usados.

También, durante las reuniones se llegó a la conclusión de que gran parte de los retrasos presentados en las entregas se debía a que los programadores no contaban con los conocimientos suficientes en el manejo del framework, por lo cual el gerente del proyecto realizó una capacitación (dado que éste conocía mejor el framework), de la cual se obtuvo buenos resultados, ya que fue posible disminuir en un 25% el porcentaje de retraso llevado hasta la fase 3.

Otra de las situaciones identificadas fue la re-estructuración del grupo, debido a que los desarrolladores no podían cumplir con las entregas acordadas, puesto que era demasiado trabajo para tres personas, una de las cuales asumía el rol de desarrollador front End. Por consiguiente el gerente del proyecto asumió también el rol de

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

desarrollador durante la tercera y cuarta fase. Asimismo, la persona encargada del análisis de requerimientos apoyo tareas relacionadas con desarrollo y gestión.

En este sentido, se puede decir que el procedimiento de análisis causal de defectos enfocado en pequeñas organizaciones de desarrollo software permitió evidenciar una gran cantidad de causas, incluyendo aquí algunas categorías identificadas: comunicación, personas, herramientas, entre otras, de las cuales se discutió su impacto en el proyecto. Además permitió reducirlas o eliminarlas y contribuir en la mejora del producto software, respondiendo la pregunta de investigación adicional (PA2), planteada en este caso de estudio.

### 4.6.3. Percepción de las personas involucradas en el proyecto

A continuación se muestra la percepción de las personas involucradas en el proyecto, la cual se obtuvo mediante una entrevista realizada a cada uno de los miembros del grupo al finalizar el proyecto (Anexo 5). La Tabla 28 presenta las preguntas efectuadas y las respuestas obtenidas.

<b>¿El procedimiento permitió evidenciar los defectos?</b>	
# de respuestas positivas	5
# de respuestas negativas	2
Observaciones	<ul style="list-style-type: none"> <li>❖ El procedimiento no es sobresaliente, ya que existen metodologías que incluyen reuniones de retroalimentación.</li> <li>❖ Sirvió para evidenciar algunos defectos básicos.</li> <li>❖ En gran parte los defectos fueron evidenciados por el gestor del proyecto.</li> </ul>
<b>¿El procedimiento permitió evidenciar las causas de los defectos?</b>	
# de respuestas positivas	6
# de respuestas negativas	1
Observaciones	<ul style="list-style-type: none"> <li>❖ Evidenciar los defectos depende de la sinceridad de las personas y su capacidad de expresión durante las reuniones</li> </ul>
<b>¿La técnica de Ishikawa fue de fácil uso?</b>	
# de respuestas positivas	4
# de respuestas negativas	3
Observaciones	<ul style="list-style-type: none"> <li>❖ Mediante la técnica es difícil detectar la causa de los defectos</li> <li>❖ El diagrama es fácil de entender y realizar</li> <li>❖ El trabajo grupal facilita hallar las causas</li> <li>❖ No se profundizó en su construcción y su estudio</li> </ul>
<b>¿La técnica de diagramas de afinidad fue de fácil uso?</b>	
# de respuestas positivas	7
# de respuestas negativas	0
Observaciones	Sin observaciones

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

<b>Aspectos positivos y aspectos a mejorar en el procedimiento</b>	
<ul style="list-style-type: none"> <li>❖ El procedimiento es bueno, ya que tiene en cuenta aspectos sociales y psicológicos de las personas involucradas</li> <li>❖ Permite obtener una retroalimentación durante el desarrollo</li> <li>❖ Ayuda a detectar causas desconocidas</li> <li>❖ Las reuniones realizadas y las técnicas empleadas fueron de utilidad para encontrar algunas causas que provocan defectos</li> <li>❖ Trabajar en grupo facilita la detección de defectos</li> </ul>	<ul style="list-style-type: none"> <li>❖ Se confunden las causas con los defectos</li> <li>❖ La ejecución de las acciones correctivas debe tener un seguimiento más profundo</li> <li>❖ Discutir los problemas con todos los miembros del grupo puede generar más problemas</li> </ul>
<b>¿El procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software ha sido útil para el desarrollo del proyecto?</b>	
# de respuestas positivas	7
# de respuestas negativas	0
Observaciones	<ul style="list-style-type: none"> <li>❖ Permite realizar correcciones durante el desarrollo y tomar acciones correctivas</li> <li>❖ Ayudó a planear y ejecutar acciones correctivas</li> <li>❖ Es muy útil para detectar causas que algunos desconocen totalmente y defectos insospechados</li> </ul>
<b>¿El procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software ha sido fácil de llevar a cabo?</b>	
# de respuestas positivas	7
# de respuestas negativas	0
Observaciones	<ul style="list-style-type: none"> <li>❖ Se pueden detectar con facilidad las causas y defectos dentro del desarrollo del proyecto y las posibles soluciones a implementar para el mejoramiento</li> <li>❖ El tiempo invertido es relativamente corto</li> </ul>

Tabla 28. Percepción de los involucrados en el proyecto

Como se puede ver en la Tabla 28 la entrevista realizada para conocer la percepción de las personas involucradas en el proyecto, revela como éstos en su totalidad han respondido que el procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software ha sido útil y fácil de llevar a cabo. También se muestra que para los entrevistados los diagramas de afinidad son de fácil uso.

Un aspecto a analizar es si el procedimiento permitió evidenciar los defectos, obteniendo un resultado de 5 respuestas positivas y 2 negativas. Es importante aclarar que el objetivo del procedimiento no es detectar defectos sino sus causas, sin embargo el procedimiento puede ayudar a detectar algunos defectos relacionados con la ejecución del proyecto. Por esta razón se decidió incluir esta pregunta en la entrevista.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

Para la pregunta relacionada con la técnica de Ishikawa, se obtuvo un resultado de 4 respuestas positivas y 3 respuestas negativas, las observaciones realizadas hacían referencia a que no se profundizó en su construcción y estudio. Dadas las observaciones realizadas se incluyó en la actividad preparación una tarea relacionada con la capacitación en las técnicas empleadas.

Otro aspecto a tener en cuenta son los aspectos a mejorar sugeridos por los entrevistados, dadas estas sugerencias se incluyó en el procedimiento (actividad Detección de causas fundamentales) la tarea iniciar solicitud de cambios, con el fin de formalizar los cambios necesarios para reducir o eliminar las causas de los defectos.

Con base en los resultados obtenidos, en relación con la utilidad y facilidad de implementación del procedimiento y con las observaciones expresadas por cada uno de los participantes se puede decir que el procedimiento puede ser adecuado para las pequeñas organizaciones, lo cual responde a la pregunta de investigación principal planteada en este caso de estudio (PA).

### 4.7. Ejemplo

A continuación se muestra de forma general un ejemplo de defectos discutidos y analizados para encontrar la causa que los provocaban:

❖ **Defecto:** No se asigna tareas para desarrollarse durante la semana

❖ **Causa encontradas**

**Método:** No se sigue el procedimiento.

**Personas:** los miembros del equipo no cuentan con un mecanismo donde puedan verificar sus tareas, No se reporta la finalización de las tareas, No se respeta la jerarquía (por ejemplo los programadores deben informar sobre su trabajo al líder de desarrollo).

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### ❖ **Recomendaciones establecidas**

Establecer el medio por el cual se notifica a los integrantes del proyecto los cambios en las fechas de las reuniones o días adicionales en que se deben reunir.

Realizar una plantilla donde se especifique por semana las tareas que se deben realizar, el responsable y la fecha límite de entrega. Ésta debe ser publicada semanalmente.

Definir la red social o medio de comunicación por la cual se va a informar sobre las fechas de las reuniones, cambios realizados y tareas terminadas.

Cada miembro del equipo debe colocar en el medio de comunicación establecido, las tareas que ha finalizado, esto con el fin de que se le pueda asignar nuevas tareas.

En el anexo 7 se encuentra en detalle el proceso de discusión, análisis, hallazgo de causa y recomendaciones dadas durante la ejecución del caso de estudio.



# Capítulo 5. Conclusiones, trabajos futuros y lecciones aprendidas

---

En este capítulo se presenta las conclusiones, los trabajos futuros y las lecciones aprendidas generadas durante la creación del procedimiento y la ejecución del caso de estudio. En este trabajo de grado se desarrolló un procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software que guía a las VSEs en cómo llevar el análisis causal de manera sistemática. El procedimiento consta de cuatro actividades obtenidas a partir del análisis de referentes internacionales y siguiendo la estrategia de investigación basada en investigación-Acción.

## 5.1. Conclusiones

A partir del trabajo desarrollado se presentan una serie de conclusiones resultado de la creación de la propuesta, entre la cuales se destaca la importancia del procedimiento que proporciona una descripción detallada de cada actividad y tarea. Además de contar con diagramas que muestran el flujo a seguir, los roles involucrados y los productos de trabajo obtenidos. Además este trabajo ha desarrollado las plantillas que guían a los responsables de cada actividad en el registro de la información requerida.

El trabajo realizado pudo constatar la importancia de ejecutar el análisis causal dentro de las organizaciones, lo que demuestra la relevancia del procedimiento propuesto para el análisis causal enfocado en pequeñas organizaciones de desarrollo software, ya que cuando se lleva a cabo es posible reducir los defectos y por ende aumentar la calidad del producto. En este sentido, el procedimiento contribuye en la identificación de las causas que están generando los defectos, así como a tomar acciones correctivas para eliminarlas o reducirlas.

Al crear el procedimiento se logró identificar que:

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- ❖ Fue importante analizar para cada una de las técnicas sus ventajas, desventajas y funcionamiento, ya que esto permitió seleccionar las técnicas óptimas para las VSEs.
- ❖ El análisis de los referentes internacionales CMMI-DEV, ISO/IEC 12207 e ISO/IEC 15504, permitió establecer las prácticas/actividades que se debían tener en cuenta en el procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software. Gracias a lo anterior fue posible tener una versión inicial del procedimiento, la cual fue refinado durante la ejecución del caso de estudio. Asimismo se hicieron mejoras con los aportes y observaciones dados por los involucrados en el caso de estudio.
- ❖ Modelar el procedimiento mediante BPMN proporciona una notación estándar, adecuada para todos los involucrados, lo que permite cerrar la brecha entre el diseño del procedimiento y su implementación, además posibilita la visibilidad del modelo, debido a que está disponible en un sitio web: <http://artemisa.unicauca.edu.co/~kyzuniga/index.html>

Respecto al caso de estudio se puede decir que:

- ❖ Permitted mejorar el procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software, ya que se observó situaciones que no se habían tenido en cuenta en la versión inicial del procedimiento. Situaciones tales como i) los defectos relacionados con la ejecución del procedimiento, los cuales pueden ser el origen de los defectos en el producto software, ii) agregar la técnica FMEA que brinda tres criterios para priorizar los defectos, lo cual permite dar mayor rigurosidad a la tarea *Determina muestra de los defectos*, iii) adición de la tarea *Iniciar solicitud de cambio* con el fin de formalizar los cambios requeridos para eliminar o reducir las causas y adición de la tarea *Capacitación de técnicas* incluida con el propósito de que el personal que hace parte del grupo de análisis causal cuente con conocimientos previos en el uso de las técnicas.
- ❖ Se aplicó en un grupo pequeño logrando ver sus ventajas en grupos que tienen recursos y personal limitado. El caso de estudio permitió hallar las causas de los defectos e implementar acciones para eliminarlas o reducirlas.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- ❖ Aplicar un caso de estudio sobre un entorno académico puede generar situaciones en las que los estudiantes no tengan una presión laboral que los motive a seguir las instrucciones dadas durante la ejecución del procedimiento. Por ejemplo para este caso en particular en algunas ocasiones los estudiantes responsables de determinados cambios no aplicaron éstos en las fechas acordadas.

Finalmente se puede concluir que el procedimiento propuesto contribuye en la identificación de las causas de los defectos, y en toma de acciones correctivas para eliminarlas o reducirlas y evitar la manifestación continua del mismo tipo de defecto.

### **5.2. Trabajos futuros**

- ❖ Desarrollar una aplicación web que contenga la documentación de procedimiento con sus respectivas guías, además de permitir crear los diagramas de Ishikawa, diagramas de afinidad y gráficos radiales. También es importante que la aplicación tenga un repositorio donde se pueda almacenar la documentación obtenida en los diferentes proyectos.
- ❖ Ejecutar nuevamente el caso de estudio en grupos académicos con el fin de constatar si la documentación obtenida en proyectos anterior ayuda a disminuir o no repetir los defectos presentados previamente.
- ❖ Realizar la validación del procedimiento con un caso de estudio en un entorno empresarial real, con el fin de comprobar la viabilidad de las actividades y técnicas sugeridas en esta investigación.
- ❖ Verificar el esfuerzo que requiere la actividad Preparación, debido a que en el caso de estudio ejecutado en este trabajo, esta actividad se omitió, porque se contaba con la persona capacitada en el procedimiento y el uso de las técnicas sugeridas.
- ❖ Al procedimiento propuesto se le podría hacer mejoras cuando los referentes internacionales sobre los cuales se ha desarrollado la investigación presenten actualizaciones. El procedimiento está sujeto a cambios en los siguientes elementos: actividades, tareas, productos de trabajo y roles.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### **5.3. Lecciones aprendidas**

A continuación se muestran un conjunto de lecciones aprendidas que se considera importante para este trabajo de grado.

- ❖ El ante-proyecto fue realizado durante seis meses, tiempo en el cual se realizó la búsqueda, compilación y síntesis de trabajos relacionados, por lo tanto se considera que esta fase fue muy importante, debido a que permitió introducirse en la temática y entrar en detalle en el tema a abordar.
- ❖ La estrategia Investigación-Acción bifurcada facilitó la culminación del presente trabajo de grado, dado que los diferentes ciclos definidos en el anteproyecto, fueron los lineamientos principales para identificar el estado actual de la temática investigada (ciclo conceptual), la creación del procedimiento (ciclo metodológico) y la evaluación de éste (ciclo técnico).
- ❖ Comparar las actividades definidas en referentes internacionales fue fácil debido a que los referentes analizados tenían estructuras similares, lo cual facilitó la trazabilidad entre las actividades o prácticas definidas en éstos.
- ❖ Es importante llevar a cabo casos de estudio que permitan ver la aplicación de cualquier propuesta en un contexto real, con el fin de refinar, evaluar y determinar si la propuesta es adecuada.
- ❖ Para casos de estudio en un entorno académico es importante concientizar, motivar y expresar los beneficios obtenidos a las personas involucradas en el proyecto, ya que de esto depende su colaboración y gran parte del éxito del caso de estudio.
- ❖ Al momento de planificar un caso de estudio el cronograma no debe quedar tan ajustado, sino que se deben incluir holguras para gestionar imprevistos.
- ❖ La tarea de capacitación es fundamental para la aplicación del procedimiento en una empresa real, ya que esto facilita y enriquece la información obtenida en las tareas propuestas por cada actividad.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

### REFERENCIAS

- [1] F. J. Pino, F. García, and M. Piattini, "Software process improvement in small and medium software enterprises: a systematic review," *Softw. Qual. J.*, vol. 16, no. 2, pp. 237–261, Nov. 2007.
- [2] F. J. Pino, F. Garcia, and M. Piattini, "Key processes to start software process improvement in small companies," *En Proc. 2009 ACM Symp. Appl. Comput.*, pp. 509–516, 2009.
- [3] N. Habra, S. Alexandre, J.-M. Desharnais, C. Y. Laporte, and A. Renault, "Initiating software process improvement in very small enterprises: Experience with a light assessment tool," *Inf. Softw. Technol.*, vol. 50, no. 7–8, pp. 763–771, Jun. 2008.
- [4] G. Czibula, Z. Marian, and I. G. Czibula, "Software defect prediction using relational association rule mining," *Inf. Sci. (Ny)*, vol. 264, no. 0, pp. 260–278, Apr. 2014.
- [5] C.-P. Chang and C.-P. Chu, "Defect prevention in software processes: An action-based approach," *J. Syst. Softw.*, vol. 80, no. 4, pp. 559–570, Apr. 2007.
- [6] T. O. a. Lehtinen, M. V. Mäntylä, and J. Vanhanen, "Development and evaluation of a lightweight root cause analysis method (ARCA method) – Field studies at four software companies," *Inf. Softw. Technol.*, vol. 53, no. 10, pp. 1045–1061, Oct. 2011.
- [7] SEI, *CMMI for development (CMMI DEV)(CMU/SEI-2010-TR-033), version 1.3*.
- [8] ISO/IEC, *ISO/IEC 12207, System and software engineering - software life cycle processes, 2008*.
- [9] ISO/IEC, *ISO/IEC FDIS 15504, An exemplar software life cycle process assessment model, 2011*.
- [10] G. Eckes, *the six sigma revolution: how general electric and other turned procces into profits. John Wiley & Sons. 2002*.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- [11] M. Leszak, D. E. Perry, and D. Stoll, "A case study in root cause defect analysis," *Proc. 2000 Int. Conf. Softw. Eng. ICSE 2000 New Millenn.*, pp. 428–437, 2000.
- [12] M. Kalinowski, G. H. Travassos, and D. N. Card, "Towards a Defect Prevention Based Process Improvement Approach," *2008 34th Euromicro Conf. Softw. Eng. Adv. Appl.*, pp. 199–206, Sep. 2008.
- [13] C.-P. Chang and C.-P. Chu, "Improvement of causal analysis using multivariate statistical process control," *Softw. Qual. J.*, vol. 16, no. 3, pp. 377–409, Jan. 2008.
- [14] S. Arreche and S. Matalonga, "Tools for defect causal analysis," *Inf. Syst. Technol. (CISTI), 2012 7th Iber. Conf.*, p. 1,7, 2012.
- [15] M. G. S. Gonçalves, "MiniDMAIC : An Approach for Causal Analysis and Resolution in Software Development Projects," pp. 166–171, 2008.
- [16] C. R. Nelms, "The problem with root cause analysis," *2007 IEEE 8th Hum. Factors Power Plants HPRCT 13th Annu. Meet.*, pp. 253–258, Aug. 2007.
- [17] D. N. Card, "Learning from Our Mistakes with Defect Causal Analysis," *software, IEEE*, vol. 15, no. 1, p. 56,63, 1998.
- [18] M. Kalinowski, D. N. Card, and G. H. Travassos, "Evidence-Based Guidelines to Defect Causal Analysis," *IEEE Softw.*, vol. 29, no. 4, pp. 16–18, Jul. 2012.
- [19] F. J. Pino, M. Piattini, and G. H. Travassos, "Managing and developing distributed research projects in software engineering by means of action-research," *Rev. Fac. Ing.*, no. 68, pp. 61–74, 2013.
- [20] Object Management Group (OMG), "Business Process Model and Notation (BPMN)," 2011. [Online]. Available: <http://www.omg.org/spec/BPMN/2.0>.
- [21] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," pp. 1–10, 2007.
- [22] S. Cromar, "Root Cause Analysis," in *From Techie to Boss*, Springer, 2013, pp. 135–161.
- [23] G. Eckes, *The Six Sigma revolution: How General Electric and others turned process into profits*. 2002.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- [24] P. Henry, "The Analysis of Defect Root Causes," in *The Testing Network SE - 7*, Springer Berlin Heidelberg, 2008, pp. 235–296.
- [25] R. Chillarege, S. Member, I. S. Bhandari, K. Jarir, M. J. Halliday, D. S. Moebus, B. K. Ray, and M. Wong, "Orthogonal Defect Classification-A Concept for In-Process Measurements," vol. 18, no. 11, 1992.
- [26] J. Huang, LiGuo and Ng, Vincent and Persing, Isaac and Geng, Ruili and Bai, Xu and Tian, "AutoODC: Automated generation of Orthogonal Defect Classifications," *Autom. Softw. Eng. (ASE), 2011 26th IEEE/ACM Int. Conf.*, pp. 412–415, Nov. 2011.
- [27] A. Gupta, J. Li, R. Conradi, H. Rønneberg, and E. Landre, "A case study comparing defect profiles of a reused framework and of applications reusing it," *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 227–255, Aug. 2008.
- [28] IBM, "Orthogonal Defect Classification v 5.2 Extensions for GUI, User Documentation, Build & NLS." 2013.
- [29] ISO/IEC, *ISO/IEC DTR 29110-1, Software Engineering — Lifecycle Profiles for Very Small Entities (VSEs) — Part 1: Overview*. 2009.
- [30] *ISO/IEC JCT1/SC7 Working Group 24 Deployment Packages repository*, <http://profs.logti.etsmtl.ca/claporte/English/VSE/index.html>. .
- [31] V. Ribaud, P. Saliou, R. O'Connor, and C. Laporte, "Software Engineering Support Activities for Very Small Entities," in *Systems, Software and Services Process Improvement SE - 15*, vol. 99, A. Riel, R. O'Connor, S. Tichkiewitch, and R. Messnarz, Eds. Springer Berlin Heidelberg, 2010, pp. 165–176.
- [32] F. J. Pino, F. Garcia, F. Ruiz, and M. Piattini, "Adaptación de las normas ISO / IEC 12207 : 2002 e ISO / IEC 15504 : 2003 para la evaluación de la madurez de procesos software en países en desarrollo," *En JISBD*, pp. 187–194, 2005.
- [33] K. C. P. Zuluaga, "Estudio del comportamiento de la industria del software en Colombia ante escenarios de capacidades de innovación y ventajas comparativas por medio de dinámica de sistemas," 2011.

## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- [34] P. N. Colombian, Southwest and Case, “Caracterización de las Empresas Pertencientes a la Industria Emergente de Software del Sur Occidente Colombiano Caso Red de Parques PARQUESOFT Characterization of Companies that Belong to,” *Rev. Av. en Sist. e Informática*, vol. 4, no. 2, 2007.
- [35] FEDESOFTE, “La industria de software colombiana,” 2012. [Online]. Available: <http://fedesoft.org/la-industria-de-software-colombiana/>.
- [36] T. O. A. Lehtinen, R. Virtanen, J. O. Viljanen, M. V Mäntylä, and C. Lassenius, “A tool supporting root cause analysis for synchronous retrospectives in distributed software teams,” *Inf. Softw. Technol.*, vol. 56, no. 4, pp. 408–437, Apr. 2014.
- [37] N. A. Pankaj Jalote, “Using defect analysis feedback for improving quality and productivity in iterative software development,” *Inf. Commun. Technol. 2005. Enabling Technologies New Knowl. Soc. ITI 3rd Int. Conf.*, pp. 701–714, 2005.
- [38] H. Jabrouni, B. Kamsu-foguem, L. Geneste, and C. Vaysse, “Engineering Applications of Artificial Intelligence Continuous improvement through knowledge-guided analysis in experience feedback,” *Eng. Appl. Artif. Intell.*, vol. 24, no. 8, pp. 1419–1431, 2011.
- [39] N. Honda and S. Yamada, “‘Defect Root-Cause Analysis and 1+n Procedure’ technique to improve software quality,” *Int. J. Syst. Assur. Eng. Manag.*, vol. 3, no. 2, pp. 111–121, Jul. 2012.
- [40] F. J. Pino, M. T. Baldassarre, M. Piattini, and G. Visaggio, “Harmonizing maturity levels from CMMI-DEV and ISO / IEC 15504,” *J. Softw. Maint. Evol. Res. Pract.*, vol. 22, no. 4, pp. 279–296, 2010.
- [41] M. E. Cournoyer, S. Trujillo, S. Schreiber, M. T. Saba, and M. C. Peabody, “Causal analysis of a glovebox glove breach,” *J. Chem. Heal. Saf.*, vol. 20, no. 2, pp. 25–33, Mar. 2013.
- [42] R. B. Grady, “Software Failure Analysis for High-Return Process Improvement,” *Hewlett Packard J.*, vol. 47, pp. 15–24, 1996.
- [43] Y. Chernak and I. C. Society, “A statical Approach to the Inspection ormal Synthesis and Improvement,” *Softw. Eng. IEEE Trans.*, vol. 22, no. 12, pp. 866–874, 1996.



## Procedimiento de análisis causal enfocado en pequeñas organizaciones de desarrollo software

- [44] C. Shrouti, P. Franciosa, and D. Ceglarek, "Root Cause Analysis of Product Service Failure Using Computer Experimentation Technique," *Procedia CIRP*, vol. 11, no. 0, pp. 44–49, 2013.
- [45] A. Jayswal, X. Li, A. Zanwar, H. H. Lou, and Y. Huang, "A sustainability root cause analysis methodology and its application," *Comput. Chem. Eng.*, vol. 35, no. 12, pp. 2786–2798, Dec. 2011.
- [46] F. O. Bjørnson, A. I. Wang, and E. Arisholm, "Improving the effectiveness of root cause analysis in post mortem analysis: A controlled experiment," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 150–161, Jan. 2009.
- [47] C. Henderson, "Managing software defects: defect analysis and traceability," *ACM SIGSOFT Softw. Eng. Notes*, vol. 33, no. 4, p. 1, Jul. 2008.
- [48] D. N. Card and C. S. Corporation, "Defect-causal analysis drives down error rates," *Software, IEEE*, vol. 10, no. July, pp. 98–99, 1993.
- [49] R. G. Mays, "Applications of defect prevention in software development," *IEEE J. Sel. Areas Commun.*, vol. 8, no. 2, pp. 164–168, 1990.
- [50] M. Kalinowski, E. Mendes, D. N. Card, and G. H. Travassos, "Applying DPPI : A Defect Causal Analysis Approach Using Bayesian Networks," pp. 92–106, 2010.
- [51] C. and CYTED, *COMPETISOFT-Mejora de procesos para fomentar la competitividad de la pequeña y mediana industria del software de Iberoamérica*. 2006.
- [52] Z. Brereton, Pearl and Kitchenham, Barbara and Budgen, David and Li, "Using a Protocol Template for Case Study Planning," *Proc. 12th Int. Conf. Eval. Assess. Softw. Eng. Univ. Bari, Italy*, 2008.
- [53] R. K. Yin, *Case study research: Design and methods*. Sage publications, 2014.
- [54] H. Background, "B+-Tree," *Encyclopedia of Database System*. pp. 1998–299, 2009.