

**PROCEDIMIENTO PARA AUTOMATIZACIÓN DE PRUEBAS UNITARIAS EN  
PEQUEÑAS EMPRESAS DESARROLLADORAS DE SOFTWARE**



**MARLON FELIPE MAMIÁN RODRÍGUEZ  
DARWIN JULIÁN MONTENEGRO CAICEDO**

Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Programa de Ingeniería de Sistemas  
Popayán  
2018

**PROCEDIMIENTO PARA AUTOMATIZACIÓN DE PRUEBAS UNITARIAS EN  
PEQUEÑAS EMPRESAS DESARROLLADORAS DE SOFTWARE**

Trabajo de grado para optar al título de:  
Ingeniero de Sistemas

MARLON FELIPE MAMIÁN RODRÍGUEZ  
DARWIN JULIÁN MONTENEGRO CAICEDO

Director:  
Mag. Carlos Alberto Ardila Albarracín

Universidad del Cauca  
Facultad de Ingeniería Electrónica y Telecomunicaciones  
Programa de Ingeniería de Sistemas  
Popayán  
2018

# Tabla de Contenido

Capítulo 1 .....	6
Introducción .....	6
1.1. Planteamiento del problema. ....	6
1.2. Objetivos .....	9
1.3. Solución propuesta .....	9
1.4. Metodología .....	9
1.5. Estructura del documento .....	10
Capítulo 2.....	12
Trabajos relacionados.....	12
2.1. Marco teórico .....	12
2.2. Protocolo de la revisión .....	14
2.3. Resultados de la revisión.....	16
2.4. Trabajos obtenidos mediante revisión sistemática.....	18
2.5. Síntesis.....	23
Capítulo 3.....	25
Caracterización de prácticas de automatización de pruebas unitarias en pequeñas empresas desarrolladoras de software.....	25
3.1. Elaboración del instrumento de recolección de información .....	25
3.2. Recolección de información .....	26
3.3. Resultados obtenidos .....	26
3.4. Síntesis.....	38
Capítulo 4.....	40
Descripción del procedimiento.....	40
4.1. Descripción conceptual del procedimiento .....	41
4.2. Descripción detallada del procedimiento .....	44
Capítulo 5.....	55
Evaluación del procedimiento .....	55
5.1. Antecedentes .....	55
5.2. Diseño .....	56
5.3. Revisión de expertos.....	57
5.4. Estudio de caso .....	62
5.5. Sujetos de investigación y unidad de análisis. ....	62
5.6. Procedimiento de campo .....	62

<b>5.7. Intervención</b> .....	63
<b>5.8. Recolección de datos</b> .....	75
<b>5.9. Resultados y análisis</b> .....	75
<b>Capítulo 6</b> .....	78
<b>Conclusiones y trabajo futuro</b> .....	78
<b>6.1. Sumario</b> .....	78
<b>6.2. Conclusiones</b> .....	78
<b>6.3. Trabajo futuro y recomendaciones</b> .....	80
<b>Referencias</b> .....	81

## Índice de Tablas

<b>Tabla 2.1.</b>	Planeación de la revisión sistemática.....	15
<b>Tabla 2.2.</b>	Conteo de artículos encontrados.....	16
<b>Tabla 2.3.</b>	Conteo de estudios primarios por categorías estructurales.....	17
<b>Tabla 2.4.</b>	Conteo de estudios primarios por año de publicación.....	18
<b>Tabla 3.1.</b>	Tabla referente a las respuestas de las empresas según los aspectos evaluados.....	36
<b>Tabla 3.2.</b>	Conglomerado de los aspectos evaluados según la calificación dada por las empresas.....	36
<b>Tabla 3.3.</b>	Tabla referente a la calificación de las empresas según los aspectos evaluados.....	37
<b>Tabla 3.4.</b>	Conglomerado de los aspectos evaluados según la calificación dada por las empresas.....	37
<b>Tabla 5.1.</b>	Participantes de la evaluación por expertos.....	57
<b>Tabla 5.2.</b>	Aspectos positivos del procedimiento.....	59
<b>Tabla 5.3.</b>	Aspectos a mejorar en el procedimiento.....	59
<b>Tabla 5.4.</b>	Observaciones realizadas al procedimiento.....	60
<b>Tabla 5.5.</b>	Modificaciones realizadas al procedimiento.....	61
<b>Tabla 5.6.</b>	Listado de plantillas por tareas clave.....	75
<b>Tabla 5.7.</b>	Modificaciones realizadas con el estudio de caso.....	76

## Índice de Figuras

<b>Figura 3.1.</b> Nivel de eficacia de las pruebas unitarias en cuanto a la cantidad de errores encontrados.....	27
<b>Figura 3.2.</b> Mayor factor de incidencia para NO ejecutar pruebas unitarias de software.....	27
<b>Figura 3.3.</b> Nivel de costos respecto al presupuesto del proyecto en el descubrimiento y depuración de errores.....	28
<b>Figura 3.4.</b> Porcentaje del presupuesto de un proyecto destinado para pruebas y depuración.....	28
<b>Figura 3.5.</b> Porcentaje del tiempo asignado en el cronograma que se emplea para hacer pruebas.....	29
<b>Figura 3.6.</b> Metodología de pruebas usada por empresas.....	29
<b>Figura 3.7.</b> Porcentaje de desarrolladores que hacen pruebas unitarias de sus propios códigos.....	30
<b>Figura 3.8.</b> Porcentaje de desarrolladores que usan técnicas de programación por pares o pruebas de rotación de código.....	30
<b>Figura 3.9.</b> Tipo de evaluación que hacen los desarrolladores.....	31
<b>Figura 3.10.</b> Porcentaje que especifica el manejo de pruebas en la empresa.....	31
<b>Figura 3.11.</b> Pruebas unitarias que realizan las empresas.....	32
<b>Figura 3.12.</b> Técnicas de prueba usadas según el enfoque de caja blanca.....	32
<b>Figura 3.13.</b> Porcentaje en el que se incrementan los costos totales en el proyecto a la hora de automatizar.....	33
<b>Figura 3.14.</b> Normas, modelos o propuestas utilizadas para documentar pruebas...	34
<b>Figura 3.15.</b> Aspectos más importantes a la hora de definir la estrategia de pruebas.	34
<b>Figura 3.16.</b> Porcentaje referente al coste de eliminar un defecto se incrementa con el tiempo de permanencia del defecto.....	35
<b>Figura 3.17.</b> Porcentaje referente a los costos de un proyecto en cuanto al presupuesto se disminuye usando herramientas automatizadas.....	35
<b>Figura 4.1.</b> Ciclos de prueba de calidad de software.....	41
<b>Figura 4.2.</b> Elementos principales del procedimiento.....	42
<b>Figura 4.3.</b> Vista general del procedimiento.....	44
<b>Figura 4.4.</b> Tareas de la práctica identificación y selección.....	46

<b>Figura 4.5.</b> Tareas de la práctica de establecimiento de criterios, procedimientos y entornos.....	49
<b>Figura 4.6.</b> Tareas de la práctica preparación, ejecución de la prueba, realización y análisis de la revisión entre pares.....	51
<b>Figura 4.7.</b> Tareas de la práctica verificación, validación y análisis de resultados....	53
<b>Figura 5.1.</b> Diagrama del procedimiento de campo.....	63
<b>Figura 5.2.</b> Diagrama de actividades a ejecutar en el caso de estudio.....	64
<b>Figura 5.3.</b> Ejemplo de la plantilla para listado de métodos.....	65
<b>Figura 5.4.</b> Ejemplo de la plantilla para el listado de productos de trabajo seleccionados.....	66
<b>Figura 5.5.</b> Ejemplo de plantilla personal involucrado para la ejecución de la prueba.	67
<b>Figura 5.6.</b> Script de pruebas del método Login.....	68
<b>Figura 5.7.</b> Primer script de pruebas del método Registrar.....	69
<b>Figura 5.8.</b> Segundo script de pruebas del método Registrar.....	70
<b>Figura 5.9.</b> Script de pruebas del método Crear Agencias.....	71
<b>Figura 5.10.</b> Primer script de pruebas del método Crear Regionales.....	72
<b>Figura 5.11.</b> Segundo script de pruebas del método Crear Regionales.....	73
<b>Figura 5.12.</b> Ejemplo de informe de no conformidades.....	74

# Capítulo 1

## Introducción

En este capítulo se presenta el procedimiento propuesto sobre automatización de pruebas unitarias de software identificando el problema, objetivos, solución propuesta, metodología y la forma en que está organizado el documento.

### 1.1. Planteamiento del problema.

En la industria del software se hace control de calidad por medio de pruebas, para las cuales no siempre se dispone de recursos suficientes y están expuestas al error humano (Candea et al., 2010). Las pruebas de software agrupan un conjunto de actividades dentro del desarrollo de software que permiten verificar y validar un producto software, estas actividades hacen parte del proceso, dependiendo del tipo de pruebas que se vayan a realizar. Para Moradbeiky et al. (2014) la manera de evaluar la confiabilidad de las pruebas software para medir la percepción, observación, seguimiento, control, predicción y examen de proyectos software, se ha convertido en un reto principal de muchas empresas, la precisión y el tiempo son factores de rendimiento principales que se emplean para dicha evaluación.

Se va evidenciando cada vez más la importancia que toma el proceso de la automatización de pruebas en la industria del software; según Moradbeiky et al. (2014) esta técnica ayuda significativamente a la precisión en las pruebas de software y a reducir el tiempo de ejecución, los cuáles son los principales factores de rendimiento de las pruebas. Según Xiaochun et al. (2008) para algunos sectores de la industria del software, el uso de pruebas automatizadas sigue siendo poco rentable en la industria actual por su elevado costo en su implementación además, en el estudio realizado se menciona que en el 50% de los encuestados tienen problemas en las pruebas por su

alto costo, la baja reutilización e inflexibilidad de guiones fijos además del desconocimiento que se tiene de las pruebas unitarias, así como también la mala administración de las mismas. Con toda esta problemática el factor de mayor incidencia en las pruebas unitarias de software es el alto consumo de tiempo en su utilización, esto se puede corroborar fácilmente en el proceso de diseño, desarrollo y ejecución de scripts de pruebas.

Para el servicio de informática de la Universidad de Alicante (SIUA, 2010) con las pruebas unitarias se facilita la vida del desarrollador ya que la calidad de su código mejorará, se reducirán costos de depuración y la corrección de incidencias ya que éstas se enfocan en la lógica del procesamiento interno y de las estructuras de datos. Por otro lado, según Herrmann E. (2011), las pruebas unitarias típicamente son automatizadas, permiten detectar efectivamente los defectos durante fases sucesivas de desarrollo. Estas se concentran en probar cada componente individualmente para asegurar que funcione de manera apropiada utilizando técnicas de prueba que recorren caminos específicos en la estructura de los componentes.

Las pruebas de software (software testing) agrupan un conjunto de actividades dentro del desarrollo de software que permiten verificar y revelar la calidad de un producto software, estas actividades pueden ser implementadas en cualquier momento del proceso, dependiendo del tipo de pruebas que se vayan a realizar.

Para Moradbeikhey et al. (2014) el cómo evaluar la fiabilidad de las pruebas software para medir la percepción, observación, seguimiento, control, predicción y examen de proyectos software se ha convertido en un reto principal de muchas empresas, la precisión y el tiempo son factores de rendimiento principales que se emplean para dicha evaluación.

Se ha evidenciado la importancia que toma el proceso de la automatización de pruebas en la industria del software, ya que esta técnica ayuda significativamente a la precisión y el tiempo de ejecución los cuáles son los principales factores de rendimiento de las pruebas.

Para algunos sectores de la industria del software el uso de pruebas automatizadas sigue siendo poco rentable en la industria actual. Como se puede evidenciar en el

estudio de Xiaochun et al., el 50% de los encuestados mencionan que el costo era el factor de mayor incidencia en la automatización de las pruebas software dado que el pasar por todo el procedimiento de diseño, desarrollo y ejecución de scripts de prueba para cada tarea es muy costoso si llegasen grandes cambios además de que se tiene una baja reutilización e inflexibilidad de los guiones fijos.

Es por esto que se busca la construcción de un procedimiento sobre automatización de pruebas unitarias para pequeñas empresas desarrolladoras de software, donde tengan disponible una guía sobre mejores prácticas en pruebas automatizadas.

De acuerdo al problema descrito anteriormente, surge la siguiente pregunta de investigación: **¿Cómo orientar las actividades para automatizar pruebas unitarias en pequeñas organizaciones desarrolladoras de software?**

Para responder a la pregunta planteada se propone la construcción de un procedimiento sobre automatización de pruebas unitarias para pequeñas organizaciones desarrolladoras de software. En donde según Safronaun y Turlo (2011), muchas empresas de la industria software pierden tiempo y recursos en la comercialización de sus productos, el implementar correctamente la automatización de pruebas unitarias es una manera eficiente de garantizar un tiempo de comercialización más rápido para sus productos, por otro lado, también se habla de los desafíos que afrontan las empresas a la hora de dar una solución a este tipo de pruebas y proporcionan recomendaciones sobre la ejecución de las mismas.

Este procedimiento consiste en una guía en la que se definieron las actividades, roles, entradas, salidas e indicadores básicos para orientar la automatización de pruebas unitarias en pequeñas empresas.

## **1.2. Objetivos**

### **Objetivo general**

Proponer un procedimiento conceptual que oriente la automatización de pruebas unitarias en pequeñas empresas desarrolladoras de software.

### **Objetivos específicos**

- a) Establecer características y necesidades de las pequeñas empresas en cuanto a la adecuación de prácticas de automatización de pruebas unitarias.
- b) Determinar elementos conceptuales y metodológicos sobre buenas prácticas de automatización de pruebas unitarias en pequeñas empresas desarrolladoras de software.
- c) Evaluar la pertinencia para orientar las prácticas, actividades y elementos del procedimiento propuesto mediante un estudio de caso.

## **1.3. Solución propuesta**

Se pretende enfrentar el problema planteado abordando y estudiando los diversos elementos, prácticas, técnicas adecuadas y/o herramientas que se centran en la automatización de pruebas unitarias, creando un procedimiento que permita orientar a esta clase de empresas en como guiar la automatización de pruebas unitarias de manera sistemática, guiando paso a paso la elaboración de los artefactos o productos de trabajo necesarios

## **1.4. Metodología**

Se utilizó el método de investigación-acción estudiado por M. Bocco et al. (2014), el cuál rige el desarrollo del proyecto y se incluye en la etapa de evaluación (observación y reflexión) el estudio de caso, en este método se identifican cuatro etapas, las cuales son:

1. Etapa de planificación: En esta etapa se hace una descripción de los problemas relevantes que guiarán en la investigación tomando como referencia las literaturas en las que se mencionen temas como la automatización de pruebas. En esta etapa también se caracterizan las prácticas actuales en diferentes empresas de la ciudad de Popayán por medio de técnicas de recolección de información.
2. Etapa de acción: En base en lo anterior se crea un procedimiento según los pasos a seguir en las pequeñas empresas para los procesos software además de tomar como referencia las prácticas de las literaturas para realizar un modelo que comprenda todos los pasos necesarios para su automatización.
3. Etapa de observación: En esta etapa se recolectará toda la información que haga referencia a las prácticas de automatización y el manejo de las diferentes pruebas en las pequeñas empresas software (estudio de caso).
4. Etapa de reflexión: En esta etapa se comparan y analizan los resultados obtenidos para ser compartidos con los interesados.

## **1.5. Estructura del documento**

La presente monografía está conformada de la siguiente manera:

Capítulo 2, presenta los trabajos relacionados con pruebas automáticas de software para la cual se realizó una revisión sistemática de la literatura, que es un medio para identificar, evaluar e interpretar investigaciones relevantes disponibles en un área temática.

Capítulo 3, trata aspectos para establecer cuales técnicas o prácticas pueden ser incorporadas con mira a la creación del procedimiento de automatización de pruebas, se presenta los resultados de encuestas realizadas para definir dichas técnicas o prácticas a utilizar.

Capítulo 4, presenta de manera detallada las características, componentes y la documentación que conforma el procedimiento para automatización de pruebas unitarias.

Capítulo 5, presenta el proceso de evaluación del procedimiento realizado mediante la revisión de expertos y el estudio de caso. Se detallan las etapas de este proceso, los resultados obtenidos y como influyeron en la afinación de la propuesta.

Por último, el Capítulo 6, presenta las conclusiones y el trabajo futuro planteado para esta tesis

## **Capítulo 2**

### **Trabajos relacionados**

Para la recopilación de trabajos relacionados se realizó una revisión sistemática de la literatura, que es un medio para identificar, evaluar e interpretar investigaciones relevantes disponibles en un área temática. El principal objetivo fue analizar trabajos relacionados con automatización de pruebas o pruebas unitarias en pequeñas empresas desarrolladoras de software, tales como marcos conceptuales, guías técnicas o procedimientos que indiquen como llevar a cabo la automatización de pruebas.

#### **2.1. Marco teórico**

En la actualidad muchas empresas realizan la calidad de sus productos mediante pruebas software que según PMOinformatica (2012) comprenden el conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación, mediante un proceso básico de pruebas que incluyen Planificación y control, selección de condiciones de prueba, diseño y ejecución de casos de prueba, comprobación de resultados, generación de informes respecto al proceso de prueba y el sistema sujeto a prueba y por ultimo finalización y completar actividades de cierre según lo planteado por Choucair Business Centric Testing (2010). En la industria software se busca que el control de calidad de software sea cada vez más seguro y eficaz, pero según Candea G, Bucur S, Zamfir C. (2010) las pruebas software tienen escasez de recursos, además de consumir mucho tiempo en la implementación y mano de obra, pero están expuestas a la omisión de errores y el error humano.

### **2.1.1. Tipos de pruebas**

Para abarcar las necesidades de la industria software en el ámbito de pruebas se tienen diversos tipos de pruebas como lo son: pruebas unitarias, pruebas de integración, pruebas de sistema y pruebas de aceptación.

**Pruebas unitarias.** Con las pruebas unitarias podemos comprobar que el código fuente esté funcionando como se espera mediante fragmentos de código. La implementación de pruebas unitarias es de gran ayuda para el desarrollador dado que la calidad del código mejorará, se reducen tiempos de depuración y corrección de incidencias. Según Choucair Business Centric Testing (2010) las convenciones de cada lenguaje de programación para la asignación de nombres a sus respectivas pruebas unitarias, se hará referencia a una prueba unitaria como prueba de módulos (“module test”) (por ejemplo, en C), prueba de clase (“class test”) (por ejemplo, en Java o C++) y prueba de unidad (“unit test”) (por ejemplo, en Pascal). En cuanto a las pruebas de componente se dividen en dos fases, la primera incluye bases de prueba las cuales contienen requisitos de componente, diseño detallado y código; la segunda son los objetos de prueba típicos los cuales contienen componentes/clases/unidades/módulos, programas y conversión de datos/migración de programas.

**Pruebas de integración.** Según M. Cillero (2009), las pruebas de integración se realizan una vez el código ha sido probado unitariamente y el objetivo de estas pruebas es verificar el correcto ensamblaje entre los diversos componentes, éstas pruebas se realizan con la finalidad de comprobar la interacción correcta de todos los componentes a través de sus interfaces (internas y externas), además de que se cubren las funcionalidades establecidas y se ajustan a los requisitos no funcionales especificados.

**Pruebas de sistema.** M. Cillero (2009) plantea que el objetivo principal de las pruebas de sistema es comprobar la integración del sistema globalmente, esto se logra verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y el resto de sistemas con los que se comunica, con este tipo de pruebas se verifica que las especificaciones funcionales y técnicas se cumplen y dan una visión del producto software en el entorno de producción.

**Pruebas de aceptación.** M. Cillero (2009) plantea que este tipo de pruebas se realizan para determinar si los requerimientos de una especificación o contrato han sido cumplidos, las pruebas de aceptación implican pruebas de caja negra antes de su entrega definitiva y se realizan principalmente para establecer el grado de confianza de un sistema, esto se determina por su grado de adherencia a las necesidades, requerimientos y procesos de negocio solicitados por el usuario o cliente, es el cliente quien decide si acepta o no el sistema que le está siendo entregado.

## **2.2. Protocolo de la revisión**

En esta sección se presentan el enfoque de la pregunta, el problema y la pregunta de investigación para la presente revisión sistemática, y en la Tabla 2.1 se muestran los demás elementos del protocolo de la revisión en su etapa de planeación.

- (a) Enfoque de la pregunta. La meta fue identificar propuestas relacionadas con automatización de pruebas especialmente en pruebas unitarias en pequeñas o muy pequeñas empresas desarrolladoras de software.
- (b) Problema. Según Candea G. et al. (2010) la industria software busca un control de calidad más seguro y eficaz, dado que las pruebas tienen escasez de recursos, emplean mucho tiempo, mano de obra y están expuestas al error humano.
- (c) Pregunta. ¿Cuáles son los enfoques existentes, marcos conceptuales, guías técnicas o procedimientos sobre automatización de pruebas especialmente pruebas unitarias para pequeñas o muy pequeñas empresas desarrolladoras de software?

Palabras clave y sinónimos	Very small, small, enterprises, organizations, companies, automation, automated, unit testing, software testing, methodology, procedure, framework.
Intervención	Analizar trabajos relacionados con automatización de pruebas unitarias o pruebas unitarias en pequeñas empresas desarrolladoras de software, entre los años 2008 al 2017.
Efecto	Estudiar evidencia sobre automatización de pruebas unitarias en pequeñas empresas desarrolladoras de software.
Población Objetivo	Propuestas de investigación sobre automatización de pruebas unitarias en pequeñas empresas desarrolladoras de software.
Definición de criterios de selección de fuentes	Las fuentes se identificaron a partir del juicio de expertos en el área de investigación, que es el ámbito de la revisión sistemática.
Idioma	Inglés y español.
Identificación de fuentes	Método de búsqueda: La búsqueda de resultados primarios se hizo con el motor de búsqueda de <a href="http://www.scopus.com">www.scopus.com</a>
	Cadenas de búsqueda: <b>CADENA A:</b> "Automated software testing" OR "small software development enterprises"
	<b>CADENA B:</b> "Automated software testing" OR "small software development enterprises" OR "Methodology, procedure, framework"
	<b>CADENA C:</b> "Software Test Automation" OR "small software development enterprises"

**Tabla 2.1.** Planeación de la revisión sistemática

## 2.3. Resultados de la revisión

### 2.3.1. Ejecución de las búsquedas e inspección de referencias

Las búsquedas se efectuaron en el sitio web de Scopus ([www.scopus.com](http://www.scopus.com)) y se encontraron 193 referencias, también se realizaron búsquedas en IEEE Xplore Digital Library donde se encontraron 43 referencias para un total de 236 referencias. Para determinar si los artículos encontrados se relacionaban con propuestas para automatización de pruebas o pruebas unitarias orientadas a pequeñas o muy pequeñas empresas, se utilizó como criterio de exclusión el análisis del título, a partir de la información encontrada se filtró toda aquella que cumpliera con los siguientes criterios de inclusión:

**Tipo de documento.** Revistas, libros, memorias de congreso, memorias de simposios, reportes técnicos que hablen de automatización de pruebas unitarias

**Máximo tiempo de publicación.** 10 años atrás.

**Manejo del tema.** Se revisó en los abstract de los documentos encontrados que la propuesta estuviera enfocada en automatización de pruebas unitarias.

De las 236 referencias encontradas se excluyeron los documentos que no aborden automatización de pruebas unitarias. Una vez aplicado este criterio, se seleccionaron 17 referencias, de las cuales solo 9 de ellas estaban disponibles para descarga gratuita.

En la tabla 2.2 se indican artículos encontrados, artículos seleccionados y artículos disponibles por cadena de búsqueda.

Obtenidos con	Encontrados	Seleccionados	Disponibles
Cadena A	136	10	4
Cadena B	57	7	4
Cadena C	43	0	1
Totales	236	17	9

**Tabla 2.2.** Conteo de artículos encontrados

### 2.3.2. Caracterización de los estudios primarios

Cuando se ingresó las cadenas de búsqueda en los motores de búsqueda como Scopus e IEEE Xplore Digital Library, nos arrojó un total de 236 referencias. A la hora de determinar si los artículos encontrados se relacionaban con el trabajo de investigación que se realiza en este caso, se encontró que la mayoría abarcaba trabajos sobre mutación de pruebas, como mejorar las pruebas software con algoritmos metaheurísticos, automatización de pruebas para aplicaciones móviles, y otros temas que no correspondían con el enfoque que queremos dar en este trabajo, que es establecer un procedimiento que oriente la automatización de pruebas unitarias en aplicaciones tipo escritorio, que incluya una descripción precisa de las prácticas y actividades a ejecutar, cómo documentarlas, las metas a lograr y las métricas que permitan verificar si se alcanzaron esas metas. El nombre de la categoría, su definición y la cantidad de estudios primarios por categoría están en la Tabla 2.3.

Categoría estructural	Definición	Conteo
Revisión de Literatura	Estudios que presentan revisión de literatura.	2
Modelos propuestos	Estudios que formulan y describen un modelo.	1
Framework	Estudios que formulan y describen un Framework.	1
Herramientas	Estudios que formulan y describen herramientas	2
Herramientas + Metodologías ágiles	Estudios que formulan y describen herramientas y su relación con las metodologías ágiles	1
Estudios exploratorios y de caracterización.	Estudios que formulan y describen un enfoque exploratorio y de caracterización.	1
Marco de trabajo + Herramientas + Framework	Estudios que formulan y describen un marco de trabajo, herramientas y un framework.	1

**Tabla 2.3.** Conteo de estudios primarios por categorías estructurales

En la Tabla 2.4 se presenta una clasificación de los estudios primarios por año de publicación. Se observa que hay pocos estudios entre el 2008 y el 2017, dado que no se tiene suficiente información sobre procedimientos o marcos de trabajo enfocados a la automatización de pruebas unitarias y no se tiene un seguimiento de como han evolucionado estos estudios en la industria del software.

Año de publicación	Cantidad
2008	1
2009	1
2010	1
2011	1
2012	1
2013	1
2014	1
2015	2

**Tabla 2.4.** Conteo de estudios primarios por año de publicación.

## 2.4. Trabajos obtenidos mediante revisión sistemática

En esta sección se presenta el estado del arte elaborado a partir del análisis de los estudios obtenidos mediante la revisión sistemática de la literatura. El contenido está organizado de acuerdo con las categorías temáticas indicadas en la Tabla 2.3.

### 2.4.1. Revisiones de la literatura

Xiaochum et al. (2008) expresan que algunos sectores de la industria la automatización de pruebas sigue siendo poco rentables dado que es costoso pasar por todo el procedimiento de desarrollo, diseño y ejecución de scripts de prueba. En cuanto al mantenimiento generalmente es muy costoso cuando llegan grandes cambios si se habla de una baja reutilización e inflexibilidad de los guiones por lo que conduce a un gran costo. Hablan de una barrera que se establece a la hora de involucrar a los probadores que no tienen habilidades de programación relacionadas y por ello requieren que los miembros conozcan los requisitos de las pruebas además de las habilidades de programación. También se menciona una solución de automatización de pruebas en la prueba funcional de interfaces de usuario la cual integra la generación, selección, ejecución e informes de los casos de prueba para facilitar el proceso mismo. En este método se presenta un controlador que está diseñado para interpretar casos de prueba en objetos de clase y encaminarlos a un motor de prueba que luego los ejecutara automáticamente.

Böhme y Paul (2015) explican que los desarrolladores buscan técnicas de pruebas automatizadas para ganar una mayor confianza en la corrección de sus programas, las pruebas automatizadas de software son una manera más fácil y eficiente de inspirar confianza en la corrección del programa. En este estudio se centran en el análisis de la eficiencia en las pruebas de software, considerando dos estrategias, pruebas aleatorias y pruebas sistemáticas. Concluyen que las investigaciones relacionadas con ingeniería de software dedican mucho tiempo y esfuerzo para desarrollar técnicas de prueba altamente eficaces, de hecho, las usan para probar la corrección de un programa, sin embargo, en la práctica se desarrollan programas muy grandes con tiempo limitado para las pruebas.

#### **2.4.2. Modelos propuestos**

Joy y Pratap (2015) determinan que el aumento en la creación de sistemas software en diferentes dominios exige una mayor necesidad de garantía de calidad. Un objetivo de las pruebas automatizadas de software es optimizar la velocidad, la eficiencia y la calidad de la aplicación junto con la reducción de costos involucrados en el proceso manual, el automatizar las pruebas de software es una de esas estrategias que permite a las organizaciones alcanzar una mayor fiabilidad y eficiencia de sus productos. En este artículo se ilustra el diseño e implementación, de un marco genérico que se puede utilizar para mejorar las capacidades de un marco de pruebas de automatización existente. Este marco aborda tres fases como son:

- Fase de pre – ejecución, que hace referencia al banco de datos de entrada requerido para ejecutar los scripts de prueba.
- Fase de ejecución y validación, se refiere a la ejecución de prueba real de todos los casos de prueba que están en el alcance como parte de cualquier liberación o el ciclo de vida de prueba de regresión, para cada caso de prueba se deben asegurar las validaciones apropiadas para determinar si el caso de prueba pasó o falló.
- Fase post- ejecución, hace referencia a la fase de generación y análisis de resultados de los casos de prueba ejecutados. Este marco genérico se utiliza para soportar datos de prueba de análisis, de archivos, que se utilizan en la fase de ejecución de prueba.

### **2.4.3. Framework**

Sávio y Duarte (2013) buscan que las pruebas de software cada vez se realicen con mayor rapidez y de forma automática, una buena exploración de estas serían las plataformas de computación en la nube para ejecutar las pruebas de software, ya que pueden proporcionar ganancias significativas en la eficiencia y la eficacia de las pruebas en comparación con los métodos tradicionales. Por lo general para desplegar el desarrollo y procesos de prueba en la nube se requiere de esfuerzos significativos en la configuración, distribución y ejecución de las pruebas. En este artículo proponen un marco llamado CloudTesting que permiten la ejecución paralela de las pruebas automatizadas de software en entornos heterogéneos disminuyendo el tiempo de prueba a través de una capa de abstracción para los usuarios, eliminando la necesidad de realizar configuraciones complejas. Concluyen que el Framework CloudTesting simplifica la ejecución de pruebas automáticas en entornos distribuidos.

### **2.4.4. Herramientas**

Moradbeiky et al. (2014) Indican que actualmente el número de herramientas para pruebas de software es muy amplio y las herramientas de pruebas son de ayuda para hacer seguimiento y control a los proyectos de software. El artículo presenta y considera diferentes herramientas de prueba como: MUnit, WCAT, Microsoft Project Test, etc.; que en primer lugar consideran la aplicabilidad y las características de herramientas como:

- MUnit: Se basa en las pruebas unitarias, además de que se soporta bajo el lenguaje .Net, pero soportada en aplicaciones de escritorio. Tiene independencia operacional y es de código abierto, además de usar un informe de variedades tipo texto.
- WCAT: Se basa en pruebas de carga y prueba de esfuerzo, además de que se basa en web y soportada en aplicaciones web. Tiene independencia operacional pero no es de código abierto, además de usar un informe de variedades tipo texto.
- Microsoft Project Test: Se basa en las pruebas unitarias, además de que se soporta bajo el lenguaje .Net, pero soportada en aplicaciones de escritorio. Tiene independencia operacional y no es de código abierto, además de usar un informe de variedades tipo texto, estadístico y gráfico.

Posteriormente, tratan de elegir un amplio rango de características incluyendo varios tipos de pruebas donde buscan herramientas de prueba de alto rendimiento incluyendo

los entornos de pruebas de software populares para extender los resultados a diversos grupos de desarrolladores de software, así se concluye que al combinar varias herramientas es un gran privilegio dado que no se está limitado a las características de herramientas específicas y así utilizar las ventajas de otras herramientas.

Safronaun y Turlo (2011) describen que muchas empresas de la industria del software pierden tiempo y recursos en la comercialización de sus productos, una implementación correcta de la automatización de pruebas puede ser una manera eficiente de garantizar un tiempo de comercialización más rápido para los nuevos productos. En ese estudio se habla de los desafíos que tienen las empresas al tratar de dar una solución de este tipo, y proporcionan recomendaciones prácticas sobre la implementación, también habla de la realización de la ejecución de pruebas automatizadas, desde la configuración del entorno y el lanzamiento de las pruebas, hasta el seguimiento de defectos y la generación de informes, basado en la integración de herramientas de prueba. La solución se basa en la integración de herramientas de prueba. Se han aplicado en la práctica y se ha demostrado su utilidad en las pruebas automatizadas.

#### **2.4.5. Herramientas + Metodologías Ágiles**

Figueiredo y Ferreira (2012) explican que un factor esencial en las pruebas de software es el tiempo que toma realizarlas, esta y otras razones en el ámbito de las pruebas ayuda a observar la importancia de que las pruebas deben ser automatizadas. Las pruebas automatizadas aumentan la eficiencia de los pasos repetitivos además de que generalmente se automatizan pruebas de aceptación en los casos de pruebas Selenium GUI<sup>1</sup> estables obteniendo un 85% de cobertura, mientras que sobre los casos de pruebas Selenium GUI inestables que no agregan valor al proyecto se ven reducidas en un 50% sobre la cobertura de automatización, también se considera que la automatización de pruebas es una actividad esencial para las metodologías ágiles sean la clave de aceleración del proceso de garantía de calidad. Este artículo presenta observaciones empíricas y los desafíos de un equipo de pruebas nuevo para prácticas ágiles y Automatización de Pruebas utilizando herramientas de prueba de código abierto integradas en proyectos de software que utilizan la metodología Scrum. Concluyen que es factible adaptar las prácticas de metodologías ágiles con pruebas

---

<sup>1</sup> Selenium GUI: Herramienta para automatizar aplicaciones web con fines de prueba.

automatizadas de software, buscando herramientas integradas de prueba de código abierto mientras se mantiene la gestión y la organización de una prueba de software.

#### **2.4.6. Estudios exploratorios y de caracterización**

Candea et al. (2010) proponen trabajar un caso de software automatizado TaaS como un servicio basado en la nube, se combina dos ideas de TaaS como son: Ofrecer pruebas de software como un servicio web y realizar pruebas en la nube y así aprovechar esos recursos elásticos, haciendo las pruebas automatizadas prácticas para el software real. Trabajan cuatro retos de investigación: (a) pruebas completamente automatizadas, (b) escala a base de código que cambia con frecuencia, (c) ser factible como un servicio útil para desarrolladores / consumidores y económicamente viable, (d) ser capaz de probar directamente los archivos binarios. Concluyen que la combinación de avances en la automatización de pruebas y la disponibilidad de servicios en la nube pueden ofrecer mejoras al proceso de pruebas.

#### **2.4.7. Marco de Trabajo + Herramienta Framework**

Ha, Chae y Moon (2009) intuyen que se evidencian cambios en la industria de software, cada vez son mayores las demandas de negocio en cuanto a la creación de productos software más grandes y complejos en menos tiempo, además uno de los papeles más importantes para la industria es la mejora de las pruebas de software en las cuales se puede apreciar que las herramientas de automatización de estas pruebas permiten a los desarrolladores y/o probadores automatizar el proceso. Para los desarrolladores y/o probadores no es fácil esta práctica debido a la falta de información, en este artículo se propone un enfoque de infraestructura y automatización junto con un conjunto de mejores prácticas que explican en detalle la implementación y la arquitectura de Framework de automatización de pruebas que combina varias estructuras de control, tomando como referencia el marco de trabajo NTAF<sup>2</sup> el cual es construido sobre una herramienta llamada FitNesse<sup>3</sup>, además de ser una mezcla de diversas herramientas, su flujo de trabajo se ve expresado gráficamente como datos de entrada y de salida esperados en entornos distribuidos. Un ejemplo de ello es al recoger nuevas versiones automáticamente, instalarlas en máquinas remotas

---

<sup>2</sup> NTAF: Marco de trabajo construido sobre FitNesse.

<sup>3</sup> FitNesse: Servidor web, wiki y herramienta automatizada para software.

y ejecutar pruebas en forma remota, esto puede lograrse enviando comandos a través de STAF<sup>4</sup> a las aplicaciones y reportando los resultados en FitNesse.

## **2.5. Síntesis**

En este capítulo se ha presentado una revisión sistemática de la literatura para determinar la disponibilidad de trabajos relacionados con la automatización de pruebas para pequeñas empresas desarrolladoras de software, tales como marcos conceptuales, marcos de trabajo, procedimientos o guías técnicas que indiquen cuáles prácticas o actividades de automatización de pruebas se deben efectuar para esta clase de organizaciones y también se muestre como debe ser su ejecución.

A partir de la revisión sistemática que se ha aplicado y de los trabajos identificados, en definitiva, se observa que no está disponible una propuesta de procedimiento que relacione prácticas, actividades y guías técnicas que puedan ayudar a las pequeñas empresas a planear, ejecutar y documentar pruebas automatizadas de software, esto se debe a que ninguno de los artículos ofrece una propuesta clara de cómo se deben realizar tales pruebas. En este sentido, se requiere investigar en este tema con el fin de aportar elementos que permitan una mejor comprensión y aplicación de la automatización de pruebas para pequeñas empresas desarrolladoras de software.

En esta revisión, se encontró que la mayoría de los estudios primarios involucrados están orientados a la definición y descripción de herramientas y/o Frameworks de automatización, y son pocos los estudios que involucran brevemente modelos propuestos sobre automatización de pruebas, además de especificarlo de manera muy simple.

Con la revisión sistemática aplicada y los trabajos relacionados se observa que no hay disponibilidad de un marco de trabajo, procedimiento o guía que relacione las actividades, roles, entradas, salidas e indicadores básicos que ayuden en la orientación para la automatización de pruebas unitarias en pequeñas empresas desarrolladoras de software, esto se debe a que en algunos estudios presentan herramientas utilizadas en áreas específicas combinadas con marcos de trabajo que

---

<sup>4</sup> STAF: Comandos de alerta para usuarios.

están relacionados con herramientas específicas y hablan de pre-ejecución, ejecución y pos-ejecución, en otros se hacen comparaciones entre ellas. En este sentido se requiere investigar en el tema con la finalidad de aportar un procedimiento que permita saber cuáles actividades o practicas se deben tener en cuenta a la hora de automatizar pruebas unitarias software.

## **Capítulo 3**

# **Caracterización de prácticas de automatización de pruebas unitarias en pequeñas empresas desarrolladoras de software**

### **3.1. Elaboración del instrumento de recolección de información**

Dadas las últimas tendencias de la tecnología en el desarrollo software en las cuales la mayoría se encaminan hacia las metodologías ágiles, es necesario abordar una de las partes más importantes del desarrollo software como son las pruebas. En este estudio se consultó sobre automatización de pruebas, definida por Bach J. (1999) como una técnica que facilita labores repetitivas con la finalidad de liberar a los testers de actividades tediosas para concentrarse en la funcionalidad del negocio que es lo esperado por el usuario, no se debe olvidar que todas las pruebas automatizadas requieren de la intervención humana para diagnosticar resultados y corregir pruebas erróneas. Es por ello que en este estudio se busca recolectar información sobre las prácticas de automatización de pruebas que puedan tener algunas empresas de la ciudad de Popayán con el fin de determinar cuáles elementos hay que tener en cuenta para incluirlos en el procedimiento objeto de este estudio.

Se elaboró una encuesta exploratoria en línea sobre automatización de pruebas unitarias de software conformada por 22 preguntas utilizando la herramienta para formularios de Google Docs, para reunir información sobre las prácticas que se llevan a cabo en pequeñas empresas desarrolladoras de software en la ciudad de Popayán, la encuesta está conformada por 2 secciones.

**Sección 1. Información básica:** Se busca reunir información de la organización para ello se solicita el nombre de quien diligencia la encuesta, la empresa a la cual representa y el cargo que ocupa en la empresa.

**Sección 2. Preguntas de selección:** Se busca obtener información que dé a conocer las prácticas, procesos, presupuestos, metodologías y herramientas de pruebas que enfoquen la automatización de pruebas unitarias de software.

## **3.2. Recolección de información**

Se obtuvo respuesta de seis (6) empresas de la ciudad de Popayán:

- CeoTic
- Namtrik Dev S.A.S.
- Eteknik
- Ideti
- Somos
- TalOSystems

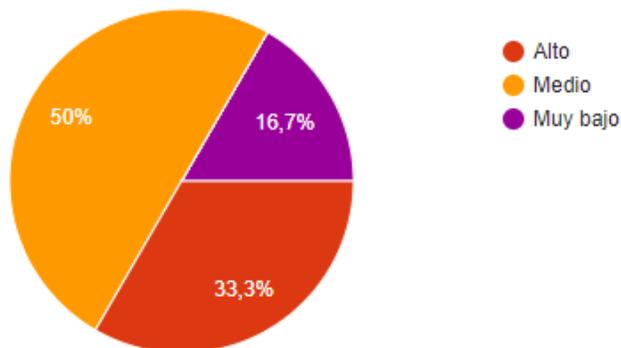
## **3.3. Resultados obtenidos**

### **3.3.1. Sección 1. Información básica**

En esta sección se preguntó el nombre y cargo de la persona que diligenció la encuesta y el nombre de la empresa a la que representa

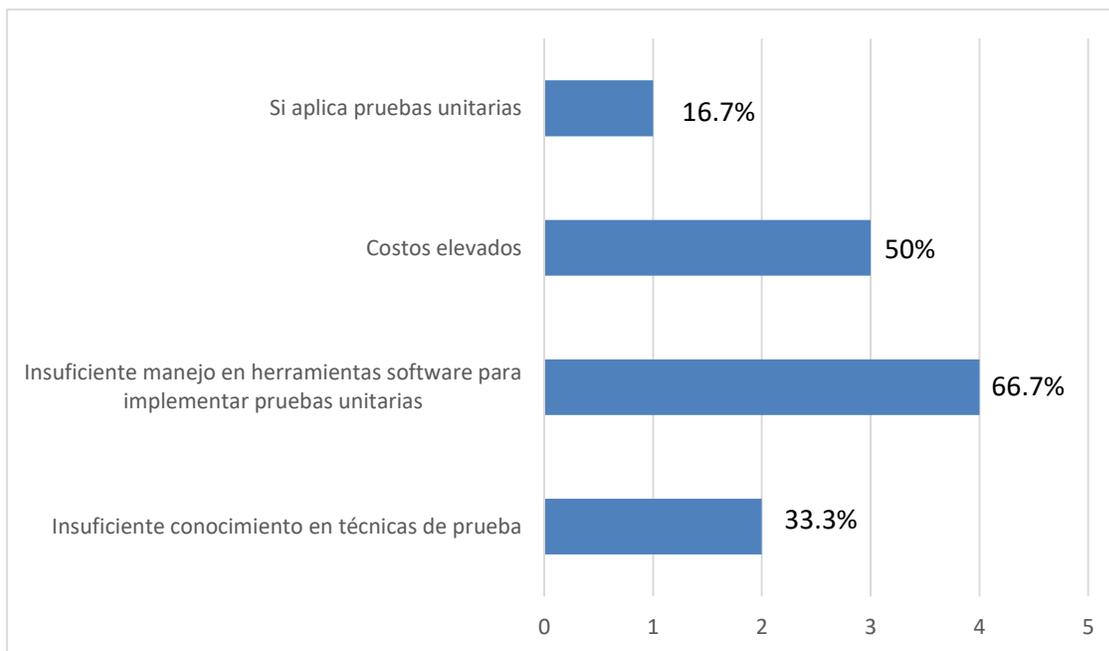
### 3.3.2. Sección 2. Preguntas de selección

**Pregunta 1.** Considera que el nivel de eficacia de las pruebas unitarias en cuanto a la cantidad de errores encontrados es:



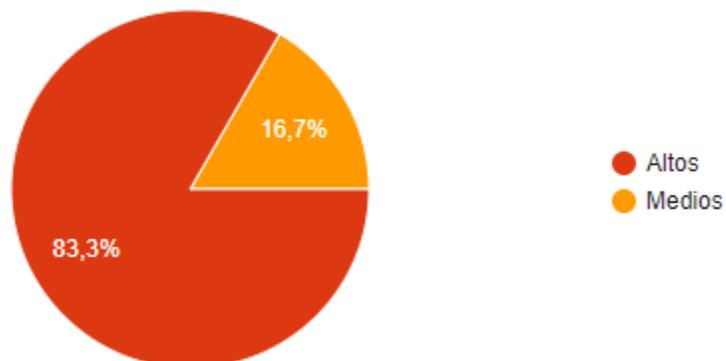
**Figura 3.1.** Nivel de eficacia de las pruebas unitarias en cuanto a la cantidad de errores encontrados.

**Pregunta 2.** ¿Cuál es el mayor factor de incidencia para NO ejecutar pruebas unitarias de software?



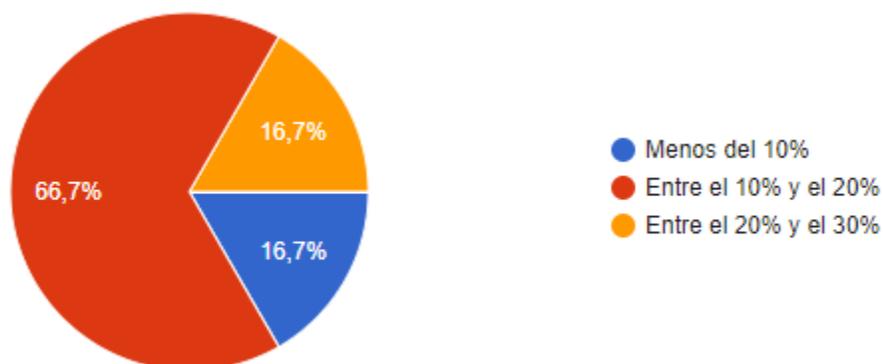
**Figura 3.2.** Mayor factor de incidencia para NO ejecutar pruebas unitarias de software.

**Pregunta 3.** Considera que los costos respecto al presupuesto del proyecto en el descubrimiento y depuración de errores son:



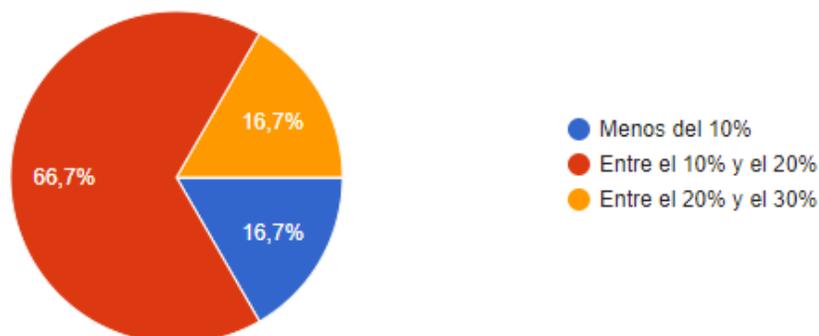
**Figura 3.3.** Nivel de costos respecto al presupuesto del proyecto en el descubrimiento y depuración de errores.

**Pregunta 4.** ¿Qué porcentaje del presupuesto de un proyecto se ha destinado para pruebas y depuración?



**Figura 3.4.** Porcentaje del presupuesto de un proyecto destinado para pruebas y depuración.

**Pregunta 5.** Qué porcentaje del tiempo asignado en el cronograma se emplea para hacer pruebas:

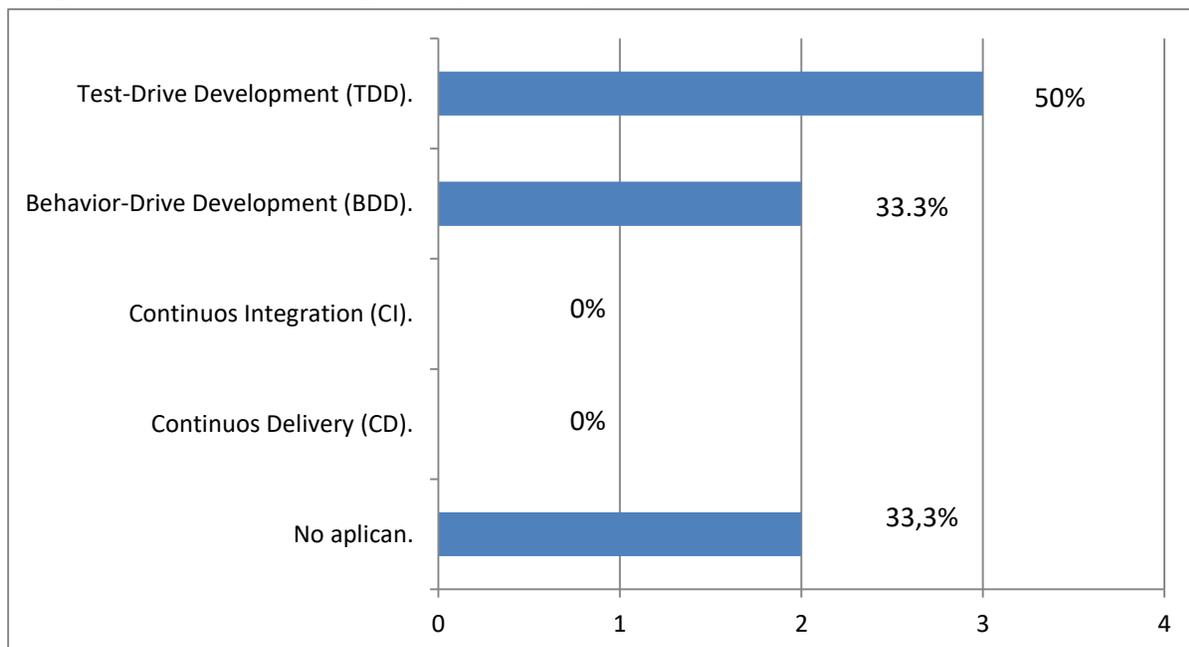


**Figura 3.5.** Porcentaje del tiempo asignado en el cronograma que se emplea para hacer pruebas.

**Pregunta 6.** ¿Cuál ha sido el costo promedio de la adquisición de herramientas para automatización de pruebas?

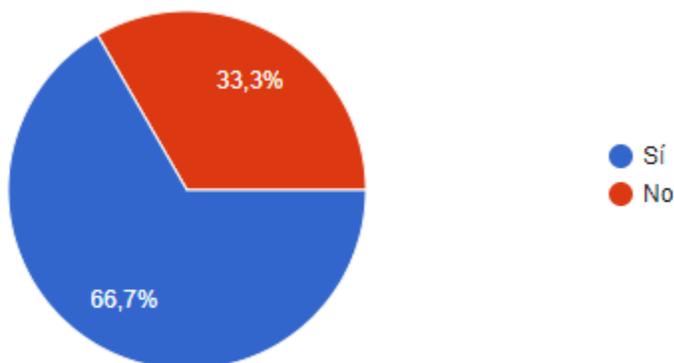
Todas las empresas tienen un costo promedio de adquisición de herramientas para automatización de pruebas de menos de \$10.000.000 de pesos.

**Pregunta 7.** La metodología de pruebas que usan es basada en:



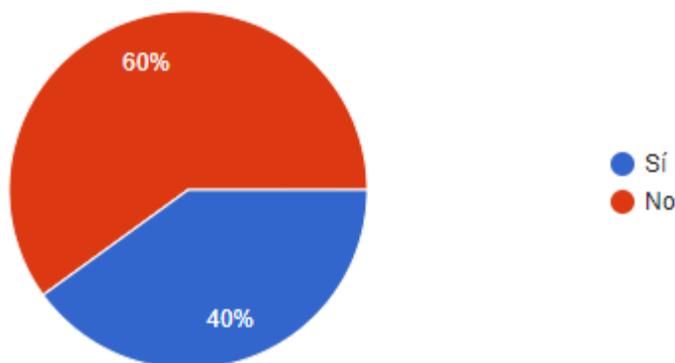
**Figura 3.6.** Metodología de pruebas usada por las empresas.

**Pregunta 8.** ¿Los desarrolladores hacen pruebas unitarias de sus propios códigos?  
Esta pregunta tiene tres tipos de respuesta en la cual de ser afirmativa pasa a la pregunta 9, en caso contrario pasa a la pregunta 10.



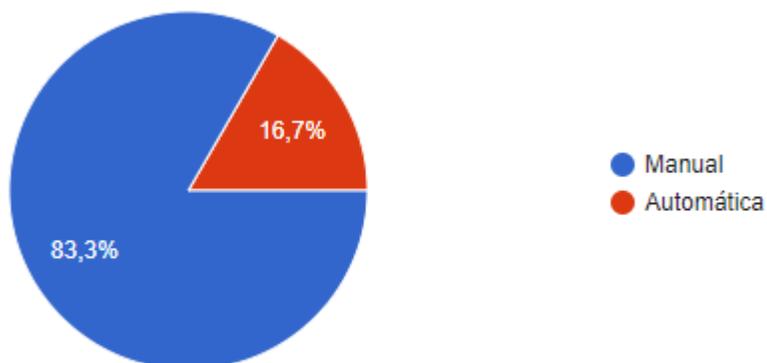
**Figura 3.7.** Porcentaje de desarrolladores que hacen pruebas unitarias de sus propios códigos.

**Pregunta 9.** ¿Usan técnicas de programación por pares o pruebas de rotación de código?  
Esta pregunta tiene tres tipos de respuesta en la cual de ser afirmativa pasa a la pregunta 10, en caso contrario pasa a la pregunta 11.



**Figura 3.8.** Porcentaje de desarrolladores que usan técnicas de programación por pares o pruebas de rotación de código.

**Pregunta 10.** La evaluación es:



**Figura 3.9.** Tipo de evaluación que hacen los desarrolladores.

**Pregunta 11.** En cuanto al manejo de pruebas...

Esta pregunta tiene cuatro tipos de respuesta en la cual si la empresa subcontrata pasa a la pregunta 12 en caso contrario pasa a la pregunta 13.



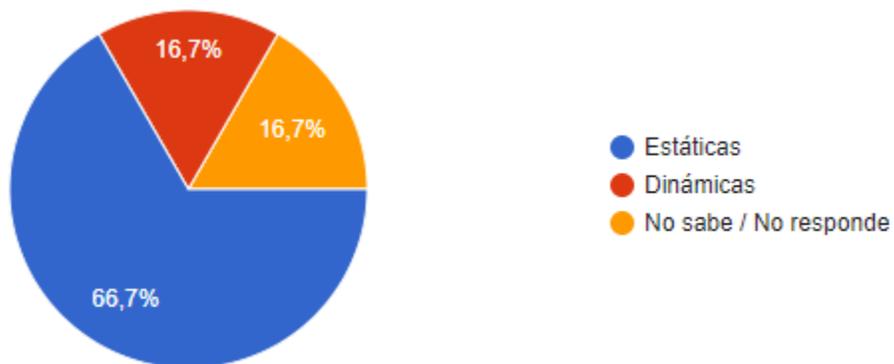
**Figura 3.10.** Porcentaje que especifica el manejo de pruebas en la empresa.

**Pregunta 12.** ¿Cuál es el rango de costos en la subcontratación respecto al presupuesto del proyecto?

El rango de costos en la subcontratación respecto al presupuesto del proyecto en todas las empresas es de menos del 5%.

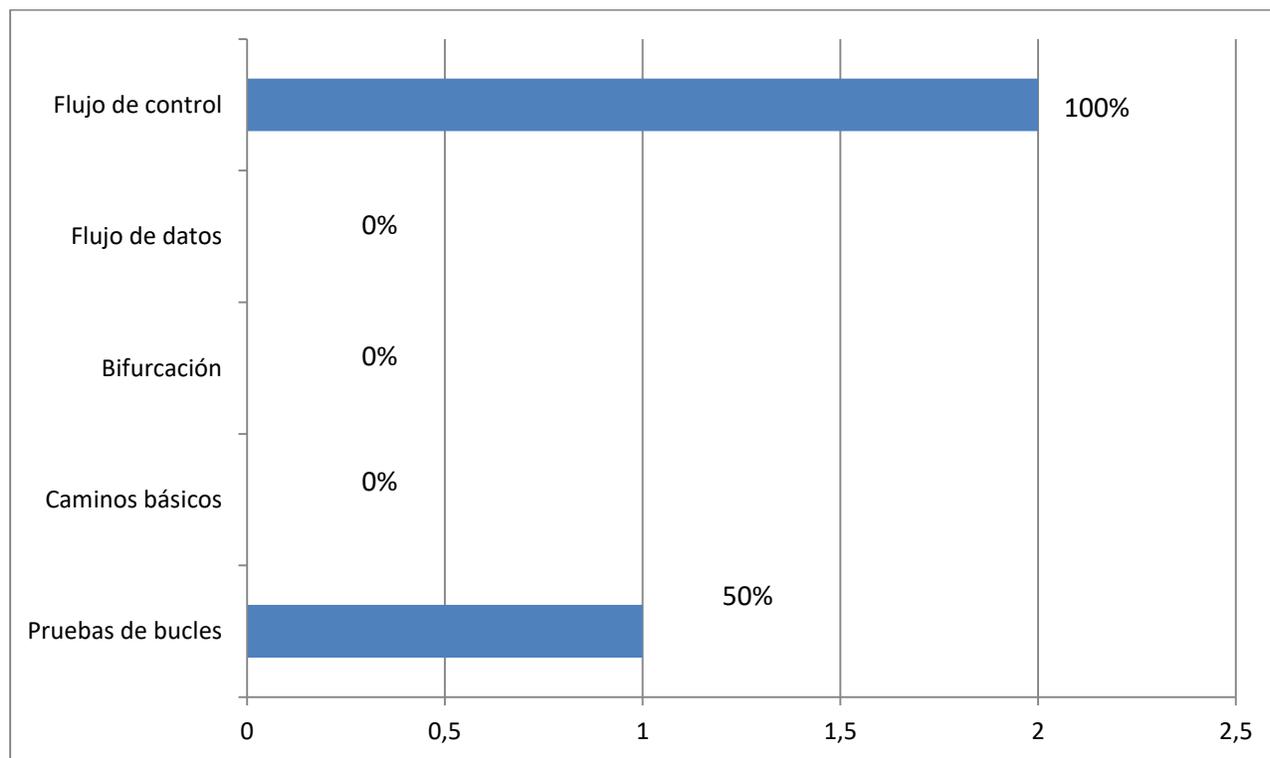
**Pregunta 13.** Las pruebas unitarias que realizan son:

Esta pregunta tiene tres tipos de respuesta en la cual de ser dinámica pasa a la pregunta 14, en caso contrario pasa a la pregunta 17.



**Figura 3.11.** Pruebas unitarias que realizan las empresas.

**Pregunta 14.** Del enfoque de caja blanca que técnicas de pruebas realizan:



**Figura 3.12.** Técnicas de prueba usadas según el enfoque de caja blanca.

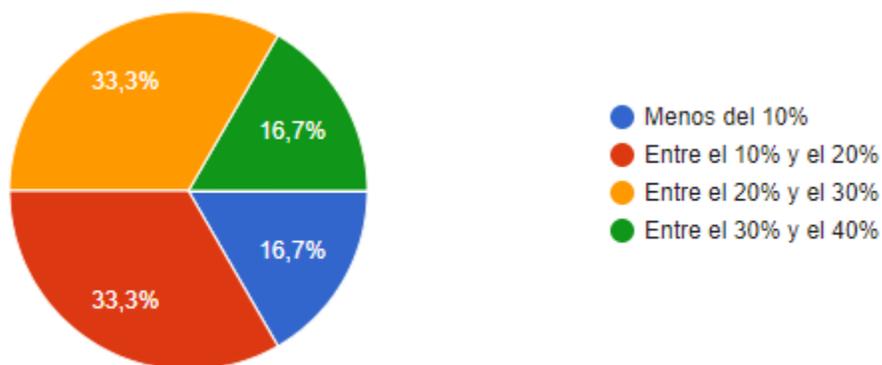
**Pregunta 15.** La (s) herramienta (s) usadas para el enfoque de caja blanca son (es):

Según el estudio realizado el 100% de las empresas usan una herramienta propia de la empresa para el enfoque de caja blanca.

**Pregunta 16.** Del enfoque de caja negra que técnicas de pruebas realizan:

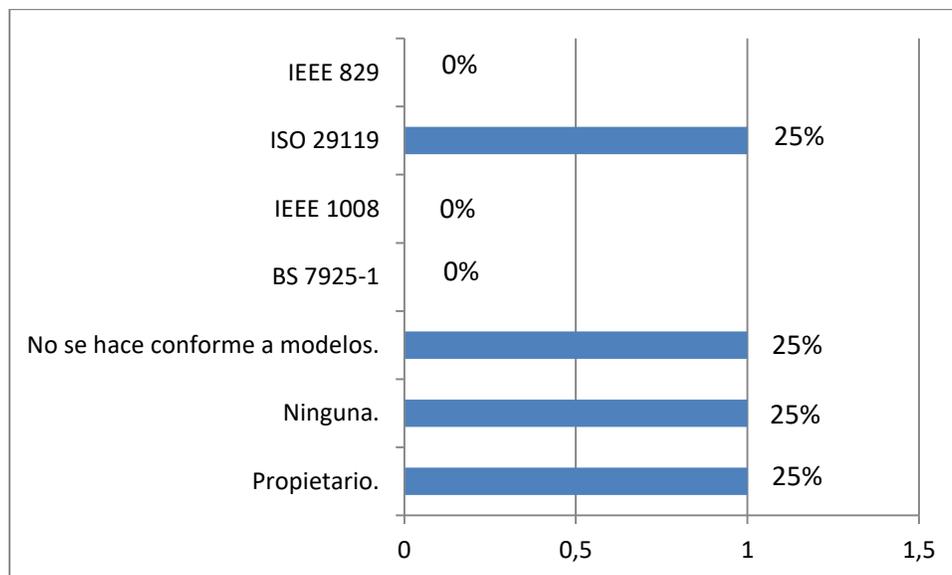
Según el estudio realizado el 100% de las empresas usan la técnica de casos de uso para el enfoque de caja negra.

**Pregunta 17.** ¿En qué porcentaje cree usted que se incrementarían los costos totales en el proyecto a la hora de automatizar pruebas?



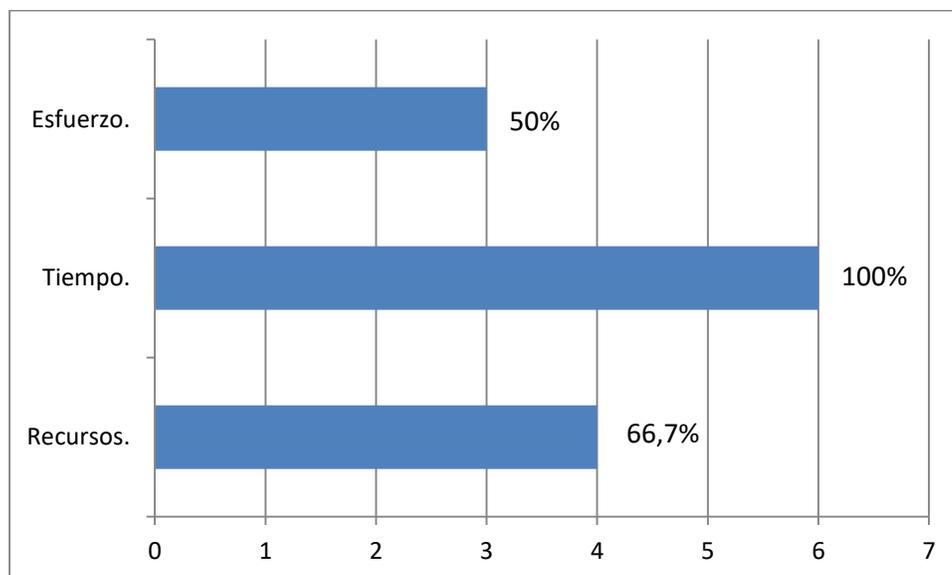
**Figura 3.13.** Porcentaje en el que se incrementan los costos totales en el proyecto a la hora de automatizar.

**Pregunta 18.** ¿Cuáles de estas normas, modelos o propuestas utilizan ustedes para documentar las pruebas?



**Figura 3.14.** Normas, modelos o propuestas utilizadas para documentar pruebas.

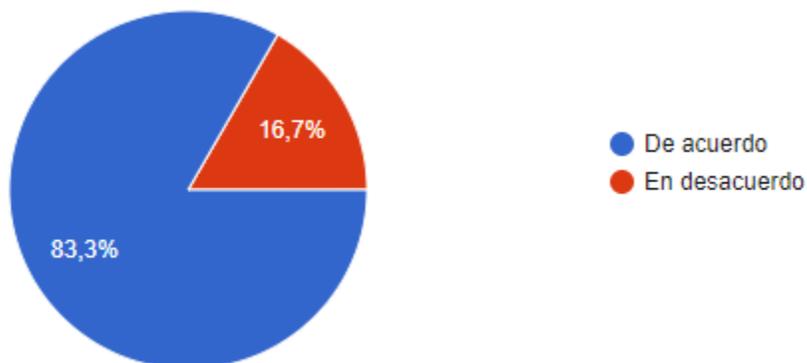
**Pregunta 19.** ¿Qué aspectos son los más importantes a la hora de definir la estrategia de pruebas?



**Figura 3.15.** Aspectos más importantes a la hora de definir la estrategia de pruebas.

**Pregunta 20.** De las siguientes afirmaciones está usted de acuerdo o en desacuerdo.

**20.1.** El coste de eliminar un defecto se incrementa con el tiempo de permanencia de dicho defecto.



**Figura 3.16.** Porcentaje referente al coste de eliminar un defecto se incrementa con el tiempo de permanencia del defecto.

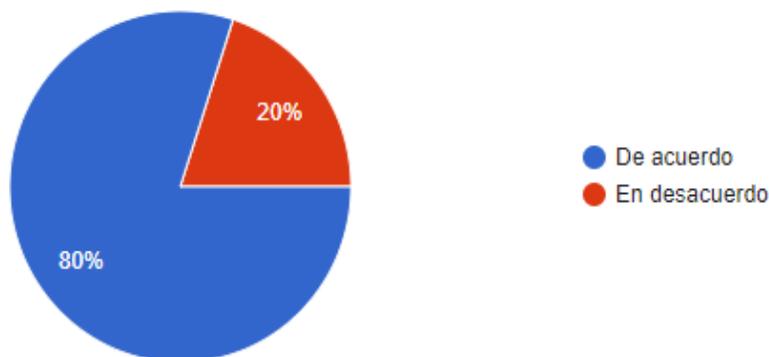
**20.2.** La detección de errores en etapas tempranas permite su corrección a menor coste.

Todas las empresas estuvieron de acuerdo en esta pregunta.

**20.3.** El usar herramientas automatizadas harían que gasten menos tiempo.

Todas las empresas estuvieron de acuerdo en esta pregunta.

**20.4.** Los costos de un proyecto en cuanto al presupuesto disminuyen usando herramientas automatizadas.



**Figura 3.17.** Porcentaje referente a los costos de un proyecto en cuanto al presupuesto se disminuye usando herramientas automatizadas.

**Pregunta 21.** Del 1 al 5 califique qué aspectos debe contemplar una prueba bien definida. Donde 1 es la menor calificación y 5 es la mejor calificación.

EMPRESAS	ASPECTOS EVALUADOS					
	Identificador	Valores de Entrada	Resultados esperados	Precondiciones	Dependencias	Acciones a requisitos vinculados
Eteknik S.A.S	5	5	4	4	3	4
CeoTic	4	4	5	4	4	5
IDETI	4	5	5	-	3	3
Namtrik Dev S.A.S	5	5	5	5	5	3
TaIO Systems	5	5	5	4	4	5
SoMoS	-	-	5	4	4	3

**Tabla 3.1.** Tabla referente a las respuestas de las empresas según los aspectos evaluados.

Con los aspectos se obtuvieron los siguientes resultados en donde se tienen los aspectos, la calificación que se le dio y el número de empresas que respondieron a cada calificación:

ASPECTOS EVALUADOS	CALIFICACIÓN	NÚMERO DE EMPRESAS
Identificador	5	3
	4	2
	Sin respuesta	1
Valores de entrada	5	4
	4	1
	Sin respuesta	1
Resultados esperados	5	5
	4	1
Precondiciones	5	1
	4	4
	Sin respuesta	1
Dependencias	5	1
	4	3
	3	2
Acciones a requisitos vinculados	5	2
	4	1
	3	3

**Tabla 3.2.** Conglomerado de los aspectos evaluados según la calificación dada por las empresas.

**Pregunta 22.** En una escala de 1 a 5 qué beneficios son los más importantes para usted a la hora de realizar pruebas automatizadas.

EMPRESAS	ASPECTOS EVALUADOS			
	Fiabilidad	Repetición	Programable	Reusabilidad
Eteknik S.A.S	5	2	4	5
CeoTic	4	4	4	5
IDETI	4	3	4	5
Namtrik Dev S.A.S	5	5	5	5
TaIO Systems	5	5	5	5
SoMoS	4	-	-	-

**Tabla 3.3.** Tabla referente a la calificación de las empresas según los aspectos evaluados.

Con los aspectos evaluados se obtuvieron los siguientes resultados en donde se tienen los aspectos, la calificación que se le dio a cada aspecto y el número de empresas que respondieron a cada calificación:

ASPECTOS EVALUADOS	CALIFICACIÓN	NÚMERO DE EMPRESAS
Fiabilidad	5	3
	4	3
Repetición	5	2
	4	1
	3	1
	2	1
	Sin respuesta	1
Programable	5	2
	4	3
	Sin respuesta	1
Reusabilidad	5	5
	Sin respuesta	1

**Tabla 3.4.** Conglomerado de los aspectos evaluados según la calificación dada por las empresas.

## **3.4. Síntesis**

### **Características**

Con la encuesta realizada se identificó que uno de los factores de incidencia para no ejecutar pruebas unitarias de software es que el 66.7% de las empresas tienen un insuficiente manejo en herramientas software para implementar pruebas unitarias, además otro aspecto importante que se descubrió fue que las empresas no ejecutan pruebas unitarias de software dado que los costos son elevados a la hora de implementar pruebas, además las empresas creen que los costos en cuanto al presupuesto de un proyecto es alto dado que éstas destinan entre el 10% y el 20% del presupuesto para pruebas y depuración y en cuanto al tiempo que le asignan a las pruebas según el cronograma del proyecto es del 10% y el 20%.

La mayoría de empresas realizan pruebas unitarias estáticas y en cuanto al enfoque de caja blanca solo dos empresas usan el flujo de control y tan solo una la prueba de bucles, pero a la hora de usar herramientas para el enfoque de caja blanca sólo una usa un software propio de la empresa, por otro lado, en cuanto al enfoque de caja negra resalta que las pruebas de caso de uso es la más usada por las empresas.

### **Necesidades**

Se evidencia que la metodología Test-Driven-Development (TDD) es la más utilizada por las empresas, es por ello que en la mayoría los desarrolladores hacen pruebas unitarias de sus propios códigos, pero no usan la técnica de programación por pares o pruebas de rotación de código. Su método de evaluación de código es de forma manual, pero si en algún momento las empresas se ven forzadas a subcontratar, estas destinan menos del 5% del presupuesto del proyecto para ello.

En cuanto a los costos de automatizar pruebas, no hay consenso al respecto, puesto que algunas empresas indican que el costo total de un proyecto aumentaría desde un 10% y otras mencionan un aumento de 30% hasta incluso un 40%.

En cuanto a las normas, modelos o propuestas las empresas encuestadas no tienen un consenso que oriente a la documentación de los procesos de prueba además se concluye que a la hora de definir una estrategia de pruebas el tiempo es el más importante, los recursos y el esfuerzo pasan a un segundo plano dado que se evaluó en pequeñas empresas desarrolladoras de software con recursos y mano de obra limitados

## **Capítulo 4**

### **Descripción del procedimiento**

En la creación del procedimiento se siguieron los lineamientos planteados por CMMI-DEV 1.3 (2010), en donde se construyeron los procedimientos y tareas claves según la información recolectada en este estudio, posteriormente se identificaron las posibles condicionales para las tareas, además de incluir las plantillas a las tareas que lo requerían, se incluyeron también los roles para cada tarea y una breve descripción de cada una de las tareas incluidas en el procedimiento. Posteriormente se refinó la propuesta con la evaluación de expertos y el estudio de caso realizada a la empresa Namtrik Developmente S.A.S.

En este capítulo se describen los procesos clave que fueron seleccionados a partir de la encuesta de caracterización descrita y analizada en el Capítulo 3, además de la revisión de la literatura que se realizó. Las tareas clave corresponden a los procesos de identificación y selección, establecimiento de criterios, procedimientos y entornos, preparación, realización y análisis de la revisión entre pares, realización de la verificación y validación y análisis de resultados. Se consideró para su estudio el modelo propuesto para pequeñas empresas desarrolladoras de software, definido por CMMI-DEV 1.3 (2010) del cual se toma la segunda parte como referencia dado que se describen dos procesos estructurados para el manejo de pruebas como son validación y verificación de los cuales se toman aspectos para incorporar en el procedimiento como los siguientes:

#### **Preparar la validación y verificación**

1. Seleccionar los productos a validar y verificar
2. Establecer el entorno de validación y verificación
3. Establecer los procedimientos y los criterios de validación y verificación

## Realizar las pruebas de verificación y validación

1. Realizar las pruebas entre pares

### Ciclo de Pruebas de Calidad de Software



Figura 4.1. Ciclos de prueba de calidad de software.

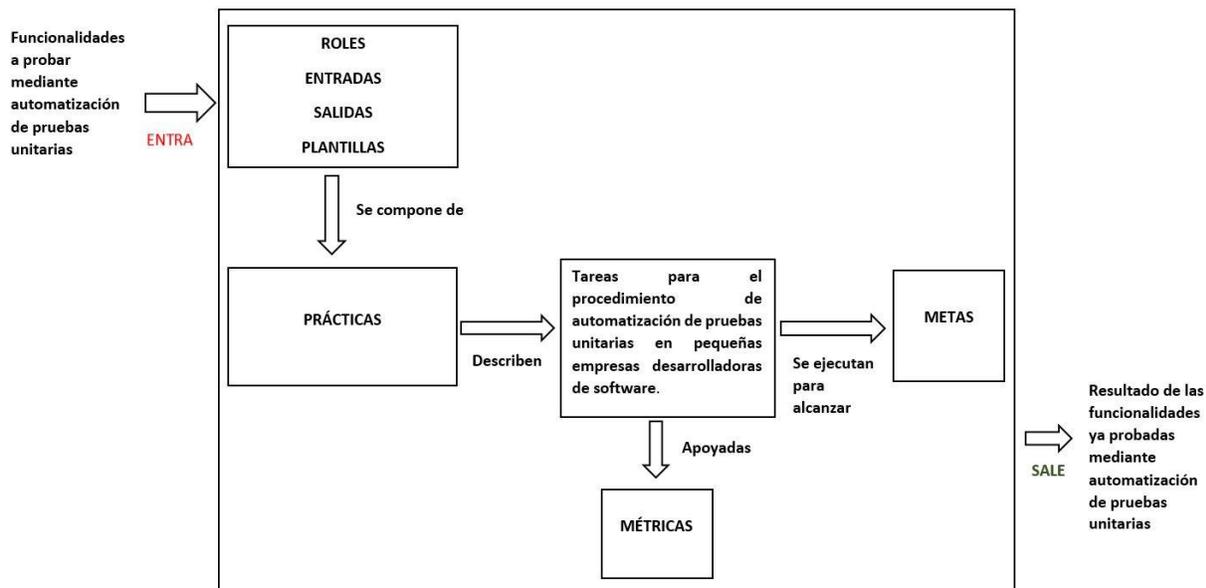
## 4.1. Descripción conceptual del procedimiento

### 4.1.1. Propósitos y objetivos

1. Establecer las prácticas y los elementos de manera detallada, ordenada, sistemática que contenga instrucciones de cómo llevar a cabo la automatización de pruebas unitarias enfocado en pequeñas empresas desarrolladoras de software.
2. Proporcionar los elementos y la guía para llevar a cabo las prácticas de automatización de pruebas software en pequeñas empresas desarrolladoras de software.

3. Fomentar las buenas prácticas de documentación de resultados para las pruebas realizadas teniendo en cuenta las necesidades del cliente y llevando un acompañamiento de la empresa desarrolladora.

#### 4.1.2. Elementos principales del procedimiento



**Figura 4.2.** Elementos principales del procedimiento

Las prácticas clave corresponden a las prácticas de identificación y selección, establecimiento de criterios, procedimientos y entornos, preparación, ejecución de la prueba, realización y análisis de la revisión entre pares, realización de la verificación y validación y el análisis de resultados, cada una de estas prácticas tiene unas tareas a alcanzar por medio de unas actividades definidas las cuales son ejecutadas por uno de los roles de Cliente, Equipo de trabajo, Desarrolladores, personal de pruebas y jefe de proyecto, el cuál corresponde a un cargo asignado únicamente para las funciones relacionadas con el rol. Las tareas están apoyadas en métricas, éstas nos brindan información en cuanto al tiempo estimado sin el procedimiento y con el procedimiento y otras métricas que también se tienen en cuenta a la hora de realizar el procedimiento.

Las tareas correspondientes a las prácticas clave fueron definidas según el estudio y el análisis realizado sobre la mejora de procesos para el desarrollo de mejores productos y servicios planteados por CMMI-DEV 1.3 (2010) en las áreas proceso de verificación y validación donde además se tomaron en cuenta los elementos que se

consideraron pertinentes y fueron sometidos a una revisión por expertos, cuyos aportes se reflejaron en mejoras y ajustes tanto al procedimiento como a las plantillas y las descripciones de las prácticas y tareas propuestas para el procedimiento. Posteriormente en un estudio de caso realizado en la empresa Namtrik Development S.A.S. donde se ejecutó el procedimiento el cual se describe en mayor profundidad en el Capítulo 5.

Las prácticas planteadas en el procedimiento fueron definidas según el estudio realizado teniendo en cuenta la interdependencia de las mismas, así como también los flujos de las condicionales del modelo, igualmente se definieron las plantillas necesarias para recolectar la información correspondiente al procedimiento, y que están soportadas en hojas de cálculo que comprenden diferente información de la empresa y del producto para someter a pruebas.

## 4.2. Descripción detallada del procedimiento

### Vista general del procedimiento en BPMN

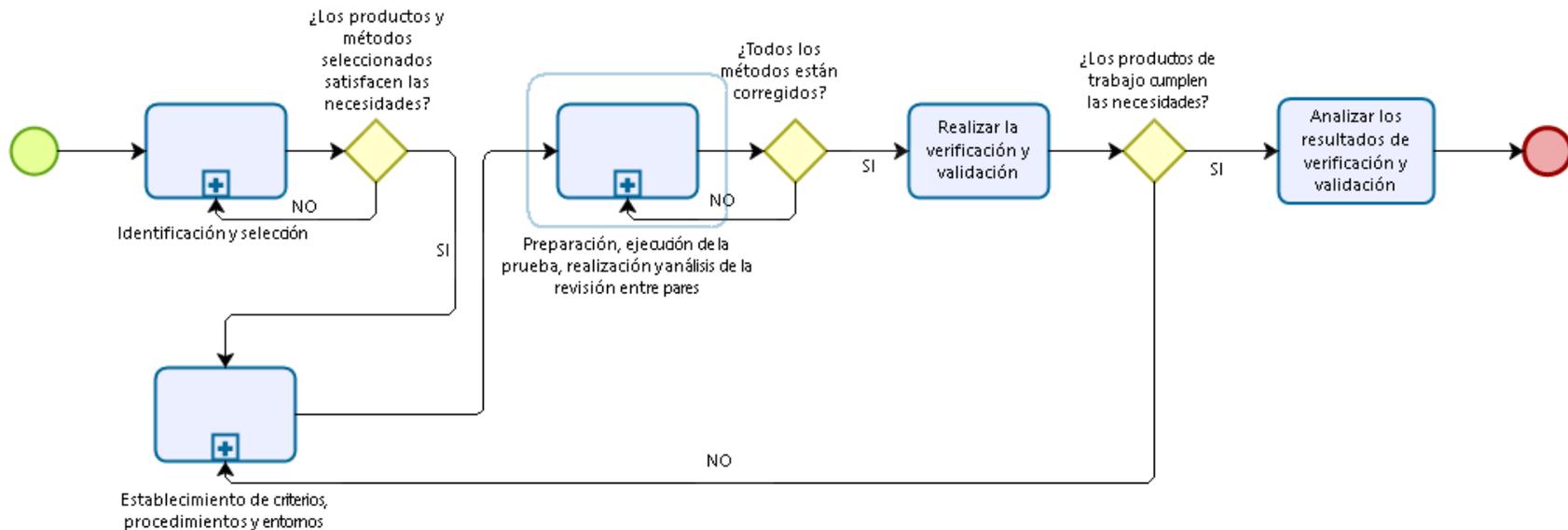


Figura 4.3. Vista general del procedimiento.

**Nota:** En este procedimiento se denominan “métodos” a los fragmentos de código a probar de manera unitaria.

## **Roles**

**Cliente:** Es el dueño del producto, es quien tiene claro lo que desea sobre el producto final.

**Equipo de trabajo:** El equipo de trabajo lo conforma: el jefe de proyecto, desarrolladores, personal de pruebas, diseñadores, entre otros. Son los encargados de desarrollar y probar la necesidad que tiene el cliente respecto a un producto software.

**Desarrolladores:** Son los encargados de codificar la solución software al problema del cliente.

**Personal de pruebas:** Son los encargados de verificar y validar el comportamiento del software, de ellos dependen los errores que tenga el software y su principal función no es la de erradicar al 100% los errores si no minimizarlos.

**Jefe de proyecto:** Es el encargado de liderar el equipo de trabajo facilitando los implementos o las necesidades que llegase a tener el equipo de trabajo, además es el puente entre el cliente y el equipo, también debe conocer perfectamente la metodología usada.

#### 4.2.1. Práctica 1. Identificación y selección

En esta práctica están comprendidas las tareas de Identificar los métodos a someter a pruebas, seleccionar los métodos a someter a pruebas y seleccionar los productos y métodos de verificación y validación, cada una de ellas se detallará con una breve descripción, sus entradas y salidas y la plantilla si la tarea lo requiere.

Vista de las tareas incluidas en la práctica:

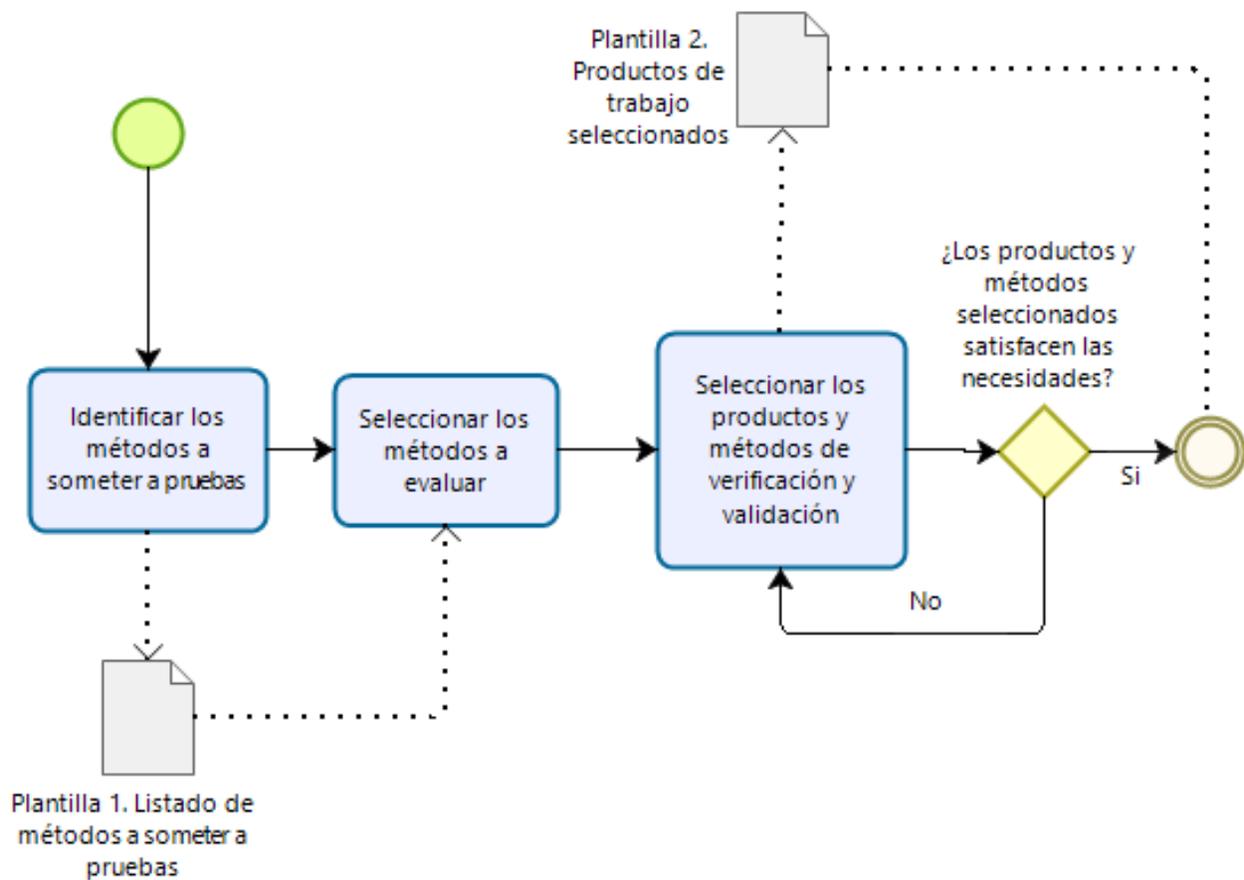


Figura 4.4. Tareas de la práctica identificación y selección.

#### 4.2.1.1. Tarea 1. Identificar los métodos que se van a someter a pruebas.

Título	<b>Identificar los métodos que se van a someter a pruebas.</b>
<b>DESCRIPCIÓN</b>	
En una reunión entre el equipo de trabajo y el personal de pruebas se especifican los métodos del proyecto software a ser probados teniendo en cuenta que se tiene el producto software ya desarrollado y se desea probar, el equipo de trabajo identificará los métodos aptos para ser sometidos a prueba y con la colaboración del personal de pruebas se verificará la viabilidad de la prueba correspondiente.	
<b>ROLES INVOLUCRADOS</b>	
Equipo de trabajo.	
<b>ENTRADAS</b>	
Métodos del producto software a ser probados.	
<b>SALIDAS</b>	
Métodos depurados y la plantilla de funcionalidades a probar.	
<b>PLANTILLA</b>	
En esta tarea se incluye la Plantilla 1, la cual corresponde al listado de métodos a ser probados, en ella se describirán los datos de la empresa como son el nombre, dirección, ciudad y teléfono, también se incluye los datos de la persona encargada de la empresa como lo son el nombre, cargo y teléfono, en cuanto a el listado de métodos se tiene un identificador único que se le da a cada una pactadas por el equipo de trabajo, la valoración de la prioridad y viene acordada por el personal de prueba, y una breve descripción del método. Al final de la plantilla se incluirán las firmas tanto del personal de pruebas como del jefe de proyecto esto con la finalidad de que ambas partes estuvieron de acuerdo con lo consignado en el documento y también la descripción de cada uno de los ítems de la plantilla.	

#### 4.2.1.2. Tarea 2. Seleccionar los métodos y/o servicios a probar.

Título	<b>Seleccionar los métodos a probar.</b>
<b>DESCRIPCIÓN</b>	
Se seleccionarán los métodos a probar por el equipo de trabajo dependiendo de su posible aplicación para pruebas unitarias. Aquellos métodos identificados y valorados se someterán a prueba en primer lugar las que tengan mayor prioridad. Además de establecer los valores límites que alcanza cada método a probar, si el equipo de trabajo no cuenta con la descripción de los métodos a probar se elaborará un documento donde se especifique la descripción de cada método y lo que se quiere probar.	
<b>ROLES INVOLUCRADOS</b>	
Desarrolladores, personal de pruebas y jefe de proyecto.	
<b>ENTRADAS</b>	
Métodos depurados.	
<b>SALIDAS</b>	
Métodos a probar.	

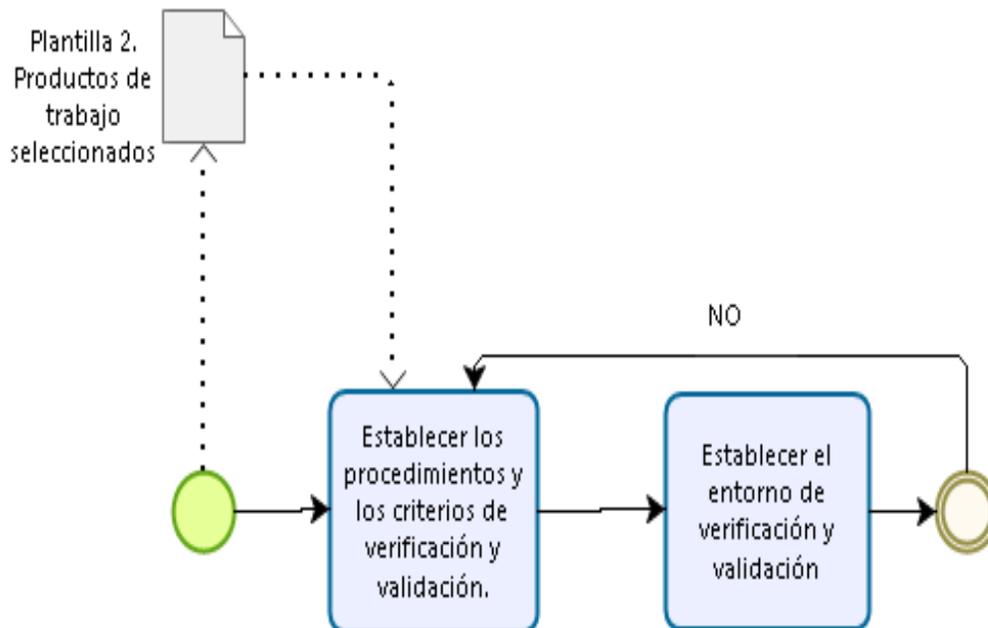
### 4.2.1.3. Tarea 3. Seleccionar los productos y métodos de verificación y validación

Título	<b>Seleccionar los productos y métodos de verificación y validación</b>
<b>DESCRIPCIÓN</b>	
<p>Permite la identificación del producto o del componente de producto a validar con base en los objetivos técnicos del producto software, así como también los métodos a usar para realizar la validación, estos son seleccionados con base en su contribución a los objetivos. Se tiene en cuenta la relación de los productos y componentes con los métodos. En cuanto a la validación para cada componente de producto se va a determinar el alcance de la validación. Las necesidades de los productos de trabajo se incluyen con los métodos de verificación los cuales tratan el enfoque para someterlo a los productos de trabajo y los enfoques específicos se utilizarán para verificar el cumplimiento de las necesidades del producto</p> <p><b><u>Algunos ejemplos de elementos que se pueden validar:</u></b>                  Producto y componentes de producto (unidades de hardware, software y documentación de servicios, manuales de usuario, materiales de formación documentación del proceso, protocolos de acceso, formatos de informes de intercambio de datos).</p> <p><b><u>Algunos ejemplos de métodos de verificación:</u></b>                  Evaluación de la conformidad de la implementación.                  Pruebas de cobertura de caminos.                  Pruebas basadas en tablas de decisión.                  Pruebas basadas en descomposición funcional.                  Reutilización de casos de prueba.</p> <p><b><u>Algunos ejemplos de productos de trabajo:</u></b>                  Lista de productos de trabajo seleccionados para la verificación o validación.                  Métodos de verificación o validación para cada producto de trabajo seleccionado.                  Necesidades para realizar la validación para cada producto o componente de producto.                  Restricciones de la validación para cada producto o componente de producto.</p> <p><b><u>Algunos ejemplos de métodos de validación:</u></b>                  Demostraciones de prototipos.                  Demostraciones funcionales.                  Pilotos de materiales de formación.</p>	
<b>ROLES INVOLUCRADOS</b>	
Desarrolladores, persona de pruebas, jefe de proyecto.	
<b>ENTRADAS</b>	
Identificar los productos y métodos de verificación y validación.	
<b>SALIDAS</b>	
Lista de productos y métodos de verificación y validación seleccionados.	
<b>PLANTILLA</b>	
<p>En esta tarea se incluye la Plantilla 2, la cual corresponde a los productos de trabajo seleccionados, en ella se describirán los datos de la empresa como son el nombre, dirección, ciudad y teléfono, en cuanto a los productos de trabajo seleccionados se tiene un código de producto, métodos de verificación y validación, un identificador de necesidad a resolver, alcance de validación y el tipo de asociación con otros servicios. Al final de la plantilla se incluirán las descripciones de cada uno de los ítems de la plantilla.</p>	
<b>Condicional</b>	<b>¿Los productos y métodos seleccionados satisfacen las necesidades?</b>
<p>Los productos y métodos seleccionados tienen que satisfacer las necesidades, en caso de no ser así se seleccionarán otros productos y métodos de verificación y validación, en caso de que si satisfaga las necesidades continuará a establecer el entorno de verificación y validación.</p>	

#### 4.2.2. Práctica 2. Establecimiento de criterios, procedimientos y entornos

En esta práctica están comprendidas las tareas de establecer los procedimientos y los criterios de verificación y validación, establecer el entorno de verificación y validación, cada una de ellas se detallará con una breve descripción, sus entradas y salidas y la plantilla si la tarea lo requiere.

**Vista de las tareas incluidas en la práctica:**



**Figura 4.5.** Tareas de la práctica de establecimiento de criterios, procedimientos y entornos.

#### 4.2.2.1. Tarea 4. Establecer los procedimientos y los criterios de verificación y validación

Título	<b>Establecer los procedimientos y los criterios de verificación y validación</b>
<b>DESCRIPCIÓN</b>	
<p>Permite el desarrollo de los procedimientos y criterios de validación y verificación que están alineados con las características de los productos seleccionados, los criterios de verificación y validación son definidos para asegurar que los productos de trabajo cumplen las necesidades y las restricciones en cuanto a la validación.</p> <p>La revisión entre pares es un buen procedimiento para la verificación ya que implica un examen metódico de productos de trabajo con la finalidad de identificar defectos para eliminarlos y recomendar cambios necesarios, este método es eficaz implementándolo mediante inspecciones, walkthroughs u otros métodos de revisión colegiados y se aplica principalmente a productos de trabajo desarrollados en los proyectos, pero también se pueden aplicar a otro tipo de productos como documentación y productos de trabajo de soporte.</p> <p><b>Ejemplos de fuentes de criterios de verificación y validación:</b> Necesidades de producto y de componente de producto, Estándares, Políticas de la organización, Parámetros de pruebas, Parámetros para establecer el equilibrio entre calidad y coste de pruebas, Tipos de productos de trabajo, Propuestas.</p> <p><b>Ejemplos de productos de trabajo:</b> Procedimientos de verificación, Criterios de verificación, Procedimientos de validación, Criterios de validación.</p>	
<b>ROLES INVOLUCRADOS</b>	
Desarrolladores, personal de pruebas, jefe de proyecto.	
<b>ENTRADAS</b>	
Los procedimientos y criterios de verificación y validación.	
<b>SALIDAS</b>	
Procedimientos y criterios de verificación y validación establecidos.	

#### 4.2.2.2. Tarea 5. Establecer el entorno de verificación y validación del producto.

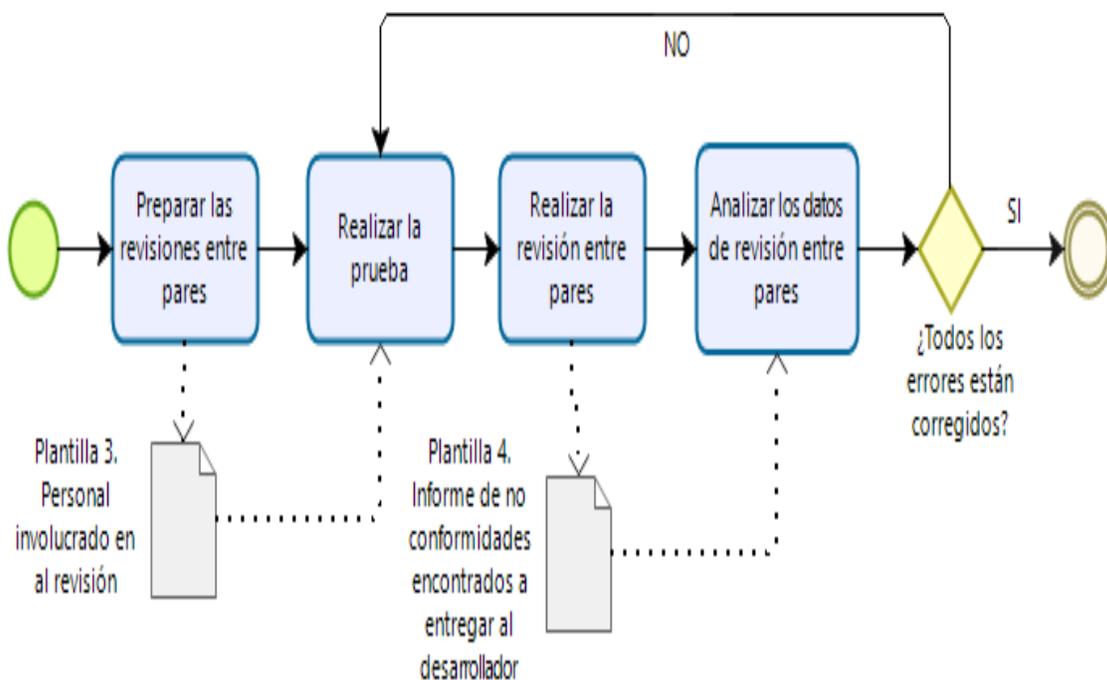
Título	<b>Establecer el entorno de verificación y validación del producto</b>
<b>DESCRIPCIÓN</b>	
<p>Esta tarea en cuanto a la verificación y validación dependen de los productos de trabajo seleccionados, métodos, componentes de producto y métodos de validación, en cuanto al entorno puede ser adquirido, desarrollado, reutilizado, modificado o utilizar una combinación de actividades y en cuanto a la selección se puede producir requisitos para la compra o el desarrollo del equipamiento, software y otros recursos. En cuanto al entorno de validación estará disponible cuando se necesite si se selecciona de forma temprana los productos o componente de producto a validar, los productos de trabajo a utilizar en la validación y los métodos de validación, este aspecto se debe controlar cuidadosamente para contemplar la replicación, el análisis de resultados y la revalidación de áreas problemáticas. También se seleccionará la herramienta a ser usada para las pruebas unitarias y la definición de un diccionario de variables.</p>	
<b>ROLES INVOLUCRADOS</b>	
Desarrolladores, personal de pruebas, jefe de proyecto.	
<b>ENTRADAS</b>	
Productos de trabajo seleccionados.	
<b>SALIDAS</b>	
Entorno de verificación y validación.	

### 4.2.3. Práctica 3. Preparación, ejecución de la prueba, realización y análisis de la revisión entre pares

En esta práctica están comprendidas las tareas de preparar las revisiones entre pares, realizar la prueba, realizar las revisiones entre pares, analizar los datos de la revisión entre pares, cada una de ellas se detallará con una breve descripción, sus entradas y salidas y la plantilla si la tarea lo requiere.

Se realiza mediante revisión entre pares, ya que esta práctica está comprendida tanto la verificación y validación, además de ser utilizada para realizar pruebas unitarias y verificar código.

**Vista de las tareas incluidas en la práctica:**



**Figura 4.6.** Tareas de la práctica preparación, ejecución de la prueba, realización y análisis de la revisión entre pares.

#### 4.3.2.1. Tarea 6. Revisión por pares

Título	<b>Revisión por pares.</b>
<b>DESCRIPCIÓN</b>	
<p>En esta tarea se identifica al personal que va a dar acompañamiento a la revisión, así mismo preparar y actualizar el material a utilizar durante las revisiones, estas revisiones pueden ser:</p> <ul style="list-style-type: none"> <li>• Listas de comprobación.</li> <li>• Criterios de revisión.</li> <li>• Calendario de revisión entre pares.</li> <li>• Criterios de entrada y salida para los productos de trabajo.</li> <li>• Material de formación de la revisión entre pares.</li> </ul> <p>Además, en esta práctica se realizará la prueba de automatización a los métodos del producto que van a ser sometidos a prueba.</p> <p>También, en esta actividad se encuentra y eliminan defectos en fases tempranas, dichas revisiones son de tipo incremental y se pueden realizar sobre productos de trabajo, diseño, pruebas e implementación. Estas revisiones deberán seguir unas pautas claves para su desarrollo óptimo como son:</p> <ul style="list-style-type: none"> <li>• Suficientemente preparadas.</li> <li>• Gestión y control de su realización.</li> <li>• Registro de datos consistentes y suficientes.</li> <li>• Registrar elementos de acción.</li> </ul> <p>Por último, en esta actividad se analiza los resultados obtenidos de la revisión entre pares, en caso de que se deba corregir algún aspecto del producto se debe hacer en el momento con la finalidad de entregarle al revisor la nueva versión para ser revisada, además se incluyen los script de pruebas para la replicación de las pruebas hechas a los métodos del producto.</p> <p><b>Ejemplo de productos de trabajo:</b> Resultados de la revisión entre pares, Cuestiones de la revisión entre pares, Datos de la revisión entre pares, Elementos de acción de la revisión entre pares.</p>	
<b>ROLES INVOLUCRADOS</b>	
Desarrolladores, personal de pruebas, jefe de proyecto	
<b>ENTRADAS</b>	
Lista de personal y material para realizar la revisión entre pares, ítems a probar.	
<b>SALIDAS</b>	
Personal y material seleccionado para realizar la revisión entre pares, lista de no conformidades de la revisión, lista de no conformidades de la revisión depuradas.	
<b>PLANTILLAS</b>	
<ul style="list-style-type: none"> <li>• Se incluye la plantilla <b>3</b>: corresponde al personal involucrado, en ella se describirán los datos de la empresa como son el nombre, dirección, ciudad y teléfono, en cuanto al personal involucrado se tiene el nombre de los revisores, la fecha de la revisión, el producto de trabajo a revisar, los criterios de entrada y criterios de salida y una lista de comprobación. Al final de la plantilla se incluirán las descripciones de cada uno de los ítems.</li> <li>• Se incluye la plantilla <b>4</b>: corresponde al informe de no conformidades, en ella se describirán los datos de la empresa como son el nombre, dirección, ciudad y teléfono, en cuanto al informe de no conformidades se tiene el código de producto, total de no conformidades, descripción de no conformidades, tipo de no conformidad, desarrollador principal, aspecto detectado y estado del defecto. Al final de la plantilla se incluirán las descripciones de cada uno de los ítems de la plantilla.</li> </ul>	
<b>Condición</b>	<b>¿Todos los errores están corregidos?</b>
Los errores encontrados deben estar ya corregidos. En caso que ya estén corregidos se seguirá a realizar la verificación y la validación, en caso de que no estén totalmente corregidos se volverá a hacer la revisión entre pares de los elementos encontrados con errores y con las personas netamente involucradas.	

#### 4.2.4. Práctica 4. Realización de la verificación, validación y análisis de resultados

Comprende las tareas de realizar la verificación y validación, analizar los resultados de verificación y validación, cada una de ellas se detallará con una breve descripción, sus entradas y salidas y la plantilla si la tarea lo requiere.

Vista de las tareas incluidas en la práctica:

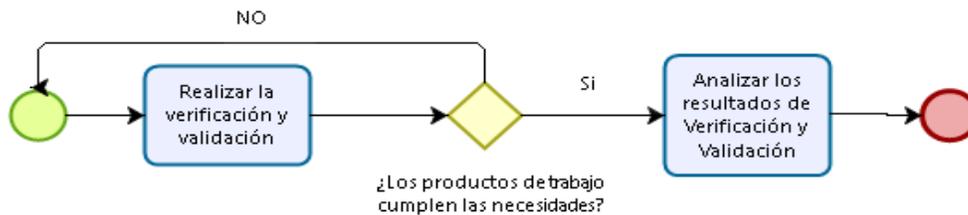


Figura 4.7. Tareas de la práctica verificación, validación y análisis de resultados.

##### 4.2.4.1. Tarea 7. Realizar la verificación y la validación del producto de trabajo seleccionado

Título	Realizar la verificación y la validación del producto de trabajo seleccionado
<b>DESCRIPCIÓN</b>	
En esta tarea se especificará la realización de la verificación y la validación, a la hora de realizar la verificación los resultados esperados ahorrarán el coste considerablemente dado que el producto está aislado de fallos y re-trabajo asociados con la resolución de problemas, en cuanto a la validación es permitida de acuerdo a los métodos, procedimientos y criterios en donde se realizan actividades de validación y se recogen los datos resultantes de acuerdo a los métodos, procedimientos y criterios establecidos, en cierto punto se deberían documentar los procedimientos de validación tal como se ejecutaron y se deberían anotar las desviaciones que ocurren durante la ejecución.	
<b>Ejemplos de productos de trabajo:</b> Resultados de la verificación, Informes de la verificación, Demostraciones, Registro de ejecución de los procedimientos, Informes de validación, Resultados de la validación, Matriz de referencias cruzadas de la validación, Registro de ejecución de los procedimientos.	
<b>ROLES INVOLUCRADOS</b>	
Desarrolladores, personal de pruebas, jefe de proyecto.	
<b>ENTRADAS</b>	
Resultados de la revisión entre pares.	
<b>SALIDAS</b>	
Producto aislado de fallos y re-trabajo, Datos resultantes.	
<b>Condicional</b>	<b>¿Los productos de trabajo cumplen las necesidades?</b>
Los productos de trabajo tienen que cumplir las necesidades establecidas previamente, de ser así se seguirá a analizar los resultados de la verificación y la validación obtenidos en todo el proceso, en caso de que los productos de trabajo no cumplan con las necesidades se volverá a establecer los procedimientos y criterios de verificación y validación y se hará de nuevo la revisión entre pares con los nuevos ítems agregados.	

#### 4.2.4.2. Tarea 8. Analizar los resultados de verificación y validación.

Título	<b>Analizar los resultados de verificación y validación</b>
<b>DESCRIPCIÓN</b>	
<p>En esta tarea los datos de las pruebas, inspecciones, demostraciones o evaluaciones se analizan y comparan con los resultados reales teniendo en cuenta los criterios de verificación y validación previamente descritos, esto con la finalidad de determinar la aceptación.</p> <p>Los resultados de las pruebas se registran como evidencia que se ha hecho la verificación y los informes de análisis indican si se cumplieron las necesidades de validación, en el caso de que existan deficiencias los documentos indican el grado de éxito o fallo y se clasifican las posibles causas de fallo.</p> <p>También se tiene en cuenta que cada producto de trabajo se analiza incrementalmente para así asegurar el cumplimiento de las necesidades y los resultados recogidos de las pruebas, inspecciones o revisiones, éstos se comparan con los criterios de evaluación establecidos para así determinar si se continúa o si se tratan cuestiones de requisitos o de diseño en los procesos de desarrollo de requisitos o de solución técnica.</p> <p>Por otro lado, los informes de análisis o la documentación de los métodos pueden indicar resultados de mala verificación y esto se debe a problemas del método, problemas de criterios o problemas de entorno de verificación. En cuanto a los informes de análisis o documentación de ejecución de la validación indican que unos malos resultados de las pruebas se deben a un problema del procedimiento de validación o un problema del entorno de validación.</p> <p><b><u>Ejemplos de productos de trabajo:</u></b></p> <ul style="list-style-type: none"> <li>• Informes de problemas.</li> <li>• Peticiones de cambio de los métodos.</li> <li>• Criterios y del entorno de verificación.</li> <li>• Informes de deficiencias de la validación.</li> <li>• Cuestiones de validación y petición de cambio del procedimiento.</li> </ul>	
<b>ROLES INVOLUCRADOS</b>	
Desarrolladores, personal de pruebas, jefe de proyecto.	
<b>ENTRADAS</b>	
Datos resultantes de la verificación y la validación.	
<b>SALIDAS</b>	
Resultados.	

## **Capítulo 5**

### **Evaluación del procedimiento**

En este capítulo se presenta la revisión por expertos profesionales del área y el estudio de caso, los cuales fueron empleados como método de evaluación del procedimiento propuesto. En cuanto al estudio de caso el entorno sobre el cual se ejecutó el procedimiento estaba conformado por una aplicación web de la empresa NAMTRIK DEV S.A.S.

#### **5.1. Antecedentes**

Se han identificado propuestas para la automatización de pruebas unitarias las cuales no siguen una estructura o proceso que conlleve a la buena práctica del procedimiento de automatización de pruebas unitarias para pequeñas empresas desarrolladoras de software. Estas propuestas se centran en la aplicación directa de las pruebas unitarias a los artefactos o componentes, dejando a un lado la documentación requerida para los casos de prueba, los fallos obtenidos para el aprendizaje de la empresa, la bitácora de artefactos ya probados, entre otros. Los estudios encontrados como Xiaochum et al. (2008), Böhme y Paul (2015), Joy y Pratap (2015), Sávio y duarte (2013), Morabdeikay et al. (2014), Safronaun y turlo (2011), Figueiredo y Ferreira (2012), Candea et al. (2010) y Ha, Chae y Moon (2009) se basan en automatización de pruebas de componentes, automatización de pruebas de integración, automatización de pruebas API y automatización de pruebas GUI, las cuales no son el enfoque dado al procedimiento puesto que el enfoque son las pruebas unitarias de software. En el proceso de ejecución de pruebas unitarias se encontraron falencias a la hora de implementar dichas pruebas, dado que algunas de ellas no llevaban a cabo la implementación de pruebas o simplemente probaban los productos superficialmente

Pregunta de investigación principal:

- ¿Cómo orientar la incorporación de buenas prácticas de automatización de pruebas unitarias en pequeñas organizaciones desarrolladoras de software?

## **5.2. Diseño**

Tomando como referencia el estudio y el análisis realizado sobre la mejora de procesos para el desarrollo de productos y servicios planteados por CMMI-DEV 1.3 (2010) en las áreas de proceso de verificación y validación donde además se tomaron en cuenta los elementos pertinentes y que fueron sometidos a una revisión por expertos. Además, se toma como enfoque las pruebas unitarias de software y el elemento de revisión es la revisión entre pares.

El objeto de estudio es un procedimiento diseñado para ser implementado en empresas pequeñas de desarrollo de software, el cual está compuesto por prácticas de automatización de pruebas unitarias de software, las cuales a su vez están conformadas por tareas quienes tienen sus entradas, salidas, roles y plantillas. El procedimiento contiene una serie de plantillas diseñadas para ser diligenciadas en corto tiempo. Las denominadas practicas clave del procedimiento corresponden a: 1) Identificación y selección, 2) Establecimiento de criterios, procedimientos y entornos, 3) Preparación, ejecución de la prueba, realización y análisis de la revisión entre pares, 4) Realizar la verificación y validación y 5) Análisis de resultados.

El líder de pruebas quien a su vez puede también ser el líder de proyecto es quien se encarga de solucionar las incongruencias o discrepancias que pueda tener el personal de pruebas, en las prácticas de revisión por pares en caso tal de que el personal de pruebas tenga discrepancias el líder de pruebas entrará como árbitro o quien dirime la discusión para este caso.

## 5.3. Revisión de expertos

La revisión por expertos es un método de validación de investigaciones que ayuda a verificar su fiabilidad, esta revisión según lo planteado por Michalus J. et. al. (2015) se compone de las fases:

1. Identificación de las características a evaluar.

Este procedimiento se crea con el fin de obtener una información detallada, ordenada, sistemática que contenga instrucciones de cómo llevar a cabo la automatización de pruebas unitarias enfocado a pequeñas organizaciones desarrolladoras de software para esta fase se evaluaron las prácticas de identificación y selección, establecimiento de criterios, procedimientos y entornos, preparación, realización y análisis de la revisión entre pares, realización de la verificación y validación y análisis de resultados y sus respectivas tareas.

2. Selección de expertos.

En la evaluación de expertos para el procedimiento de automatización de pruebas unitarias participaron tres (3) profesionales adscritos a la academia y además, cuentan con experiencia de estar vinculados a empresas de software, el listado de los profesionales que participaron en la evaluación de software se presenta en la Tabla 5.3.1.

Nombre	Institución	P <sup>5</sup>
Mg. Sandra Lorena Buitrón Ruiz	Unicauca (Popayán)	P1
PhD. Oscar Mauricio Caicedo Rendón	Unicauca (Popayán), SoMos Ltda.	P2
Mg. Ricardo Antonio Zambrano Segura	Unicauca (Popayán), SoMos Ltda.	P3

**Tabla 5.1.** Participantes de la evaluación por expertos.

---

<sup>5</sup> P: Participante

### 3. Diseño del instrumento de recolección de datos.

Se diseñó un documento resumen del Capítulo 4 el cual está como anexo A donde se explica brevemente la descripción de cada práctica del procedimiento.

### 4. Consulta a expertos.

Se envió la información correspondiente a el modelo propuesto y el resumen del capítulo 4 para su revisión, posteriormente se planeó una fecha de reunión para la explicación del procedimiento y la retrospectiva del mismo.

Fechas de reunión:

Mg. Sandra Lorena Buitrón Ruiz: 13 – Abril – 2018

PhD. Oscar Mauricio Caicedo Rendón: 27 – Abril -2018

Mg. Ricardo Antonio Zambrano Segura: 27 – Abril -2018

Agenda a seguir:

Presentación de la propuesta - Retroalimentación

### 5. Procesamiento de la información (Análisis y reporte de resultados)

En esta fase se analiza la información recolectada en la revisión por expertos y se indica los aspectos que influyeron en la afinación del procedimiento. A partir de la revisión del documento sintetizado (resumen) del Capítulo 4 y los archivos de audio recolectados en las secciones de la revisión por expertos estos resultados y análisis se clarificaron en tres categorías las cuales están consignadas en los aspectos positivos (Tabla 5.3.2), aspectos por mejorar (Tabla 5.3.3) y Observaciones (Tabla 5.3.4.).

<b>Aspectos Positivos</b>	
P1	El procedimiento ahorra gastos en producción.
P2	El procedimiento es útil dado que ayuda a ahorrar tiempo y costos en una empresa a la hora de realizar pruebas, ya que se explica y detalla paso a paso la realización de pruebas y se consigna toda la información recolectada en las plantillas.
P3	El procedimiento es bueno dado que sirve como referencia al ahora de ejecutar pruebas.

**Tabla 5.2.** Aspectos positivos del procedimiento.

<b>Aspectos por mejorar</b>	
P1	<ul style="list-style-type: none"> <li>• Replantear la descripción de las tareas 1,2 y 3 de la práctica 1 para especificar con mayor claridad lo que se quiere lograr con cada tarea puesto que la descripción hace referencia al levantamiento o análisis de requerimientos de un proyecto.</li> <li>• Acotar la descripción de la tarea 3 de la práctica 1 especificando que se refiere al producto a evaluar.</li> </ul>
P2	<ul style="list-style-type: none"> <li>• La práctica 4 se debe separar en dos tareas, una tarea es realizar la verificación y la validación y la otra tarea es analizar los resultados de la verificación y la validación.</li> <li>• En la práctica 3 (Preparación, ejecución de la prueba, realización y análisis de la revisión entre pares) se menciona la revisión entre pares, pero no se menciona una tarea que indique donde se hacen las pruebas.</li> </ul>
P3	<ul style="list-style-type: none"> <li>• Se sugiere agregar una tarea donde se especifique en donde se lleva a cabo la automatización de pruebas dado que la descripción diera a entender que son pruebas manuales.</li> <li>• En la plantilla 1 (Listado de métodos) se debe especificar si los métodos del proyecto tienen prioridad y si se tiene dependencia o relación entre los mismos.</li> </ul>

**Tabla 5.3.** Aspectos a mejorar en el procedimiento.

<b>Observaciones</b>	
P1	<ul style="list-style-type: none"><li>• Visualizar los elementos de CMMI más importantes que se están teniendo en cuenta.</li><li>• En general el Capítulo 4 requiere un mejor contexto acerca del propósito de la propuesta, dado que no se permite ubicar exactamente la parte del proyecto en donde se hacen estas tareas.</li></ul>
P2	<ul style="list-style-type: none"><li>• Hay un error en la práctica 3 (Preparación, ejecución de la prueba, realización y análisis de la revisión entre pares) porque el retorno de la condicional (¿Todos los errores están corregidos?) no finaliza en la tarea realizar la revisión entre pares si no en la tarea realizar la prueba.</li><li>• Los elementos principales del procedimiento como lo son las actividades deben llamarse prácticas y cada práctica tiene contenida tareas, además se necesita saber las entradas y las salidas de las tareas.</li></ul>
P3	<ul style="list-style-type: none"><li>• En la última condicional no se mencionan los requisitos si no las necesidades.</li><li>• En la plantilla 4 (Informe de no conformidades) se debe organizar mejor las columnas de total de no conformidades y el tipo de no conformidad.</li></ul>

**Tabla 5.4.** Observaciones realizadas al procedimiento.

**Modificaciones realizadas al procedimiento a partir de la revisión por expertos:**

En la tabla 5.3.5 se señalan los elementos modificados en el procedimiento a partir del proceso de revisión por expertos, la versión definitiva se presenta en el Capítulo 4.

Componente	Modificación
Procedimiento	<ol style="list-style-type: none"> <li>1. Se incluyó la tarea “realizar la prueba” en la práctica de preparación, ejecución de la prueba, realización y análisis de la revisión entre pares.</li> <li>2. La práctica “Realización de la verificación, validación y análisis de resultados” se dividió en 2 practicas “Realizar la verificación y validación” y “Analizar los resultados de verificación y validación”.</li> <li>3. El condicional “todos los errores están corregidos” de la practica 3 que en la opción del NO regresaba a la tarea realizar la revisión entre pares, se lo redirecciono a la tarea realizar la prueba.</li> </ol>
Descripciones de las tareas	Las tareas de la practica 3 se las describe de mejor manera, que hagan referencia a la evaluación de las funcionalidades de un producto mediante automatización de pruebas que es lo que se busca con el procedimiento.
Plantillas	<ol style="list-style-type: none"> <li>1. La plantilla 1 (Listado de métodos) se organizó de tal manera que describa las prioridades y dependencia de cada método del producto a evaluar.</li> <li>2. En la plantilla 2 (Listado de productos de trabajo seleccionados) se organizaron los nombres de las variables para que haya correlación entre nombres.</li> <li>3. La plantilla 3 (Personal involucrado) se modifica agregando el total de no conformidades y el tipo de no conformidades</li> </ol>
Elementos principales del procedimiento	Se agregan las entradas y salidas que tendrán los elementos principales del procedimiento, también se cambia el nombre de actividades por prácticas, y lo que se tenía como practicas se le pasa a llamar tareas.

**Tabla 5.5.** Modificaciones realizadas al procedimiento.

## **5.4. Estudio de caso**

El estudio de caso se empleó como método de evaluación del procedimiento de automatización de pruebas, una vez realizadas las correcciones y modificaciones de la revisión por expertos. El entorno sobre el cual se ejecutó este proceso estaba conformado por una aplicación web de la empresa Namtrik Development S.A.S. los componentes de este estudio de caso se tomaron como referente de Brereton P. et. al. (2008)

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

La empresa Namtrik Development S.A.S. Es una empresa de desarrollo de aplicaciones móviles y plataformas web de la ciudad de Popayán. La aplicación cuenta con un total de 146 usuarios activos y los procedimientos a ser probados son inicio de sesión, gestión de usuarios, gestión de agencias, gestión de regionales y gestión de zonas de un almacén. En este estudio de caso se desea evaluar la viabilidad y desempeño del procedimiento, como resultado de su implementación en la organización en la que se pretende incorporar mejores prácticas de automatización de pruebas unitarias de software utilizando los recursos y el personal que posee la empresa. La empresa no practica automatización de pruebas, por lo cual se espera que incorporen prácticas de automatización de pruebas unitarias y evaluar su desempeño.

El personal de pruebas corresponde a las personas de la organización que ocupan cargos involucrados en la ejecución de las pruebas en los artefactos software, como son: líder de pruebas, téster y/o desarrolladores.

## **5.6. Procedimiento de campo**

El procedimiento de campo está clasificado según las necesidades de la organización, iniciando con las actividades de preparación, relacionadas con obtener información de la empresa y los artefactos a ser sometidos a pruebas para generar posteriormente un diagnóstico del estado de ejecución de pruebas y proceder a realizar el procedimiento. Se ejecutará la información necesaria sustentada en las métricas definidas en el procedimiento. En la Figura 5.1. se puede observar el flujo del procedimiento a seguir.

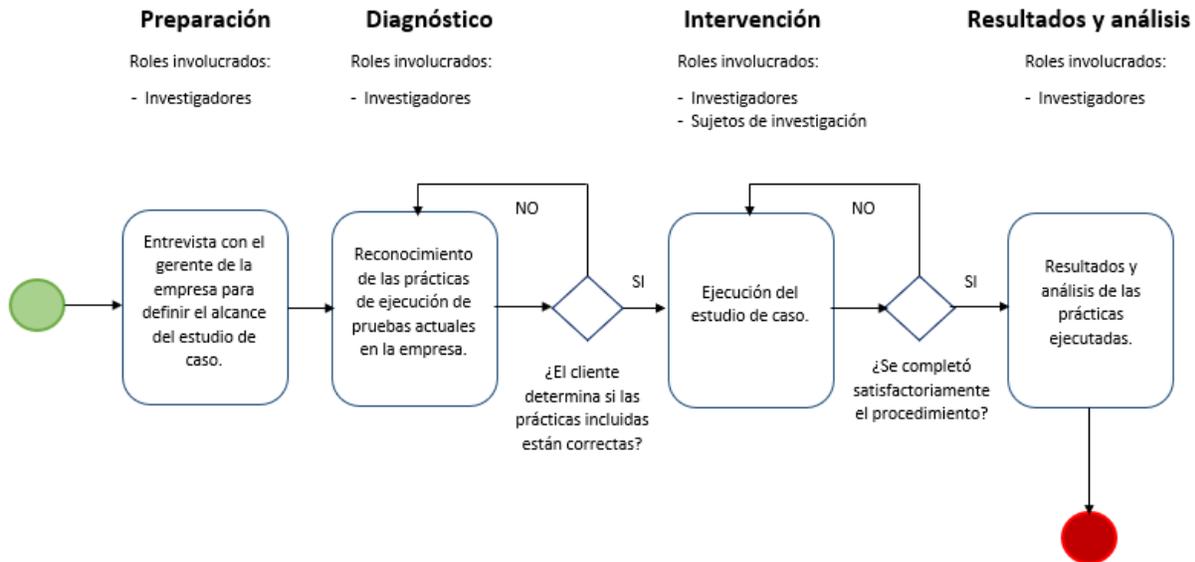


Figura 5.1. Diagrama del procedimiento de campo.

## 5.7. Intervención

En la intervención realizada se siguieron los pasos del procedimiento definidos anteriormente, donde se presentaron los siguientes pasos resultantes:

### 5.7.1. Preparación

Se realizó una reunión con el grupo de trabajo de Namtrik Development S.A.S. en el cual se recolectó la información necesaria del proyecto, sobre el cuál la empresa requirió implementar el procedimiento, con la finalidad de implementarlo la empresa desplegó una copia del proyecto que está en producción y entregó los ítems a los cuales se ejecutaron las pruebas unitarias.

### 5.7.2. Diagnóstico

Se encontró que la empresa no ejecuta un plan pruebas unitarias en sus proyectos, por lo cual se implementó el procedimiento al proyecto más reciente y con mayor número de usuarios activos, dado que el proyecto abarcaba muchos procesos se requirió acotarlo en los más fundamentales, los procesos para la implementación del procedimiento fueron seleccionados por la empresa.

### 5.7.3. Intervención

En la figura 5.2. se muestran las actividades que se llevaron a cabo en la ejecución del estudio de caso, incluyendo las prácticas clave de manera general y las actividades de retroalimentación y corrección necesarias para el procedimiento propuesto.

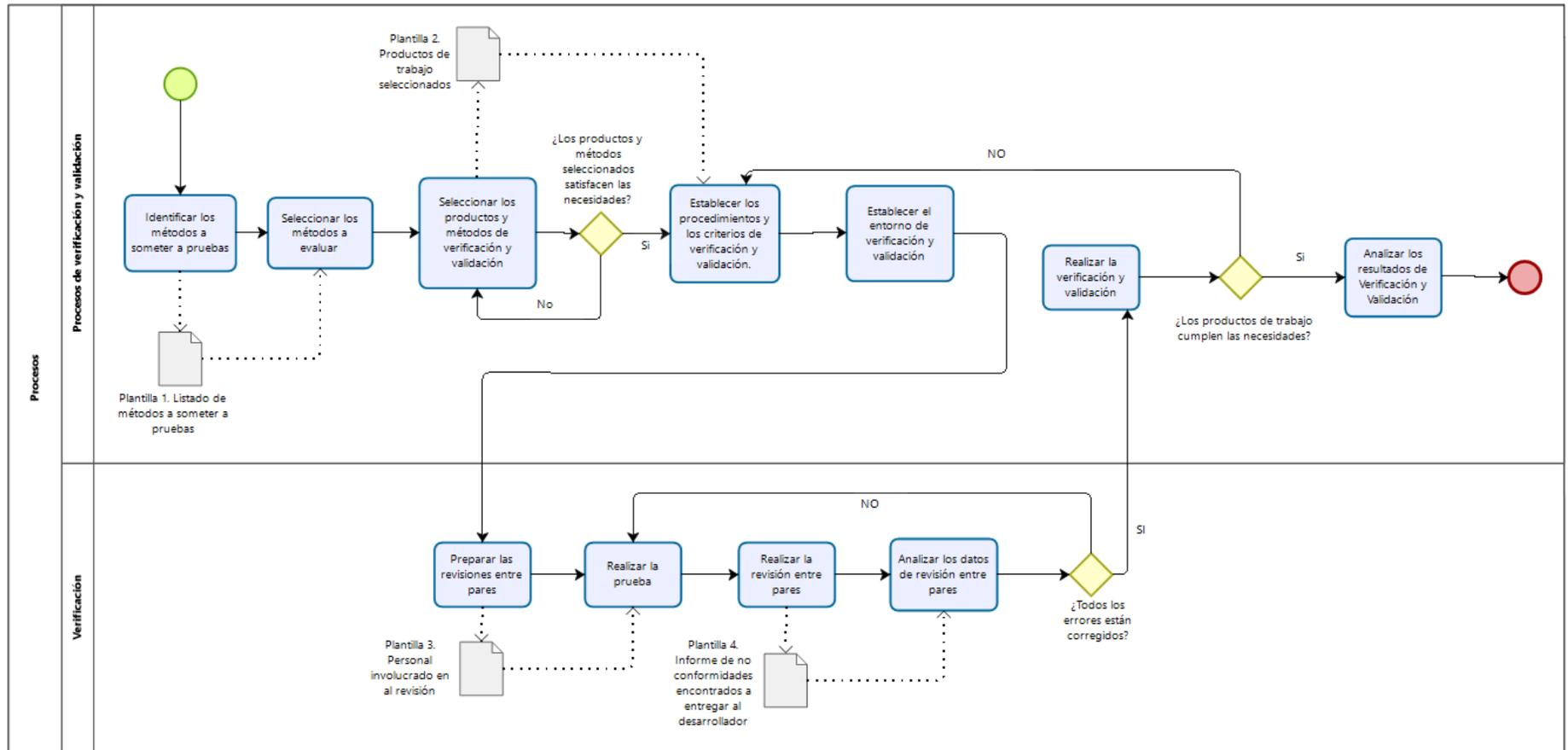


Figura 5.2. Diagrama de actividades a ejecutar en el estudio de caso.

#### 5.7.4. Ejecución del procedimiento mediante el estudio de caso

**Tarea 1. Identificar los métodos a probar.** En esta tarea se identificó los métodos del proyecto para ser sometido a pruebas y se las evidenció en la plantilla denominada listado de métodos. En la Figura 5.3. se presenta la plantilla diligenciada en la empresa Namtrik Development S.A.S.

### LISTADO DE MÉTODOS

<b>Datos de la empresa</b>	Nombre: Dirección: Ciudad: Teléfono:	Namtrik Development S.A.S. Carrera 9 # 4N - 29 Of 704 Popayán - Cauca (2) 8339016
<b>Datos de la persona encargada</b>	Nombre: Cargo: Teléfono:	Cristhiam Gabriel Fernández Ruales Representante Legal (2) 8339016

Código	Prioridad	Descripción del método a probar	Dependencia
N-001	5	Iniciar sesión	N-002
N-002	4	Usuarios	
N-003	4	Regionales	N-002
N-004	4	Zonas	N-002, N-003
N-005	4	Agencias	N-002, N-004

Figura 5.3. Ejemplo de la plantilla para listado de métodos.

**Tarea 2. Seleccionar los métodos a evaluar.** En esta tarea se describió los métodos seleccionados en la plantilla para el listado de métodos y se basa en el listado de requerimientos y se especificaron los valores límites para cada método.

**Tarea 3. Seleccionar los productos y métodos de verificación y validación.** En esta tarea se seleccionaron los productos de trabajo a ser sometidos a prueba, a estos productos se les definió el método de validación y verificación, el alcance de la validación y la herramienta usada para la ejecución de la prueba. En la figura 5.4. se presenta la plantilla del listado de productos de trabajo seleccionados por la empresa Namtrik Development S.A.S.

LISTADO DE PRODUCTOS DE TRABAJO SELECCIONADOS					
<b>Datos de la empresa</b>		Nombre: Dirección: Ciudad: Teléfono:	Namtrik Development S.A.S. Carrera 9 # 4N - 29 Of 704 Popayán - Cauca (2) 8339016		
Código de producto	Métodos de validación y	Identificador de necesidad a	Alcance de	Tipo de asociación con otros servicios	Herramienta
PROD-001	Demostraciones funcionales, Tablas de decisión	N-001	Comportamiento Operativo	En producción	PHPUnit
PROD-002	Demostraciones funcionales, Tablas de decisión	N-002	Comportamiento Operativo	Plataforma en funcionamiento	PHPUnit
PROD-003	Demostraciones funcionales, Tablas de decisión	N-003	Comportamiento Operativo	Plataforma en funcionamiento	PHPUnit
PROD-004	Demostraciones funcionales, Tablas de decisión	N-004	Comportamiento Operativo	Plataforma en funcionamiento	PHPUnit
PROD-005	Demostraciones funcionales, Tablas de decisión	N-005	Comportamiento Operativo	Plataforma en funcionamiento	PHPUnit

**Figura 5.4.** Ejemplo de la plantilla para el listado de productos de trabajo seleccionados.

**Tarea 4. Establecer los procedimientos y los criterios de verificación y validación.** En esta tarea se tomó como documento de entrada la plantilla para el listado de productos de trabajo seleccionados y se estableció los procedimientos y los criterios de verificación y validación entre los cuales se establecieron los parámetros de pruebas, los tipos de productos de trabajo y el plan de pruebas.

**Tarea 5. Establecer el entorno de verificación y validación del producto.** En esta tarea se especificó el escenario de pruebas el cual se dependía del alcance del producto y de las funcionalidades a ser probadas, las pruebas realizadas en la empresa Namtrik Development S.A.S. fueron hechas mediante la revisión entre pares, también se realizó la identificación de un diccionario de variables con la finalidad de agilizar la realización de las pruebas unitarias.

**Tarea 6. Preparar la revisión entre pares.** En esta tarea se especificó el personal involucrado para realizar la revisión entre pares, en donde se obtuvo cinco plantillas (una por cada producto) en las cuales se establecieron los criterios de entrada, criterios de salida y listado de comprobación, en el listado de comprobación se describieron las pruebas realizadas, también se adicionó en script con la prueba realizada para ser replicada y el chequeo para llevar un listado de el cumplimiento de las pruebas. En la Figura 5.5. se presenta la plantilla para el personal involucrado en las pruebas realizadas a un ítem del producto.

## PERSONAL INVOLUCRADO

<b>Datos de la empresa</b>	Nombre: Dirección: Ciudad: Teléfono:	Namtrik Development S.A.S. Carrera 9 # 4N - 29 Of 704 Popayán - Cauca (2) 8339016
----------------------------	---	--

<b>Nombre del revisor 1</b>	<b>Nombre del revisor 2</b>	<b>Fecha de revisión</b>	<b>Código de producto</b>
Marlon Mamián	Darwin Montenegro	25/07/2018	PROD-001

<b>Criterio de entrada</b>
Nombre de usuario
Contraseña

<b>Criterio de salida</b>
Acceso concedido (Status 200)
Acceso denegado (Status 400)

Lista de comprobación		
Criterio	Script	Chequeo
Criterios de entrada correctos	LoginTest.php	<input checked="" type="checkbox"/>
Criterio de entrada contraseña incorrecta	LoginTest.php	<input checked="" type="checkbox"/>
Criterio de entrada nombre de usuario incorrecto	LoginTest.php	<input checked="" type="checkbox"/>
Campo nombre de usuario modificado	LoginTest.php	<input checked="" type="checkbox"/>

**Figura 5.5.** Ejemplo de plantilla personal involucrado para la ejecución de la prueba.

**Tarea 7. Realizar la prueba.** En esta tarea se realizaron las pruebas a los productos seleccionados, se utilizó el entorno de pruebas PHPUnit en donde se obtuvo los scripts de las diferentes pruebas realizadas, en la Figura 5.6. se muestra el script de pruebas realizada al método Login, en la Figura 5.7. y 5.8. se muestran los scripts de pruebas realizada al método Registrar, en la Figura 5.9. se muestra el script de pruebas realizadas al método Crear Agencias, en las Figuras 5.10. y 5.11. se muestran los scripts realizados al método Crear Regionales, se tiene en cuenta que se presentan las pruebas realizadas ocultando la información relevante de la empresa Namtrik Development S.A.S. dado que se firmó un contrato de confidencialidad a lo que la empresa no nos permite evidenciar la información del producto software desarrollado.

```
<?php

namespace Tests\Feature;

use Tests\TestCase;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;

class LoginTest extends TestCase
{
    public function testExample()
    {
        $response = $this->json('POST', '/login', ['username' => 'superadmin',
        'password' => 'namtrik123']); // Correcta
        //$response = $this->json('POST', '/login', ['username' => 'superadmin',
        // 'password' => 'namtrik1234']); // Contraseña incorrecta
        //$response = $this->json('POST', '/login', ['username' => 'superadmin1',
        // 'password' => 'namtrik123']); // Correo incorrecto
        //$response = $this->json('POST', '/login', ['username' => 'superadmin',
        // 'password' => 'namtrik123']); // Dato esperado es incorrecto

        $response->assertStatus(200);

        //$this->assertTrue(true);
    }
}
```

**Figura 5.6.** Script de pruebas del método Login.

```
<?php

namespace Tests\Feature;

use Tests\TestCase;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;

class RegisterTest extends TestCase
{
    public function testExample()
    {
        $response = $this->json('POST', '/user_create',
            ['first_name' => 'Tayzon',
            'last_name' => 'Junior',
            'password' => '1234567891',
            'password_confirmation' => '1234567891',
            'phone' => '1234567891',
            'role' => 'Administrador',
            'subsidiary_id' => '',
            'username' => 'namtrik123',
            'zone_id' => '',
            ]);
        $response->assertJson(['message' => 'Unauthenticated.']);
        $response->assertStatus(401);
        //dd($response->decodeResponseJson());
        // $this->assertJson('message', 'Usuario creado correctamente');
        /* $this->assertJson(
            ['message' => 'Unauthenticated.' ]
        );*/
        /* $response->seeJsonStructure(['message']);*/
        /* $this->post('/user_create')
            ->seeJsonStructure([
                'name',
                'pet' => ['name', 'age']
            ]);*/
    }
}
```

**Figura 5.7.** Primer Script de pruebas del método Registrar.

```
<?php

namespace Tests\Feature;

use Tests\TestCase;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;

class RegisterTrueTest extends TestCase
{
    public function testExample()
    {
        $this->json('POST', '/login', ['username' => 'superadmin', 'password' => 'namtrik123']);

        $response = $this->json('POST', '/user_create',
        ['first_name' => 'Tayzon 1',
        'last_name' => 'Junior',
        'password' => '1234567891',
        'password_confirmation' => '1234567891',
        'phone' => '44444',
        'role' => 'Administrador',
        'subsidiary_id' => '',
        'username' => 'user2',
        'zone_id' => '',
        ]);

        $response->assertJson(['message' => 'Usuario creado correctamente,']);

        $response->assertStatus(200);

        //dd($response->decodeResponseJson());

        // dd($response->decodeResponseJson());
    }
}
```

**Figura 5.8.** Segundo Script de pruebas del método Registrar.

```
<?php

namespace Tests\Fixture;

use Tests\TestCase;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;

class CreateAgencyTrueTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testExample()
    {
        $this->json('POST', '/login', ['username' => 'superadmin', 'password' => 'namtrik123']);

        $response = $this->json('POST', '/subsidiary_create',
        [
            'subsidiary_address' => 'Zona 1',
            'subsidiary_city_town' => 'Julian',
            'subsidiary_city_town_code' => 19142,
            'subsidiary_name' => 'Zona 1',
            'subsidiary_phone_number' => '12a43455',
            'subsidiary_state' => 'Cauca',
            'subsidiary_state_code' => '19',
            'zone_id' => 10,
        ]
        );

        $response->assertStatus(200);
    }
}
```

**Figura 5.9.** Script de pruebas del método Crear Agencias.

```
<?php

namespace Tests\Feature;

use Tests\TestCase;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;

class CreateRegionsTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testExample()
    {
        $this->json('POST', '/login', ['username' => 'superadmin', 'password' => 'namtrik123']);

        $response = $this->json('POST', '/regional_create',
            ['regional_name' => 'Regional Oriente',
            'user_id' => 45,
            ]);

        //$response->assertJson(['message' => '901',]); // El jefe de zona ya existe

        $response->assertStatus(400);
    }
}
```

**Figura 5.10.** Primer script de pruebas del método Crear Regionales.

```
<?php

namespace Tests\Feature;

use Tests\TestCase;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;

class CreateRegionsTrueTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testExample()
    {
        $this->json('POST', '/login', ['username' => 'superadmin', 'password' => 'namtrik123']);

        $response = $this->json('POST', '/regional_create',
            ['regional_name' => 'Regional OccidenSte',
            'user_id' => 28,
            ]);

        //$response->assertJson(['message' => 'Región creada correctamente',]);

        $response->assertStatus(400);
    }
}
```

**Figura 5.11.** Segundo script de pruebas del método Crear Regionales.

### **Tarea 8. Realizar la revisión entre pares.**

En esta tarea se obtuvo un conglomerado de los resultados obtenidos en las pruebas, en donde se encontró el tipo de no conformidad, el total de no conformidades halladas en las pruebas, la descripción de la no conformidad, el desarrollador a cargo, los aspectos detectados en las pruebas es decir si la prueba estuvo suficientemente preparada, hubo buena gestión y control en su realización, si los registros de datos fueron consistentes y suficientes y un registro de elementos de acción, además se tiene un estado del defecto para llevar un control de su proceso. En la Figura 5.6. se presenta la plantilla para el informe de no conformidades encontrados en el producto.

INFORME DE NO CONFORMIDADES									
Datos de la empresa		Nombre:	Namtrik Development S.A.S.						
		Dirección:	Carrera 9 # 4N-29 Of 704						
		Ciudad:	Popayán - Cauca						
		Teléfono:	(2) 8339016						
Código del	Tipo de no	Total de no conformida	Descripción no conformidades	Desarrollador	Aspecto detectado				Estado del defecto
					SP	GC	RDC	REA	
PROD-001	Ninguna	0	Ninguna	Carlos Ordoñez	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Finalizado
PROD-002	Status 200	2	Registro de usuarios con datos incorrectos (Nombre con números, Teléfono con caracteres)	Carlos Ordoñez	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	En Observación
PROD-003	Ninguna	0	Ninguna	Carlos Ordoñez	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Finalizado
PROD-004				Carlos Ordoñez	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Finalizado
PROD-005	Status 200	2	Registro de agencias con los datos incorrectos (Campo teléfono con caracteres, Zonas y departamentos sin validar codigos ni datos)	Carlos Ordoñez	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	En Observación

Figura 5.12. Ejemplo del informe de no conformidades.

**Tarea 9. Analizar los datos de la revisión entre pares.** En esta tarea se tomó como datos de entrada la plantilla del informe de no conformidades y se presentó a la empresa para su corrección.

**Tarea 10. Realizar la verificación y la validación.** En esta tarea se verificó que los procedimientos y los criterios de verificación y validación que se establecieron en la Tarea 4 se realicen correctamente.

**Tarea 11. Analizar los resultados de verificación y validación.** En esta tarea se analizó los resultados entregados y especificados en las plantillas a lo largo del procedimiento como informe de no conformidades, peticiones de cambios de los métodos, criterios y del entorno de verificación, informes de eficiencia de validación.

## 5.8. Recolección de datos

Para la recolección de la información de las tareas que lo requerían se diseñó unas plantillas (Anexo B, C, D, E), las cuales corresponden a las actividades de suministro de información o ejecución de las pruebas.

<b>Tareas claves</b>	<b>Listado de plantillas</b>
Identificar los métodos a ser sometidos a pruebas	Plantilla del listado de métodos a ser sometidos a pruebas (Anexo B)
Seleccionar los productos y métodos de verificación y validación	Plantilla de los productos de trabajo seleccionados (Anexo C)
Preparar las revisiones entre pares	Plantilla del personal involucrado en la revisión (Anexo D)
Realizar la revisión entre pares	Plantilla de informe de no conformidades encontrados (Anexo E)

**Tabla 5.6.** Listado de plantillas por tareas clave.

## 5.9. Resultados y análisis

En este punto se evidencia los resultados y análisis de la información recolectada en el estudio de caso y se muestra como influyó en la afinación del procedimiento para la automatización de pruebas unitarias en pequeñas empresas desarrolladoras de software. A partir de la encuesta de evaluación, las plantillas diligenciadas y las reuniones realizadas se muestra de manera sintetizada cuáles fueron los aportes.

### **Aspectos positivos**

- Se realizó la ejecución del procedimiento y se encontraron errores que no habían sido encontrados en las pruebas manuales.
- El procedimiento les ayuda a tener una organización a la hora de verificar que funcionalidades han sido probadas, cuales faltan por probar y en cuales hay fallos y que tipo de fallos.
- Las prácticas y las tareas del procedimiento se consideran adecuadas a la hora de soportar la realización de pruebas automáticas.

- Se muestra una descripción clara de los diagramas y plantillas representados en el procedimiento.
- El procedimiento planteado sirve para llevar una documentación de las pruebas realizadas.

### Aspectos a mejorar

- Evidenciar mejor los criterios de entrada y de salida de las pruebas en la plantilla 3 (Personal involucrado).
- Cuáles serían los aspectos mínimos para saber si una prueba cumple con los requisitos propuestos en la plantilla 3 (Personal involucrado).

### Observaciones

- Se sugiere validar comprobar los resultados esperados y no confiar únicamente en el código de respuesta de http/https arrojados por las pruebas automatizadas.
- Considerar entregar un script de pruebas realizadas.
- Se sugiere tener una lista de herramientas para ejecutar pruebas automáticas.

#### 5.9.1. Modificaciones realizadas al procedimiento después de realizado el estudio de caso

Componente	Modificación
Plantilla 3 (Personal involucrado)	Se modificó la descripción de los criterios de entrada y salida.
	Se adicionó el ítem pruebas en donde se incluirá el script de pruebas ejecutadas.

**Tabla 5.7.** Modificaciones realizadas con el estudio de caso.

## **5.9.2. Sumario**

El proceso de evaluación del procedimiento apoyado en la técnica de revisión por expertos ayudó en la afinación del procedimiento mediante la evaluación de cada uno de los participantes a las prácticas y tareas del procedimiento de modo que se tuviera una propuesta mejorada sobre el procedimiento de automatización de pruebas unitarias en pequeñas organizaciones desarrolladoras de software.

Teniendo en cuenta las recomendaciones realizadas por los participantes en la revisión de expertos se realizó las modificaciones al procedimiento donde se incluyeron elementos faltantes en el modelo, se realizan mejoras en la descripción de las tareas y se refinaron las plantillas.

Se consideró necesario evaluar el procedimiento mediante un estudio de caso ya que esta técnica según Bocco M. et. al. (2014) es un método de investigación flexible que trata con las complejas y dinámicas características de los fenómenos del mundo real y añade conocimiento a lo ya existente, ofreciendo beneficios como obtención de información de manera rápida además de sugerencias y observaciones para mejorar el procedimiento con la ventaja de emplearlo en el ámbito empresarial.

Con las recomendaciones planteadas por la empresa Namtrik Development S.A.S. en el estudio de caso se incluyeron elementos faltantes en las plantillas del procedimiento.

## **Capítulo 6**

# **Conclusiones y trabajo futuro**

Este capítulo expone las conclusiones de la presente investigación, así como también los trabajos futuros.

### **6.1. Sumario**

En este trabajo se presenta un procedimiento para la automatización de pruebas unitarias en pequeñas empresas desarrolladoras de software el cual abarca prácticas clave como: identificación y selección, establecimiento de criterios, procedimientos y entornos, preparación, realización y análisis de la revisión entre pares, realización de la verificación y validación y análisis de resultados. El punto de partida para el manejo de pruebas unitarias fue CMMI-DEV 1.3 (2010) en cuanto a verificación y validación. Se realizó una caracterización del manejo de pruebas unitarias en empresas de la región donde se encontraron falencias en el manejo de las pruebas y la implementación de las mismas, lo cual sirvió de punto de partida para la creación de la propuesta. Se implementó la propuesta en una pequeña empresa desarrolladora de software de la ciudad de Popayán para evaluar las prácticas, tareas y documentos necesarios para la automatización de pruebas unitarias.

### **6.2. Conclusiones**

A continuación, se presentarán las conclusiones obtenidas a partir de la revisión sistemática, los hallazgos en la caracterización y la evaluación del procedimiento, además se presenta el trabajo futuro y recomendaciones hechas al procedimiento.

- La técnica de revisión sistemática de la literatura fue muy útil para establecer el estado global del conocimiento en el área de automatización de pruebas en pequeñas empresas, además nos permitió valorar los resultados de los estudios primarios. Cabe resaltar que en los estudios primarios no se evidenció información referente a un procedimiento o marco de trabajo que diera a conocer todo lo concerniente a la automatización de pruebas unitarias de software.
- El 83.3% de las empresas de la región que participaron en la caracterización mostraron que realizan las pruebas a sus productos software de manera manual por lo que resultó viable realizar el procedimiento para la automatización de este tipo de pruebas.
- Según las empresas participantes en la caracterización, la reusabilidad es uno de los beneficios más importantes a la hora de realizar pruebas automatizadas dado que una prueba bien definida con el tiempo se hace reusable para otros productos software.
- El 66.7% de las empresas tienen falencias a la hora de manejar pruebas unitarias dado que no conocen este tipo de pruebas o simplemente no tienen la información necesaria para realizarlas, además, estas empresas no tienen el suficiente manejo de herramientas para automatizar, por lo que el procedimiento es adecuado dado que se guía a la empresa a la realización de pruebas unitarias, así como también la documentación de las pruebas.
- El método de evaluación del estudio de caso realizado al procedimiento de automatización de pruebas unitarias fue de gran utilidad dado que se realizó en una empresa en donde se facilitó un producto que ya estaba en producción y que cuenta con 146 usuarios, por lo que se resultó viable implementar el procedimiento a esta empresa.
- Dado que la mayoría de empresas no tienen la suficiente información referente a la automatización de pruebas unitarias se aportó en la línea de investigación de ingeniería del software implementando un procedimiento que refleje todo lo relacionado con el manejo de pruebas unitarias, este procedimiento incluye un modelo con prácticas y tareas que se deben hacer de forma consecutiva para llevar

a cabo una buena práctica de manejo de pruebas, así como también la documentación de las mismas para llevar un orden en el procedimiento.

- El proceso de evaluación apoyado en la técnica de revisión por expertos y el estudio de caso son pertinentes para la valoración de propuestas como este procedimiento debido a que ofrece la flexibilidad para explorar aspectos que no estuvieron planteados previamente, además permite recolectar diversos aportes que presenta cada uno de los participantes mediante el uso de medios de captura de información como encuestas o archivos de audio de manera que fueran la herramienta principal para conocer los aspectos positivos, a mejorar y observaciones del procedimiento propuesto.
- Se puede concluir que el procedimiento fue pertinente dado que ayuda a las empresas a tener un control sobre las pruebas realizadas además de incentivar a las empresas a realizar la automatización de pruebas unitarias de software.

### **6.3. Trabajo futuro y recomendaciones**

- Como trabajo futuro se propone llevar este procedimiento a un marco de trabajo para la automatización de pruebas unitarias en organizaciones desarrolladoras de software donde se tenga mejores herramientas para orientar la automatización de pruebas unitarias.
- Se recomienda realizar el seguimiento de las prácticas y tareas formuladas mediante la creación de métricas para medir la efectividad del procedimiento en empresas desarrolladoras de software, así como también la revisión completa de todos los documentos resultantes.
- Aplicar el procedimiento de automatización de pruebas en otras empresas con diferentes características para realimentarlo y comprobar su facilidad de ejecución.

## Referencias

- [1] Candea G., Bucur S., Zamfir C., "Automated Software testing as a Service", *SoCC '10 Proceedings of the 1<sup>st</sup> ACM Symposium on Cloud Computing*, pp. 155-160, 2010.
- [2] Moradbeiky A., Khatibi Bardsiri V., Aghajani S., Heydari R., "The Application of Automated Tools to Improve the Software Testing Process", *International Journal of Software Engineering and Its Applications*, vol. 8, no. 10, pp. 95-114, 2014.
- [3] Xiaochun Z., Bo Z., Juefeng L., Qiu G., "A Test Automation Solution on GUI Functional Test", *2008 6th IEEE International Conference on Industrial Informatics*, pp. 1413-1418, 2008.
- [4] SIUA - Servicio de Informática de la Universidad de Alicante, "Curso sobre Pruebas Unitarias en C#", Universidad de Alicante, [Online], Disponible: <https://goo.gl/ih2A9G>
- [5] Herrmann E., (2011, Nov, 14), "Pruebas de software: Herramientas: Pruebas Unitarias" [Online], Disponible: <https://goo.gl/qEodB8>
- [6] Safronau V., Turlo V., "Dealing with Challenges of Automating Test Execution: Architecture proposal for Automated Testing Control System Based on Integration of Testing Tools", *The third International Conference on Advances in System Testing and Validation Lifecycle*, 2011.
- [7] Bocco M., Lemus J, Piattini M., "Métodos de investigación en ingeniería del software", *RA-MA Editorial*, Madrid, 2014, pp. 1-287.
- [8] Böhme M., Paul S., "A probabilistic analysis of the efficiency of automated software test", *IEEE Transactions on Software Engineering*, vol. 42, no. 4, pp. 345-360, 2016.
- [9] Joy J., Pratap Singh D., "A Generic Framework Design to Enhance Capabilities of an Enterprise Test Automation Framework", *International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pp. 207-212, 2015.

- [10] Sávio de Oliveira G., Duarte A. “A Framework for Automated Software Testing on the Cloud”, *Proceedings of 14th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'13)*, 2013.
- [11] Figueiredo Collins E., Ferreira de Lucena Jr V., “Software Test Automation Practices in Agile Development Environment: An Industry Experience Report”, *7th International Workshop on Automation of Software Test, AST 2012 – Proceedings*, pp. 57-63, 2012.
- [12] Ha Kim E., Chae Na J., Moon Ryoo S., “Implementing an Effective Test Automation Framework”, *33<sup>rd</sup> Annual IEEE International Computer Software and Applications Conference*, vol. 2, pp. 534-538, 2009.
- [13] Bach James, “Test Automation Snake Oil”, *14th International Conference and Exposition on Testing Computer Software*, vol. 2.1, 1999.
- [14] Equipo del producto CMMI, “Mejora de los procesos para el desarrollo de mejores productos y servicios”, *CMMI para Desarrollo (CMMI-DEV)*, versión 1.3. Carnegie Mellon University, 2010.
- [15] Michalus J., Sarache W., Hernández G., “Método de expertos para la evaluación Ex-Ante de una solución organizativa”, *Revista científica Visión del Futuro*, vol. 19, no. 1, pp. 1-17, 2015.
- [16] Brereton P., Kitchenham B., Budgen D., Li Z., “Using a Protocol Template for Cause Study Planning”, In *EASE*, vol. 8, pp. 41-48, 2008.
- [17] La oficina de proyectos de informática, “Pruebas de software”, *PMOinformatica*, [Online], Disponible: <https://bit.ly/2yqVdjp>
- [18] CHOUCAIR Business Centric Testing, “Probador Certificado – Nivel Básico”, Formación para el “Probador Certificado – Nivel Básico” de acuerdo al programa de estudios 2010 del ISTQB, versión 1.2a\_V ES 1.0. Díaz & Hilterscheid Unternehmensberatung GmbH, 2010.
- [19] Cillero M. “Pruebas de integración”, *mi circunstancia digital*, [Online], Disponible: <https://bit.ly/2D3QPuu>