



# **Universidad Nacional de Costa Rica**

**Sistema de Estudios de Posgrado**

**Maestría en Administración de Tecnologías de la Información (MATI)**

**Énfasis en Administración de Proyectos**

**Optimización de Procesos de Control de Calidad (QA) en Empresa de**

**Desarrollo de Software con Equipos Reducidos**

**Estudiante:**

**Carlos Andrés Obando Jiménez**

**Heredia, Costa Rica Febrero, 2024**

# DECLARACIÓN JURADA DE RESPETO AL DERECHO DE AUTOR

5 de diciembre del año 2024

Universidad Nacional  
Facultad de Ciencias Exactas y Naturales  
Escuela de Informática  
Posgrado en Gestión de la Tecnología de Información y Comunicación (ProGesTIC)

<p><b>FORMULARIO DE DEPÓSITO LEGAL, AUTORIZACIÓN DE USO DE DERECHOS PATRIMONIALES DE AUTOR E INCORPORACIÓN A REPOSITARIOS INSTITUCIONALES DE INFORMACIÓN DE ACCESO PÚBLICO</b></p>
--

La persona abajo firmante, en condición de estudiantado de la Maestría en Tecnologías de la Información (MATI) con Énfasis en Administración de Proyectos y autor del Trabajo final de graduación titulado:

**Optimización de Procesos de Control de Calidad (QA) en Empresa de Desarrollo de Software con Equipos Reducidos**

Para optar al grado académico de Máster en:

**Tecnologías de la Información con Énfasis en Administración de Proyectos**


De conformidad con lo establecido en el documento de generales para la realización del trabajo final de relacionada con estos trabajos de graduación, DECLARO BAJO FE DE JURAMENTO conociendo la responsabilidad civil, penal o administrativa en que podría incurrir al no decir la verdad, lo siguiente:

disposición de público la obra en mención, a través de los espacios físicos o virtuales que se posea, así como a través del Repositorio Institucional; a partir del cual los usuarios de dichas plataformas puedan acceder al documento y hacer uso de este en el marco de los fines académicos, no lucrativos y de respeto a la integridad del contenido del mismo, así como la mención del autor poseedor de sus derechos.

7. Manifiesto que todos los datos de citas dentro de texto y sus respectivas referencias bibliográficas, así como las tablas y figuras (ilustraciones, fotografías, dibujos, mapas, esquemas u otros) tienen la fuente y el crédito debidamente identificados y se han respetado los derechos de autor.
8. Autorizo la licencia gratuita no exclusiva de los derechos patrimoniales de autor para reproducir, traducir, distribuir y poner a disposición pública en formato electrónico, el documento depositado, para fines académicos, no lucrativos y por plazo indefinido en favor de la Universidad Nacional, que incluye además los siguientes actos:
  - a. La publicación y reproducción íntegra de la obra o parte de esta, tanto por medios impresos como electrónicos, incluyendo Internet y cualquier otra tecnología conocida o por conocer.
  - b. La traducción a cualquier idioma o dialecto de la obra o parte de esta.
  - c. La adaptación de la obra a formatos de lectura, sonido, voz y cualquier otra representación o mecanismo técnico disponible, que posibilite su acceso para personas no videntes parcial o totalmente, o con alguna otra forma de capacidades especiales que le impida su acceso a la lectura convencional del TFG.
  - d. La distribución y puesta a disposición de la obra al público, de tal forma que el público pueda tener acceso a ella desde el momento y lugar que cada quien elija, a través de los mecanismos físicos o electrónicos de que disponga.
  - e. Cualquier otra forma de utilización, proceso o sistema conocido o por conocerse que se relacione con las actividades y fines académicos a los cuales se vincula la maestría, la colección de trabajos finales del ProGesTIC, la Escuela de Informática y la Universidad Nacional.
9. Reconozco que la colección de trabajos del ProGesTIC manifiesta actuar con diligencia para evitar la existencia en su sitio web de contenidos ilícitos y en caso de que tenga conocimiento efectivo de la existencia de infracciones a los derechos de propiedad intelectual, se reserva el derecho de proceder a bloquear el acceso durante el trámite del debido proceso para comprobar el incumplimiento y en caso de

verificarse la falta. retirarse definitivamente el acceso al TFG depositado.

10. Acepto que la publicación y puesta a disposición del público del trabajo final de graduación, así como la presente autorización de uso de la obra, se regirá por la normativa institucional de la Universidad Nacional y la legislación de la República de Costa Rica. Adicionalmente, en caso de cualquier eventual diferencia de criterio o disputa futura, acepto que esta se dirimirá de acuerdo con los mecanismos de Resolución Alternativa de Conflictos y la Jurisdicción Costarricense.

Autor:	Carlos Andrés Obando Jiménez
Fecha de entrega:	05/12/2024
Correo:	carlitos181220@gmail.com
Firma:	

## **Dedicatoria**

A Yerling Morales López, mi amada esposa, por ser mi compañera incondicional y una fuente inagotable de motivación en cada etapa de este largo y desafiante camino. Tu amor, comprensión y paciencia han sido fundamentales para superar las dificultades y mantenerme enfocado en la meta. Este logro no habría sido posible sin tu apoyo constante y tus palabras de aliento.

A Katia Lorena Jiménez Matarrita, mi madre, por inculcarme desde niño los valores de esfuerzo, perseverancia y dedicación. Tu ejemplo de vida y tus sabios consejos han sido una guía en mi trayectoria académica y personal. Siempre encontré en tus palabras la fortaleza necesaria para continuar, incluso en los momentos más complicados.

A Yeimy Rodríguez Alvarado, mi filóloga, por tu valiosa dedicación, profesionalismo y guía a lo largo del desarrollo de este trabajo final de graduación. Tu apoyo fue determinante para la estructuración del contenido, la formación de las ideas y la claridad en la expresión de cada uno de los conceptos que conforman esta investigación. Tu labor enriqueció significativamente este proyecto, y por ello te estaré siempre agradecido.

A todas ustedes, les dedico este logro con el más profundo aprecio y gratitud.

**Carlos Andrés Obando Jiménez**

## **Agradecimientos**

Agradezco profundamente al profesor Fulvio Lizano Madriz, quien fue el profesor guía durante los cursos Proyecto Final de Graduación I y II, por su invaluable orientación, experiencia, paciencia y prontitud en las respuestas durante el desarrollo del trabajo final de graduación. Su apoyo constante y su aporte intelectual fueron fundamentales para alcanzar este proyecto.

Extiendo mi más sincera gratitud a los profesionales encargados de las evaluaciones pares:

- M.Sc. Lawrence Fowks Peña,
- M.Sc. José Miguel Chacón Araya, y
- M.Sc. Alex Daniel Villegas Carranza,

por sus aportes, comentarios, recomendaciones y por compartir su conocimiento técnico-profesional, lo cual enriqueció significativamente este trabajo final de graduación. Su contribución fue de vital importancia para aumentar la calidad y el rigor de esta investigación.

A todos ustedes, gracias por formar parte de este importante logro académico.

**Carlos Andrés Obando Jiménez**

# Índice General

<b>DECLARACIÓN JURADA DE RESPETO AL DERECHO DE AUTOR .....</b>	<b>2</b>
<b>Dedicatoria .....</b>	<b>5</b>
<b>Agradecimientos .....</b>	<b>6</b>
<b>Índice General.....</b>	<b>7</b>
<b>Índice de Figuras .....</b>	<b>13</b>
<b>Índice de Tablas.....</b>	<b>13</b>
<b>Resumen ejecutivo .....</b>	<b>14</b>
<b>Capítulo I. El problema y su importancia.....</b>	<b>15</b>
1.1 Título del tema de investigación .....	16
1.2 Antecedentes .....	16
1.3 Justificación.....	18
1.4 Problema .....	18
1.5 Objetivo general .....	19
1.6 Objetivos específicos .....	19
1.7 Metas por alcanzar por objetivo.....	20
<b>Capítulo II. Marco teórico o referencial.....</b>	<b>21</b>
2.1 Historia y evolución del control de calidad en el desarrollo de software .....	22
2.2 Procesos de control de calidad en empresas de desarrollo de software .....	24
2.2.1. La revisión y la validación de código.....	25
2.2.2. Las pruebas de software .....	25
2.2.3. La configuración del software .....	26

2.2.4.	La automatización de pruebas .....	26
2.2.5.	Integración continua .....	28
2.2.6.	Revisión y mejora continua .....	29
2.3	Tipos de procesos de calidad en el desarrollo de software .....	29
2.4	Organización de la calidad en el desarrollo de software .....	30
2.4.1.	Estándares de calidad .....	32
2.4.2.	Distribución de Roles dentro de los Equipos de Calidad de Software .....	32
2.4.3.	Estrategias de Coordinación y comunicación en los Equipos de Calidad de software .....	33
2.5	Equipos de calidad de software reducidos .....	34
2.5.1.	Comparativa entre equipos grandes y pequeños de calidad de software .....	35
2.5.2.	Experiencias en la implementación de estrategias para optimizar la calidad de software .....	37
2.5.3.	Eficiencia en la solución y detección de defectos en los equipos de calidad de software .....	38
2.5.4.	Relevancia de la experiencia previa en la detección de errores .....	39
2.5.5.	Adaptabilidad hacia las nuevas herramientas y metodologías .....	40
2.6	Estrategias para optimizar la calidad en el desarrollo de software .....	41
2.7	Tipos de estrategias para optimizar la calidad en el desarrollo de software .....	42
	<b>Capítulo III. Marco metodológico.....</b>	<b>45</b>
3.1	Enfoque de la investigación .....	46

3.2	Tipo de investigación .....	47
3.2.1	Revisión de la literatura .....	47
3.2.2	Entrevistas .....	48
3.2.3	Caso de estudio.....	49
3.2.4	Evaluación de pares.....	50
3.3	Fuentes y sujetos de información.....	51
3.3.1.	Fuentes de información .....	51
3.3.2	Sujetos .....	52
3.4	Población y muestra .....	53
3.4.1.	La población .....	53
3.4.2.	Muestra.....	53
	<b>Capítulo IV. Diagnóstico y análisis de resultados.....</b>	<b>55</b>
4.1	Revisión de la literatura .....	56
4.2	Entrevistas.....	57
4.2.1	Datos demográficos.....	58
4.2.2	Análisis de datos.....	58
4.2.2.1	Eficiencia de los equipos reducidos de calidad de software .....	60
4.2.2.2	Comunicación entre equipos .....	61
4.2.2.3	Influencia del tamaño de los equipos en el cumplimiento de estándares de calidad..	
	.....	63

4.2.2.4	Influencia del tamaño de los equipos en el cumplimiento de estándares de calidad..	64
4.2.2.5	Importancia del personal capacitado para la detección de errores .....	66
4.2.2.6	Relevancia de experiencia previa para la detección de errores .....	68
4.2.2.7	El personal experimentado contribuye a la mejora de procesos de prueba.....	69
4.2.2.8	Dependencia en la adaptación de nuevas herramientas y tecnologías .....	71
4.2.2.9	Asignación de roles y su efectividad para la optimización de procesos de control de calidad .....	72
4.2.2.10	Automatización de pruebas como compensación de recurso humano .....	74
4.2.2.11	Satisfacción en los procesos de control de calidad en su equipo de trabajo reducido .....	75
4.3	Casos de estudio.....	76
4.3.1.	Contexto general del caso de estudio: Descripción de la empresa.....	76
4.3.2.	Objetivo del caso de estudio: Justificación de la elección de la empresa .....	76
4.3.3.	Descripción general de Golabs Tech.....	77
4.3.4.	Equipos de QA: Roles, tamaños y responsabilidades .....	78
4.3.5.	Procesos de QA: Metodologías y herramientas .....	79
4.3.6.	Desafíos previos de la compañía .....	79
4.3.7.	Revisión de la documentación.....	80
4.3.8.	Flujo de trabajo y actividades diarias .....	80

4.3.9.	Planes de capacitación.....	81
4.4	Evaluaciones pares.....	82
<b>Capítulo V. Propuesta de Solución .....</b>		<b>83</b>
5.1	Propuesta de solución.....	84
<b>Capítulo VI. Análisis financiero .....</b>		<b>89</b>
6.1	Análisis financiero .....	90
6.2	Costos de implementación .....	90
6.2.1	Recursos Humanos.....	90
6.2.2	Herramientas tecnológicas .....	91
6.2.3	Costos fijos.....	92
6.2.4	Capacitaciones y automatizaciones .....	92
6.3	Ingresos esperados respecto a la implementación de la propuesta .....	93
6.3.1	Reducción de costos por detección temprana de errores.....	93
6.3.2	Retención de clientes.....	94
6.3.3	Reducción de tiempos de entrega.....	94
6.4	Flujo de caja .....	95
6.4.1.	Resultados del Valor Actual Neto (VAN) y Tasa Interna de Retorno (TIR) .....	96
6.4.2.	Recuperación de la Inversión .....	96
<b>Capítulo VII. Conclusiones y recomendaciones .....</b>		<b>98</b>
7.1	Conclusiones .....	99
7.1.1	Conclusión general.....	99

7.1.2	Conclusiones específicas .....	99
7.2	Recomendaciones.....	102
<b>Capítulo VIII. Análisis retrospectivo.....</b>		<b>104</b>
8.1	Análisis retrospectivo.....	105
<b>Referencias Bibliográficas .....</b>		<b>106</b>
<b>Apéndice .....</b>		<b>116</b>
Apéndice 1: Entrevista.....		117
Apéndice 2: Caso de estudio .....		123
Apéndice 3: Implementación de propuesta .....		125
Apéndice 4: Revisión de José Miguel Chacón .....		119
Apéndice 5: Revisión de Lawrence Fowks Peña.....		122
Apéndice 5: Revisión de Alex Daniel Villegas Carranza.....		126
<b>Anexos.....</b>		<b>130</b>
Anexo 1: Respuestas del cuestionario .....		131
Anexo 2: Career Path.....		134

## Índice de Figuras

Figura 1 Gráfico de distribución de respuestas en Escala de Likert.....	60
Figura 2 Cronogramas de actividades .....	13

## Índice de Tablas

Tabla 1 Metas por objetivos de investigación .....	20
Tabla 2 Datos de entrevista .....	59
Tabla 3 Costos de recurso humano del departamento de QA .....	91
Tabla 4 Costos de herramientas empleadas a por el departamento de QA.....	92
Tabla 5 Costos de capacitaciones .....	93
Tabla 6 Detalles del flujo de caja .....	96
Tabla 7 Resultados del Valor Actual Neto (VAN) y Tasa Interna de Retorno (TIR) .....	96
Tabla 8 Recuperación de la Inversión .....	97

## Resumen ejecutivo

En el competitivo y dinámico ámbito del desarrollo de software, garantizar la calidad del producto es clave para el éxito empresarial. Sin embargo, las empresas con equipos de aseguramiento de calidad (QA) reducidos enfrentan desafíos significativos debido a sus recursos limitados. Este TFG se centra en el desarrollo de una propuesta estratégica para optimizar los procesos de control de calidad (QA) en empresas de desarrollo de software que operan con equipos reducidos. La estrategia se fundamenta en un análisis de casos de éxito y una revisión sistemática de la literatura reciente, lo que permitió desarrollar un marco teórico sólido sobre calidad y desarrollo de software en equipos reducidos. Además, se realizó un diagnóstico situacional para comprender cómo se gestionan estos equipos, identificando retos específicos y oportunidades clave.

Con base en estos hallazgos, se diseñó una propuesta estratégica adaptada a las características y limitaciones de los equipos reducidos. La estrategia aborda elementos como la priorización de tareas, la automatización de procesos y las capacitaciones cruzadas, garantizando un enfoque práctico y escalable.

Finalmente, la propuesta fue validada por pares expertos en QA, quienes contribuyeron a su mejora y aseguraron su relevancia en entornos reales. Este proceso refuerza la aplicabilidad y eficacia de la estrategia en el sector.

El resultado es una contribución significativa al conocimiento y las prácticas sobre la optimización de procesos de calidad en empresas de desarrollo de software, ofreciendo soluciones alineadas con las necesidades de equipos de QA reducidos.

**Palabras claves:** Equipos reducidos, Propuesta estratégica, Estándares de calidad, Optimización de QA.

## **Capítulo I. El problema y su importancia**

En este capítulo, se expone el tema principal del trabajo final de grado, además de aspectos como los antecedentes, la justificación, el problema y los objetivos tanto generales como específicos, con el fin de abordar tanto el problema como su importancia.

### **1.1 Título del tema de investigación**

Optimización de los procesos de control de calidad (QA) en empresa de desarrollo de software con equipos reducidos.

### **1.2 Antecedentes**

En los últimos años, el desarrollo ágil de software ha ganado gran relevancia, destacando la importancia de la implementación de la calidad para asegurar una entrega de productos confiables y funcionales. Según Conboy et al. (2020), el enfoque ágil promueve ciclos de desarrollo de software tanto iterativos como colaborativos que resultan en procesos de QA flexibles y eficientes; sin embargo, estos beneficios no se han generalizado en todos los tipos de empresas, especialmente en aquellas con equipos reducidos.

Grant (2023) afirma que los equipos reducidos en la calidad de software surgen principalmente de la necesidad de las compañías de maximizar resultados con recursos limitados. En el contexto actual donde los desarrolladores de software están cada vez más involucrados en procesos de calidad, estos equipos pequeños deben asumir mayores responsabilidades que abarcan desde la revisión del código hasta la automatización de pruebas, manteniendo altos estándares de calidad mientras se acortan los tiempos de desarrollo y se optimizan los recursos, incorporando el proceso de calidad desde las fases iniciales del desarrollo de software.

Por otra parte, cuando se hace referencia a equipos de trabajo reducidos, estos se enfrentan a retos específicos al implementar procesos de calidad de software, en gran medida debido a problemas relacionados con los recursos técnicos disponibles. En la actualidad, numerosos estudios han resaltado que una correcta implementación de estrategias como la automatización de pruebas y herramientas de integración continua puede asegurar estándares de calidad muy altos, incluso cuando se cuenta con personal limitado (Rahman et al., 2019). En este sentido, uno de los factores más relevantes a mencionar es que, en empresas con recursos limitados, la automatización de pruebas se ha consolidado como una práctica vital para este tipo de equipos reducidos, mejorando significativamente la eficiencia y disminuyendo el riesgo de errores humanos, lo que ha permitido que los desarrolladores de software puedan enfocarse en procesos de mayor complejidad (Erich et al., 2017).

Un aspecto determinante en el proceso de desarrollo de software es la integración y el despliegue continuo de código, ya que estas prácticas han marcado un antes y un después en la manera en que los equipos de desarrollo gestionan el control de calidad, permitiendo la detección temprana de errores y facilitando una retroalimentación rápida, lo cual es un aspecto determinante para mantener la calidad en ciclos de desarrollo cortos y frecuentes (Shahin et al., 2017).

Finalmente, la colaboración efectiva entre los equipos de desarrollo y los equipos de QA es fundamental para asegurar el éxito del control de la calidad, especialmente en entornos de equipos reducidos. Pascarella et al. (2019) subrayan que la comunicación y la comprensión mutua de los objetivos y desafíos son esenciales, ya que contribuyen significativamente a mejorar tanto la eficiencia como la efectividad del proceso de control de calidad de software.

### **1.3 Justificación**

En la actualidad, en el ámbito del desarrollo de software, es de vital importancia que las empresas garanticen que los productos que desarrollan mantengan altos estándares de calidad para lograr el éxito en el mercado; para ello, estas cuentan con equipos encargados de esta tarea, los cuales se denominan equipos de Quality Assurance (QA por sus siglas en inglés), y a menudo están integrados por un bajo número de integrantes, situación frecuentemente atribuida a limitaciones de recursos.

Este proyecto tiene como objetivo identificar estrategias eficaces que permitan que estos equipos reducidos logren trabajar de manera óptima; con ello, se busca colaborar con empresas del sector del desarrollo de software para enfrentar los desafíos que plantea un entorno con recursos limitados, asegurando al mismo tiempo la calidad de los productos desarrollados.

### **1.4 Problema**

El problema principal que impulsa esta investigación tiene relación con la dificultad que enfrentan las empresas de desarrollo de software con equipos limitados en el área de aseguramiento de la calidad de sus productos. En un mercado altamente competitivo, donde la exigencia en los estándares de calidad es muy alta, estas compañías deben de cumplir con dichas expectativas a pesar de contar con equipos reducidos destinados para esta área.

Debido a esta problemática surge la pregunta principal de esta investigación: ¿cómo se pueden optimizar los procesos de control de calidad para maximizar la eficiencia y efectividad de estos equipos reducidos, garantizando al mismo tiempo la calidad del software desarrollado? Como se observa, la investigación busca identificar métodos y estrategias que permitan a estos equipos mejorar sus procesos.

## **1.5 Objetivo general**

Desarrollar una propuesta de estrategia para optimizar procesos de control de calidad (QA) en empresa de desarrollo de software con equipos reducidos, mediante un proceso de análisis y evaluación de casos de éxito previos.

## **1.6 Objetivos específicos**

- a) Revisar los referentes teóricos para brindar un sustento referencial a la investigación a través de una verificación sistemática de literatura.
- b) Revisar la gestión de los equipos de calidad en entornos reducidos de desarrollo de software, por medio de un diagnóstico situacional.
- c) Desarrollar una propuesta de estrategia para optimizar los procesos de control de calidad en equipos reducidos, tomando en cuenta los antecedentes teóricos y la situación de contexto explorada.
- d) Evaluar la estrategia propuesta por medio de una evaluación de pares para depurarla.

## 1.7 Metas por alcanzar por objetivo

**Tabla 1**

*Metas por objetivos de investigación*

<b>Objetivo</b>	<b>Meta</b>
Revisar los referentes teóricos para brindar un sustento referencial a la investigación a través de una verificación sistemática de literatura.	Obtener una base sólida de información sobre los equipos reducidos en la calidad del software a través de una revisión exhaustiva y sistemática de la literatura actual.
Revisar la gestión de los equipos de calidad en entornos reducidos de desarrollo de software, por medio de un diagnóstico situacional.	Elaborar un diagnóstico detallado que describa el funcionamiento de estos equipos de calidad de software en entornos reducidos, basado en una revisión extensiva de casos actuales.
Desarrollar una propuesta de estrategia para optimizar los procesos de control de calidad en equipos reducidos, tomando en cuenta los antecedentes teóricos y la situación de contexto explorada.	Desarrollar una propuesta de estrategia con la finalidad de optimizar procesos de control de la calidad en equipos reducidos.
Evaluar la estrategia propuesta por medio de una evaluación de pares para depurarla.	Refinar la estrategia propuesta, basados en la retroalimentación de profesionales del área, garantizando su eficacia y aplicabilidad en contextos reales.

## **Capítulo II. Marco teórico o referencial**

En este capítulo se muestran los fundamentos teóricos que han permitido elaborar la estructura en la que se evidencian los conceptos relacionados con el tema de investigación, los cuales estarán divididos en siete subapartados, que se mencionan a continuación: Historia y evolución del control de calidad en el desarrollo de software; procesos de control de calidad en empresas de desarrollo de software; tipos de procesos de calidad en el desarrollo de software; organización de la calidad en el desarrollo de software; equipos de calidad de software reducidos; historia y evolución de las estrategias para optimizar la calidad en el desarrollo de software; y tipos de estrategias para optimizar la calidad en el desarrollo de software.

## **2.1 Historia y evolución del control de calidad en el desarrollo de software**

La calidad de software ha experimentado transformaciones significativas a lo largo de su evolución, buscando la manera de adaptarse continuamente a las tecnologías y las metodologías empleadas en la industria del software. Su evolución se ha dividido en varias fases clave que han modelado las prácticas actuales. A continuación, se presentan dichas fases:

- a) **Primera fase (1950-1970):** Esta etapa marca el inicio de las metodologías de calidad en el desarrollo de software, originadas a partir de la aplicación de técnicas estadísticas en industrias como la manufacturera y adaptadas al desarrollo de software. En los años de 1950, W. Edwards Deming y Joseph M. Juran lograron introducir métodos para el control de la calidad, los cuales impactaron significativamente en el desarrollo de software, subrayando la importancia de la

planificación y la evaluación continua de procesos (American Society for Quality, 2021).

- b) **Segunda fase (1980 y 1990):** En esta fase surgieron nuevas metodologías específicas para el desarrollo de software, integrando el control de calidad como un componente central. Un claro ejemplo es el Modelo de Capacidad de Madurez (CMM) y el Proceso Unificado de Rational (RUP), que tienen como objetivo estandarizar los procesos de desarrollo y asegurar la calidad a través de revisiones y pruebas sistemáticas, lo que contribuyó a establecer procedimientos más uniformes y eficientes en la industria (Stumbles, 2021).
- c) **La tercera fase (años 2000):** Esta etapa se caracteriza por la adopción de métodos ágiles que marcaron el inicio del siglo XXI. Metodologías como Extreme Programming (XP) y SCRUM se popularizaron ampliamente, promoviendo ciclos de desarrollo cortos e iterativos, dando como resultado una integración continua de pruebas y retroalimentación, lo que mejoró la calidad del software de manera dinámica y adaptativa (Stumbles, 2021).
- d) **Cuarta etapa (década de 2010 en adelante):** Según Do & Reich (2023), en la última década, el enfoque ha evolucionado hacia prácticas como DevOps, que integran el desarrollo y las operaciones con el objetivo de mejorar la eficiencia y la calidad del software, cuya principal función es mantener un ciclo de desarrollo continuo, acompañado de pruebas automatizadas, implementación continua y monitoreo constante, lo que asegura la calidad en todas las etapas del ciclo de vida del software.

Actualmente, la calidad en las organizaciones supone conservar procesos de control de calidad para su implementación.

## **2.2 Procesos de control de calidad en empresas de desarrollo de software**

En la actual industria del desarrollo de software, existen diversos procesos diseñados para asegurar que los productos entregados sean eficientes, robustos y cumplan con las especificaciones de los usuarios; sin embargo, antes de adentrarse en ello, es necesario conocer lo que se entiende por calidad en sí. Según Arriagada et al. (2023), "la calidad es un concepto complejo y multifacético" (p.6), lo que implica que su definición y aplicación varían dependiendo del contexto y los aspectos que prioricen.

El concepto de calidad en software es intrínsecamente dinámico y evolutivo, por lo que siempre se encuentra en constante evolución, conforme a que la tecnología y las metodologías avanzan. Según la Organización Internacional de Normalización (ISO, 2015), la calidad se define como el nivel en el cual un conjunto de atributos propios de un producto, servicio o proceso logra satisfacer las expectativas del cliente o de las partes involucradas. Esta definición resalta la naturaleza flexible de las mismas, las cuales no solo dependen de los requerimientos técnicos de software, sino también de las expectativas y necesidades cambiantes del usuario.

De acuerdo con Pacheco (2024), en el ámbito de la ingeniería de software, los procesos de calidad se estructuran en torno a una serie de metodologías, estándares y prácticas diseñadas para identificar, corregir y prevenir defectos a lo largo del ciclo de vida del producto, con el fin de garantizar que todos los servicios cumplan con los estándares y necesidades establecidos por el cliente.

A continuación, se destacan seis de los procesos más relevantes en este ámbito:

### **2.2.1. La revisión y la validación de código**

Conocida también como *peer review*, este proceso implica que varios desarrolladores revisen el código fuente para identificar posibles errores y áreas de mejora. Dependiendo de las condiciones, este proceso podría ser formal o informal, y muy frecuentemente se utilizan técnicas como la programación en pares (Cialis et al., 2017).

El objetivo de la revisión de código es garantizar que el software desarrollado sea de alta calidad y libre de errores, mejorando las buenas prácticas de codificación dentro del equipo, fomentando la adherencia a los estándares de codificación, la optimización del rendimiento y la mejora de la legibilidad del código. De acuerdo con Hijazi et al. (2022), este proceso no solamente se basa en la detección de errores funcionales, sino que también sirve para identificar áreas de mejoras en los equipos de desarrollo.

### **2.2.2. Las pruebas de software**

Las pruebas de software son un componente fundamental en el desarrollo, ya que permiten verificar la funcionalidad y la calidad del producto final. Bajo esta línea, Cruz (2024) menciona que el propósito de estas es que le permiten a los desarrolladores la detección de defectos y asegura que el sistema cumpla con los requisitos establecidos.

Actualmente, existen diversos tipos de pruebas, cada una con un enfoque específico y complementario a las demás, tales como:

- a) Las pruebas unitarias: Encargadas de evaluar componentes individualmente del software para lograr determinar su correcto funcionamiento.
- b) Las pruebas de integración: Consiste en verificar la correcta integración entre diferentes módulos o componentes del sistema.

- c) Las pruebas de sistema: Estas se enfocan en evaluar el sistema completo en su entorno operativo.
- d) Las pruebas de aceptación: Son las realizadas por el usuario final para asegurar que el sistema cumple con sus expectativas (Nogueira et al., 2016).

### ***2.2.3. La configuración del software***

Dentro de las tecnologías de información, la configuración del software es un proceso determinante, ya que asegura que los atributos de un producto mantengan coherencia a lo largo de su ciclo de vida. Respecto al contexto tecnológico, se puede mencionar que este proceso está encargado de gestionar y monitorear cada elemento configurado de un sistema, como software o servidores, lo que a la postre genera una visión integral de los activos con que cuenta TI, desde componentes individuales hasta los sistemas más complejos (Buchanan, s.f.).

Este proceso implica el uso de sistemas de control de versiones para gestionar los cambios en el código fuente. Actualmente, herramientas como Git resultan indispensables, ya que permiten mantener la integridad y la coherencia del software a lo largo del ciclo de vida del software (Aleem et al., 2016).

### ***2.2.4. La automatización de pruebas***

La automatización de pruebas es un proceso esencial en el desarrollo de software, diseñado para ejecutar pruebas repetitivas de manera eficiente, mejorando la cobertura y reduciendo el esfuerzo manual involucrado. Según Paz (2024), su principal objetivo es optimizar la eficiencia, precisión y rapidez de las pruebas, lo que permite una validación más efectiva y

reduce significativamente los recursos y tiempo necesarios para comprobar la calidad del software.

Entre las herramientas más utilizadas para la automatización de pruebas se encuentran Selenium y JUnit, que son comúnmente implementadas para realizar automatizaciones de pruebas de interfaz de usuario y pruebas unitarias (Araújo et al., 2016).

#### **2.2.4.1. Automatización de pruebas como compensación de recurso humano**

La automatización de pruebas ha surgido como una solución efectiva para reducir la dependencia del personal humano en la ejecución de pruebas repetitivas y rutinarias, permitiendo que los equipos pongan sus esfuerzos en pruebas que demanden mayor análisis y ejecución (Paz, 2024). Al sustituir tareas manuales repetitivas, la automatización no solo mejora la eficiencia del proceso de pruebas, sino que permite optimizar los recursos humanos, asignando a las profesionales tareas más complejas y de mayor valor estratégico (Sindhu, 2019).

El principal beneficio de la automatización radica en su capacidad para ejecutar pruebas de manera continua y precisa, sin las limitaciones que impone el trabajo manual. Según Aleem et al. (2016), la automatización expande significativamente la cobertura de pruebas al ejecutar miles de escenarios de prueba en una fracción del tiempo, lo cual sería inviable manualmente. Esto asegura que el software sea probado en más combinaciones posibles, lo que facilita la identificación temprana de fallos en el ciclo de desarrollo y contribuye a una mayor calidad del producto final. IEEE (2019) también respalda estos beneficios, destacando que la automatización permite realizar pruebas repetitivas de manera sistemática, asegurando que se mantenga una cobertura exhaustiva en cada iteración de desarrollo.

Además, la implementación de la automatización de pruebas representa un ahorro considerable a largo plazo. Aunque la inversión inicial puede ser significativa, los beneficios en términos de reducción de tiempos y costos son evidentes: el esfuerzo inicial se compensa con una menor demanda de recursos humanos para tareas operativas y una aceleración en el tiempo de comercialización del software (Shin et al., 2011). La automatización no solo optimiza el uso del recurso humano, sino que también contribuye a un ciclo de desarrollo más ágil y a una mejora continua de la calidad del software, generando valor tanto para los equipos de desarrollo como para los usuarios finales.

#### **2.2.5. Integración continua**

La integración continua es una práctica esencial en el desarrollo ágil de software que permite a los equipos de desarrollo integrar y verificar regularmente sus cambios de código en un repositorio compartido. Como mencionan Aleem et al. (2016), es una práctica en donde los desarrolladores de software están integrando su trabajo de manera regular, mientras las pruebas automatizadas se ejecutan continuamente con la finalidad de poder encontrar fallas tempranas en el proceso de desarrollo. Por lo tanto, se afirma que "se centra en asegurar que el software se encuentre en un estado de implementación constante". (Cruz, 2024, p. 31), garantizando que el producto se mantenga en funcionalidad y libre de errores a lo largo del ciclo de vida del proyecto.

La integración continua emplea herramientas como Jenkins y Traian CL que facilitan este proceso, resultando en una notable mejora en la calidad del software que es entregado y una reducción en los tiempos de entrega.

### **2.2.6. *Revisión y mejora continua***

El proceso de revisión y mejora continua es fundamental para asegurar la evolución constante del software y la optimización de los procesos de desarrollo. Este enfoque se sustenta en las metodologías ágiles como SCRUM y Kanban, las cuales promueven la mejora mediante iteraciones cortas y retroalimentación constante, utilizando retrospectivas y reuniones diarias en los equipos de trabajo para identificar áreas de mejora e implementar cambios efectivos en el desarrollo del software (Nogueira et al., 2016).

Además, según Pacheco (2024), estas metodologías ofrecen un enfoque dinámico y flexible para la gestión de proyectos, destacando la importancia de entregar de manera continua valor al cliente, fomentar una colaboración eficaz dentro del equipo y tener la capacidad de responder rápidamente a los cambios que puedan surgir a lo largo del ciclo de vida del proyecto.

## **2.3 Tipos de procesos de calidad en el desarrollo de software**

En el desarrollo de software, los procesos de calidad son cruciales para asegurar la fiabilidad y eficacia del producto final, ya que, según García et al. (2020), se pueden identificar diversos enfoques clave en la gestión de la calidad del software. Uno de ellos son los procesos de Calidad Basada en Normativas, que se enfocan en cumplir con estándares y regulaciones específicas, como la norma ISO/IEC 25010:2011, sustituida por la serie de normas ISO/IEC 25000:2019, la cual proporciona directrices detalladas para evaluar la calidad del producto software en términos de características como la funcionalidad, fiabilidad y mantenibilidad (ISO, 2019).

Por otro lado, los procesos de calidad basados en modelos, como el Capability Maturity Model Integration (CMMI), facilitan la evaluación y el perfeccionamiento de la madurez de los procesos en el desarrollo de software (SEI, 2018), permitiendo a las organizaciones identificar áreas de mejora y establecer metas claras de calidad. Asimismo, los procesos de calidad ágiles, enmarcados en metodologías como Scrum y eXtreme Programming (XP), destacan la colaboración, la entrega iterativa y la adaptabilidad a los cambios en los requisitos del cliente (Beck et al., 2018), buscando mejorar la calidad del producto final, mediante la iteración continua y la retroalimentación rápida.

Finalmente, los procesos de calidad DevOps se basan en prácticas como la integración continua (CI) y la entrega continua (CD), que buscan automatizar y optimizar el ciclo de vida de desarrollo de software para mejorar la calidad a través de pruebas automatizadas y despliegues frecuentes (Humble & Farley, 2010).

Una vez abarcados los principales tipos de procesos de calidad en el desarrollo de software, es fundamental entender cómo las empresas organizan la calidad para garantizar que todos estos procesos sean implementados de una manera efectiva. A continuación, se abordarán las estrategias y enfoques clave, resaltando la importancia de los sistemas de gestión, las metodologías ágiles, la cultura organizacional y la capacitación continua.

## **2.4 Organización de la calidad en el desarrollo de software**

La organización de la calidad en el desarrollo de software es un proceso crítico que involucra diversas estrategias y enfoques para asegurar que el producto final cumpla con los estándares esperados de rendimiento y funcionalidad. Un aspecto esencial es la implementación de un Sistema de Gestión de Calidad (SGC) que proporciona un marco estructurado para gestionar y mejorar continuamente la calidad del software (ISO/IEC 25010, 2023), ayudando a las

organizaciones a definir políticas, procedimientos y responsabilidades para alcanzar los objetivos de calidad del software.

La incorporación de metodologías ágiles ha demostrado ser efectiva en la gestión de la calidad del software, ya que promueven ciclos de desarrollo iterativos e incrementales, permitiendo una revisión y mejora continua del software durante su desarrollo. Las prácticas ágiles, como las pruebas automatizadas y la integración continua, son cruciales para la detección temprana de defectos y la mejora de la calidad del software (Beck et al., 2018).

La cultura organizacional también juega un papel fundamental en la calidad del software, ya que fomenta la colaboración, la comunicación abierta y la responsabilidad compartida entre los equipos de desarrollo y pruebas. Es crucial para lograr altos estándares de calidad. En este contexto, la implementación de DevOps, que integra las operaciones y el desarrollo de software, facilita una colaboración más estrecha y una mayor eficiencia en la entrega de software de alta calidad (Kim et al., 2021).

Además, la capacitación continua del personal es indispensable para mantener y mejorar la calidad del software en entornos ágiles. Invertir en la formación sobre mejores prácticas de calidad y nuevas tecnologías no solo garantiza que los equipos tengan las habilidades necesarias para producir software de alta calidad, sino que también ayuda a los empleados a adaptarse a nuevas metodologías y herramientas, mejorando su desempeño y compromiso, lo cual es crucial para el éxito de los proyectos de desarrollo de software (Flores-Cerna et al., 2022).

Tras examinar la importancia de la organización de la calidad en el desarrollo de software para asegurar que los productos finales entregados sean de alta calidad, es fundamental centrarse en los equipos responsables de implementar estas estrategias. A continuación, se analizará el papel y la efectividad de los equipos de calidad de software reducidos, quienes,

a pesar de su tamaño, desempeñan un papel crucial en la adaptación rápida a los cambios y el mantenimiento de altos niveles de calidad dentro de entornos ágiles.

#### ***2.4.1. Estándares de calidad***

En el ámbito de la calidad de software, los estándares juegan un papel crucial, ya que proporcionan un marco estructurado para definir y asegurar los criterios de calidad aplicables a un producto, permitiendo una evaluación consistente y objetiva de su desempeño y conformidad (Pacheco, 2024). De acuerdo con Díaz y Silega (2021), la industria exige productos y servicios de alta calidad, respaldados por modelos, estándares y metodologías de reconocimiento internacional.

Existen varios estándares que establecen un marco estructurado para definir procesos de prueba, y entre los más reconocidos a nivel global se encuentran la ISO/IEC 14598:2001 y la ISO/IEC 25040:2011, que establecen directrices para la evaluación de productos de software, promoviendo un enfoque metódico para definir, documentar y ejecutar procesos de prueba. Estos estándares no solo detallan cómo evaluar la calidad del producto, sino que también se adaptan a características específicas del software, asegurando que los productos cumplan con criterios técnicos y de satisfacción del usuario en diversas etapas del desarrollo (Díaz y Silega, 2021).

#### ***2.4.2. Distribución de Roles dentro de los Equipos de Calidad de Software***

Dentro del ambiente del aseguramiento de la calidad en el desarrollo de software, la asignación de roles o puestos de trabajo es de vital importancia para lograr garantizar la eficiencia y el éxito del proceso de pruebas que se llevan a cabo a lo largo del ciclo de vida de un producto; es por eso que dentro de los diferentes equipos de calidad se encuentran diversos roles, los cuales cumplen responsabilidades específicas que contribuyen al ciclo

entero de pruebas. Cabe resaltar que estos roles varían dependiendo de la empresa y dentro de este apartado se resaltan dos categorías que son los líderes o gerentes de pruebas, quienes están a cargo de todo el proceso de pruebas, mientras que por el otro lado se encuentran los ingenieros de prueba, quienes se encargan de la ejecución directa de las pruebas al software (Baquero, 2024).

#### ***2.4.3. Estrategias de Coordinación y comunicación en los Equipos de Calidad de software***

Diversos estudios destacan la importancia de la comunicación, la coordinación y cooperación para el éxito del proyecto, como pilares fundamentales para el éxito de los proyectos en los equipos de calidad de software (QA), especialmente en entornos reducidos, donde cada miembro desempeña un rol crucial para garantizar la efectividad del equipo (Dietrichson et al., 2022; Severo y Andrade, 2024).

De acuerdo con De Farias et al. (2021), existen diversas estrategias diseñadas para mejorar la coordinación y comunicación en los equipos de QA. Estas estrategias están respaldadas por evidencia empírica y enfoques prácticos que buscan abordar los desafíos específicos de los entornos colaborativos, maximizando el potencial de los equipos y mitigando los riesgos asociados a la falta de sincronización.

Una de las estrategias más relevantes consiste en fomentar la transparencia y la fluidez de los canales de comunicación mediante herramientas colaborativas que permitan compartir información de manera efectiva. Esto puede lograrse mediante documentación clara y accesible para todos los miembros, la realización de reuniones diarias o semanales, como las stand-ups propias de metodologías ágiles, para compartir avances, resolver dudas y anticipar problemas, así como el uso de herramientas colaborativas como Trello o Microsoft Teams, que agilizan las interacciones y mejoran la cohesión del equipo (De Farias et al., 2024).

Otro de los puntos claves es la coordinación de los equipos, lo cual implica definir los roles y responsabilidades de manera que cada uno tenga tareas asignadas, para evitar errores, omisiones o redundancias. Las plataformas como Jira, Trello o Asana permiten asignar tareas, priorizarlas y monitorear el progreso en tiempo real (Severo y Andrade, 2024).

Como se observa, estas estrategias son complementarias y no solo fortalecen la dinámica interna de los equipos de QA, sino que también mejoran su capacidad para adaptarse a los desafíos específicos que enfrentan en entornos de trabajo reducidos.

## **2.5 Equipos de calidad de software reducidos**

Los equipos de calidad de software han ganado relevancia en el desarrollo ágil de software moderno, en gran parte debido a su capacidad de adaptarse rápidamente a los cambios y mantener altos niveles de calidad. Estos equipos, aunque pequeños, implementan prácticas ágiles como Scrum, que favorecen la colaboración estrecha y la rápida iteración de productos. La agilidad en equipos pequeños permite una comunicación más eficiente y una mayor flexibilidad en la toma de decisiones, ya que tienden a ser más cohesivos y pueden responder rápidamente a las necesidades del proyecto, lo que resulta en una mejora continua de la calidad del software. Además, la implementación de metodologías ágiles como Scrum se ha demostrado eficaz no solo en equipos pequeños colocalizados, sino también en contextos distribuidos, lo que subraya su versatilidad y adaptabilidad en diversas situaciones empresariales (Spiegler et al., 2021).

Las prácticas de colaboración y comunicación efectiva dentro de equipos pequeños facilitan la distribución equitativa del liderazgo y la responsabilidad. En un entorno ágil, roles como el del Scrum Máster pueden ser desempeñados por diferentes miembros del equipo a medida que este madura, promoviendo la autoorganización y la reducción de la dependencia de un único líder (Spiegler et al., 2021). Respecto a este enfoque, no solo mejora la distribución de

tareas, sino que también contribuye a un ambiente de trabajo más colaborativo y empoderado, lo que a su vez puede conducir a una mayor motivación y productividad del equipo (Spiegler et al., 2021). Además, los equipos pequeños se benefician de una mayor cohesión y tienen la capacidad de adaptarse rápidamente a los cambios del proyecto, manteniendo siempre un alto nivel de calidad en el software entregado (Trippensee & Remané, 2021).

Además, la implementación de marcos ágiles escalados, como el Scaled Agile Framework (SAFe), permite que incluso los equipos pequeños se integren en proyectos más grandes sin perder la esencia de su agilidad y eficacia; estos marcos proporcionan estructuras para mantener la coherencia y la calidad en equipos distribuidos y en proyectos de mayor envergadura (Santos & de Carvalho, 2022).

Por consiguiente, al explorar la dinámica y las ventajas de los equipos de calidad de software reducidos dentro de entornos ágiles, resulta importante analizar cómo las estrategias para optimizar la calidad en el desarrollo de software han evolucionado con el tiempo. Seguidamente, se presenta una revisión histórica de las principales estrategias utilizadas para asegurar la calidad de software, destacando los hitos clave que han moldeado las prácticas actuales en la industria.

### ***2.5.1. Comparativa entre equipos grandes y pequeños de calidad de software***

Es determinante para las empresas de desarrollo de software conocer las principales diferencias entre los tamaños de sus equipos con el fin de optimizar sus procesos y asegurar el cumplimiento de los estándares de calidad del producto. Esto deriva en la necesidad de conocer muy bien la cantidad de profesionales que necesitan para poder cumplir con lo mencionado anteriormente. Según Marsicano et al. (2017), se resaltan tres puntos clave para la comparativa entre estos dos tipos de equipos: el manejo de los roles en equipos ágiles de

desarrollo, la madurez y la calidad del equipo y, por último, el tema de la escalabilidad y la coordinación en los equipos grandes. Es importante conocer cada uno de estos puntos, ya que brindan un panorama claro sobre las labores y principales diferencias que distinguen a estas dos estructuras o modelos de trabajo. A continuación, se detallan estos para dar una perspectiva clara de ellos:

- a) **Manejo de los roles en los equipos:** Para los equipos, tanto de desarrollo como de calidad, estos se destacan por su autonomía y por su auto-organización, lo que deriva en una adaptabilidad en aspectos como los proyectos ágiles. Mientras que, cuando se observan a los equipos grandes, se observa que estos enfrentan retos relacionados con la coordinación y con una clara definición de los roles, en especial cuando se habla de entornos que son distribuidos, ya que en estos campos la complejidad se ve aumentada significativamente.
- b) **Madurez de los equipos y la calidad:** Es importante mencionar que la madurez genera un impacto en equipos de calidad de software sin importar el tamaño; no obstante, los equipos de gran tamaño necesitan estructuras más sólidas con el fin de mantener la consistencia en los procesos de calidad. Por otro lado, los equipos reducidos suelen adaptarse más rápido a los cambios y priorizar tareas vitales debido al número reducido de integrantes; sin embargo, estos pueden carecer de personas especializadas, lo que en equipos grandes sí pueden ofrecer.
- c) **Coordinación entre equipos de gran tamaño y su escalabilidad:** Es fundamental evidenciar que cuando se habla de equipos de trabajo grandes en la calidad del software, se enfatiza que estos equipos se ajustan más a proyectos de grandes escalas, en gran medida por la posibilidad que estos adoptan de poder dividir las responsabilidades en subgrupos especializados. No obstante, este tema también se ve

sujeto a aspectos que pueden generar problemas con la comunicación, lo cual impacta de manera negativa la calidad del servicio si este no es gestionado de una manera correcta. En contraste con esto, para abordar los equipos pequeños, estos, al contar con un número reducido de personas, se ven afectados positivamente con comunicaciones más fluidas y además con procesos iterativos más cortos (Hoda et al., 2012).

### ***2.5.2. Experiencias en la implementación de estrategias para optimizar la calidad de software***

La búsqueda de un producto final de alta calidad en el ámbito del software ha sido una prioridad fundamental para las organizaciones, lo que ha conducido a la adopción de diversas estrategias para mejorar el desarrollo y la entrega del software. Las experiencias con estas estrategias varían según el contexto de las organizaciones, las características de los equipos de desarrollo y las especificidades de cada proyecto.

Alanoca (2021) reporta que la implementación de un modelo de pruebas de regresión automatizadas dentro de procesos de integración continua en el desarrollo de sistemas web que emplean metodologías ágiles condujo a una mejora del 29.16 % en la detección temprana de errores en el software. Este modelo fue implementado en un sistema web para la gestión de agenda de entidades financieras que facilita la administración de la cartera de clientes, incluyendo el registro y recuperar datos del cliente, así como el monitoreo de crédito, flujo de caja y rendimiento.

Lalband & Kavitha (2020) determinaron que los proyectos que adoptan metodologías ágiles alcanzan una tasa de éxito del 71 %, sugiriendo con ello que estas metodologías no solo permiten prever con mayor precisión los requisitos de personal en función de la complejidad

del producto y los plazos de entrega, sino que también incrementan significativamente las probabilidades de éxito en los proyectos de software, tal y como lo confirma el estudio de Arenas (2022).

La implementación de prácticas DevOps ha mostrado una mejora considerable en la calidad del software, ya que al integrar la cultura a lo largo del ciclo de vida del producto ha demostrado ser eficiente para mejorar la calidad del software mediante la automatización y la integración continua. Diversos estudios sugieren que la adopción de DevOps facilita una entrega más rápida y eficiente, además de fomentar una mejor colaboración entre los equipos de desarrollo y operaciones (Mishra & Otaibi, 2020).

### ***2.5.3. Eficiencia en la solución y detección de defectos en los equipos de calidad de software***

El aseguramiento de la calidad del software (QA) es un proceso esencial en el ciclo de vida del desarrollo de software, especialmente en la identificación y resolución de errores, ya que el realizarlo no solo contribuye a la mejora del producto final, sino que también minimiza los costos adicionales derivados de estos, al no ser detectados a tiempo (Mahmoud & Santos, 2021). Bajo esta misma línea, Pariona & Vargas (2024) sostienen que la calidad no solo depende de las funcionalidades del producto, sino también de la capacidad para identificar y corregir defectos antes de que este afecte al cliente o usuario.

En este sentido, la adopción de metodologías ágiles, como Scrum y Kanban, ha sido ampliamente reconocida como una estrategia efectiva para mejorar la eficiencia en la detección y solución de defectos. Para ello, Nogueira et al. (2016) mencionan que las metodologías ágiles facilitan un enfoque de tipo iterativo y promueven la detección de errores a través del desarrollo de los diversos ciclos de desarrollo cortos y la colaboración constante

entre los diversos departamentos de la empresa de desarrollo de software. Además, el uso de herramientas automatizadas para la ejecución de pruebas, como Selenium o JUnit, permite una mayor cobertura de pruebas y una detección más rápida de defectos en las primeras fases del ciclo de vida del software.

La eficiencia en la detección de defectos también depende de una gestión efectiva de los procesos de retroalimentación. Según Poppendieck y Poppendieck (2003), la retroalimentación continua es crucial en las metodologías ágiles, ya que permite a los equipos reaccionar rápidamente a los problemas detectados, ajustando el desarrollo según sea necesario. Este enfoque iterativo contribuye a una solución más eficiente de los defectos, al permitir correcciones inmediatas y ajustar la dirección del proyecto antes de que los problemas se agraven.

#### ***2.5.4. Relevancia de la experiencia previa en la detección de errores***

De acuerdo con Vélez et al. (2023), los errores preanalíticos son los más comunes en el proceso de desarrollo de software, y la experiencia juega un papel fundamental en ella, ya que el contar con profesionales ampliamente calificados permite que estos puedan identificar defectos con mayor precisión, lo que permite reducir significativamente tanto el costo económico como el tiempo necesario para corregir estos a lo largo del ciclo de vida del desarrollo. Por otro lado, Cabable y Troilo (2021) sostienen que el personal experimentado no solo identifica los errores de manera eficiente, sino que también aporta un valor considerable al equipo, ya que logra intervenir de manera eficiente en las áreas necesarias del proyecto, siendo proactivo, colaborativo y mejorando continuamente el desempeño colectivo, optimizando los resultados del equipo.

Asimismo, Guevara et al. (2022) identifica varios factores que son claves y que influyen directamente en la calidad del software, de los cuales menciona la elección del modelo de pruebas, la experiencia del colaborador, la combinación de pruebas automáticas y manuales, la estimación de costos, la selección de la metodología y la tipología de pruebas; y aclara que la interacción de esta garantiza la efectividad del proceso de aseguramiento en la calidad.

#### ***2.5.5. Adaptabilidad hacia las nuevas herramientas y metodologías***

En la actualidad, es esencial para las empresas relacionadas con el desarrollo de software adaptarse a nuevas herramientas y metodologías, con el fin de garantizar tanto su competitividad en el mercado, así como la sostenibilidad de ella misma a lo largo del tiempo. Implementar sistemas de gestión de proyectos híbridos fomenta dentro de las empresas crear estructuras detalladas de documentación en los proyectos, como también una comunicación tanto asertiva como fácil de realizar (Leong et al, 2023). Por otro lado, Rocha et al. (2023) mencionan que el uso con medida de herramientas a la vanguardia significa aspectos que acercan a estas empresas a un éxito en la calidad del servicio que se brinda, logrando mantener una línea competitiva y eficiente. Es importante mencionar que, cuando se habla de la adaptabilidad en nuevas herramientas y metodologías, se deben de considerar aspectos tales como:

- a) La flexibilidad entre metodologías: Combinando diferentes enfoques como lo son las metodologías tradicionales y las ágiles, ya que en la actualidad se observa como cada vez las empresas empiezan a implementar estos enfoques, debido a que, por ejemplo, enfoques como el de cascada señalan que mantener una documentación clara y precisa es de vital importancia, mientras que enfoques ágiles fomentan una iteración

rápida, lo cual da como resultado una mejora considerable en la sostenibilidad de los proyectos.

- b) Capacitaciones y la adaptabilidad de los equipos: En la actualidad es determinante para las empresas capacitar a su personal, para mantener una competitividad y además garantizar que estos puedan adaptarse tanto a nuevas herramientas como a nuevas prácticas.
- c) Automatización de pruebas y la adaptabilidad en los equipos: Integrar herramientas para la automatización con enfoques ágiles ha evidenciado la gran mejora en temas de costos y tiempos a la hora de realizar pruebas repetitivas, lo cual genera una mejora en la productividad de los equipos. Es importante mencionar que la automatización de pruebas libera de muchas tareas repetitivas a los encargados de hacer estos procesos, lo que a la postre produce que estas personas puedan encargarse de otro tipo de tareas. Por otra parte, la agilidad busca fomentar la flexibilidad dentro de los equipos, lo que impulsa una entrega continua y, además, que genere valor (Leong et al., 2023).

## **2.6 Estrategias para optimizar la calidad en el desarrollo de software**

De acuerdo con Sommerville (2016), el desarrollo de software ha experimentado una evolución significativa en las estrategias para garantizar la calidad del producto final. Inicialmente, se empleaban métodos de prueba manual intensivos para identificar y rectificar errores en el código, pero con el avance notorio en temas tecnológicos y la complejidad creciente de los sistemas, surgieron enfoques más sofisticados y eficaces.

Un punto donde se marcó un antes y un después en esta evolución fue la adopción extendida de metodologías ágiles, como Scrum y la Programación Extrema, a principios de los años 2000's. Estas metodologías se caracterizan por su enfoque en el desarrollo iterativo e

incremental, la colaboración cercana entre equipos multifuncionales y la entrega continua de funcionalidades (Sutherland, 2020). Introdujeron prácticas clave como la integración de pruebas automatizadas y la retroalimentación continua por parte de los clientes, las cuales fueron aspectos determinantes para mejorar tanto la eficiencia como la calidad del software que se desarrolla.

En los últimos años, el enfoque en DevOps ha transformado la colaboración entre desarrollo y operaciones, promoviendo la automatización de procesos de prueba, despliegue y monitoreo en todo el ciclo de vida del software (Kim et al., 2021). Esta metodología no sólo acelera los tiempos de entrega, sino que también mejora la estabilidad y la eficiencia operativa de los sistemas en producción.

Actualmente, las organizaciones adoptan prácticas DevOps y metodologías ágiles para mantenerse competitivas en un mercado dinámico y exigente, puesto que estas aseguran la calidad del software y mejoran la satisfacción del cliente al proporcionar productos más rápidos, seguros y confiables.

Tras haber examinado y revisado la evolución histórica de las estrategias para optimizar la calidad en el desarrollo de software, es imperativo investigar las estrategias que, en la actualidad, se consideran fundamentales en este ámbito. En el siguiente apartado se procederá a un análisis detallado de distintas metodologías y prácticas que, en su conjunto, constituyen el núcleo de las estrategias modernas para asegurar la calidad en el desarrollo de software.

## **2.7 Tipos de estrategias para optimizar la calidad en el desarrollo de software**

El desarrollo de software es una actividad compleja que demanda la implementación de diversas estrategias para asegurar la calidad del producto final, que pueden clasificarse en varias categorías principales, como lo son metodologías ágiles, desarrollo basado en pruebas,

integración continua y automatización de pruebas y la gestión de la configuración del software. A continuación, se presenta cada una de estas categorías.

### ***2.7.1. Metodologías ágiles***

Las metodologías ágiles, como Scrum y Kanban, se destacan por su flexibilidad y la capacidad de respuesta rápida a los cambios. Estas metodologías dividen el trabajo en ciclos cortos y manejables, llamados sprints, y al final de cada uno, se entrega un incremento funcional del producto. Según Mishra & Al Zoubi (2023), su enfoque en la colaboración y la adaptabilidad las hace particularmente beneficiosas para proyectos pequeños y medianos.

### ***2.7.2. Desarrollo basado en pruebas***

El desarrollo basado en pruebas (TDD) es una práctica que consiste en escribir las pruebas antes del código de producción, lo que asegura que cada nueva función esté cubierta por pruebas automatizadas, facilitando la detección temprana de errores en el ciclo de desarrollo. Basado en lo anterior, Barrado et al. (2021) mencionan que un TDD puede mejorar significativamente la calidad del software al promover un diseño más limpio y un mejor entendimiento de los requisitos.

### ***2.7.3. Integración continua y automatización de pruebas***

La integración continua (CI) y la entrega continua (CD) son prácticas que involucran la automatización de la construcción, prueba y despliegue del software, lo que permite una detección y corrección de defectos más rápida y efectiva, mejorando la calidad del producto final. Según Khatri (2022), la CI/CD minimiza los errores humanos y proporciona una solución rentable al integrar la calidad dentro del ciclo de entrega del software (Khatri, 2022).

#### ***2.7.4. Gestión de la configuración del software***

La gestión de la configuración del software implica el seguimiento y control de los cambios en el software para mantener la integridad y la trazabilidad a lo largo del ciclo de vida del desarrollo. Esta práctica es crucial para gestionar las versiones del software, asegurando que cada cambio sea documentado y reversible en caso de problemas. De acuerdo con Van Casteren (2017), una gestión de configuración efectiva es clave para mantener la calidad del software, especialmente en proyectos grandes y complejos.

Para comprender la eficacia de estas estrategias, es crucial analizar cómo han sido implementadas en diferentes entornos y proyectos. Este análisis proporcionará una visión más completa y realista sobre los beneficios y desafíos de cada una de ellas. Seguidamente, se abordarán diversas experiencias y estudios que han documentado la implementación de estas estrategias para optimizar la calidad de software en contextos específicos.

## **Capítulo III. Marco metodológico**

En este capítulo se abordará la metodología a implementar, incluyendo el enfoque, el tipo de investigación que se llevará a cabo, así como otros elementos que complementan esta sección.

### **3.1 Enfoque de la investigación**

Esta investigación facilitó el análisis, recopilación y extracción de datos de calidad provenientes de unas fuentes bibliográficas confiables, así como también la implementación de diversas técnicas como la revisión de la literatura, entrevistas, estudios de caso de estudio y evaluación por pares.

En cuanto al enfoque investigativo, hace alusión a la dirección y orientación que pretende guiar la investigación hacia la solución del problema de estudio. De acuerdo con Hernández Sampieri (2020), este permite determinar todo el proceso investigativo, incluyendo las etapas, elementos y resolución del procesamiento de datos, por lo cual cada enfoque tiene sus propias características.

En este sentido, Ramírez (2020) clasifica los enfoques investigativos en tres categorías principales: cualitativo, cuantitativo y mixto. Dado que esta investigación se centra en la optimización de procesos de control de calidad (QA) en empresas de desarrollo de software con equipos reducidos, el enfoque mixto resulta ser el más adecuado. Este enfoque permite la combinación de los enfoques cualitativo y cuantitativo. La integración de ambos permite una perspectiva que ayuda a comprender mejor el fenómeno de estudio. (Hernández Sampieri 2020). Desde la parte cuantitativa se busca recolectar y analizar datos de carácter numérico sobre la percepción y experiencia de los profesionales del área, mediante utilizarán encuestas estructuradas que permitirán medir variables relacionadas con la eficiencia de los procesos de QA, la satisfacción de los miembros del equipo y la efectividad de las estrategias

implementadas. Por otro lado, el componente cualitativo se encargará de analizar los contextos y procesos a través de una revisión literaria y el análisis situacional, utilizando un caso de estudio. explorando las experiencias y percepciones de profesionales del área. La revisión permitirá contextualizar la investigación en el marco teórico existente, identificando teorías y modelos relevantes sobre la optimización de QA, mientras que el análisis de caso proporciona una comprensión detallada de la aplicación práctica de las estrategias a implementar, por medio de un ejemplo concreto de caso real, enriqueciendo el análisis.

### **3.2 Tipo de investigación**

Esta investigación facilitó el análisis, recopilación y extracción de datos de calidad provenientes de fuentes bibliográficas confiables, así como también la implementación de diversas técnicas como las entrevistas, estudios de caso y evaluaciones por pares. Estas técnicas no solo ayudaron al investigador a estructurar información, sino que también subrayan la importancia de conocer las definiciones, características, ventajas y desventajas de dichas técnicas. A continuación, se detalla cada una de ellas:

#### ***3.2.1 Revisión de la literatura***

Según Lloyd (2018), la revisión de la literatura consiste en analizar y sintetizar investigaciones previas y teorías relevantes al tema de estudio. Este proceso es de vital importancia por varias razones, como la contextualización y justificación del estudio, las decisiones metodológicas, el desarrollo del marco teórico y la identificación de tendencias y patrones.

En cuanto a las ventajas de esta técnica, Green Arias (2023) menciona que esta técnica proporciona un panorama general sobre los conceptos más relevantes para la investigación en curso. que tienen que ver con lo que se está investigando. Debido a su naturaleza, permite

acceder a una amplia diversidad de fuentes y recursos, lo cual facilita la recopilación de información. Además, resulta útil para evaluar, facilita la evaluación de citas bibliográficas previas en relación con las utilizadas en el estudio actual (Green, 2023).

Esto concuerda con lo propuesto por García-Peñalvo (2017), que señala que una revisión sistemática implica una búsqueda minuciosa sobre un tema en particular. Una vez identificados y recopilados los estudios, los resultados son sintetizados siguiendo un método preestablecido y claramente definido.

La principal desventaja de la revisión de la literatura radica en que requiere un esfuerzo considerable en la búsqueda y recopilación de información, lo cual consume un tiempo significativo.

En este TFG, la aplicación de esta técnica proporcionó una base sólida de materiales bibliográficos para el desarrollo de la investigación.

### **3.2.2 Entrevistas**

Díaz-Bravo et al. (2013) definen la entrevista como una conversación con un propósito que trasciende el simple intercambio verbal, describiéndola como una forma de comunicación interpersonal entre el investigador y el sujeto del estudio, cuyo objetivo es obtener información relevante sobre el problema de investigación. Entre las principales ventajas de esta técnica, destacan su amplia versatilidad y aplicación, ya que no está limitada por restricciones de espacio o tiempo. Además, permite enfocar el tema de manera específica y ofrece una perspectiva tanto interna como externa sobre el asunto en cuestión (Díaz et al., 2013).

Por su parte, Folgueiras Bertomeu (2016) señala que la entrevista es una técnica de recolección de datos ampliamente utilizada en los procesos de investigación, aunque presenta algunas limitaciones. Estas incluyen la dificultad para identificar a las personas idóneas para entrevistar, la complejidad en la interpretación y el análisis de los datos obtenidos, así como el aumento de costos y el tiempo requerido, dependiendo de la cantidad de muestras.

En el contexto de este TFG, la entrevista contribuyó significativamente a estructurar la información de manera precisa, ya que las preguntas se formularon de forma específica en relación con los temas analizados, utilizando la escala de Likert, que permitió que los entrevistados expresaran su grado de acuerdo o desacuerdo con afirmaciones relacionadas con QA, facilitando así la recolección de datos cuantificables, lo que ayudó a minimizar el sesgo de información por parte de los profesionales entrevistados.

### ***3.2.3 Caso de estudio***

Jiménez (2012) propone que el enfoque de estudio de caso se centra en profundizar en los datos específicos del fenómeno investigado, revelando toda la información disponible para ofrecer un panorama preciso del tema o situación de estudio. Este enfoque se distingue por su adaptabilidad a diversos tipos de investigación y por la posibilidad de combinarse tanto con métodos cuantitativos como cualitativos. Esta técnica permite realizar un análisis profundo y detallado del fenómeno estudiado, lo que enriquece el conocimiento existente y facilita la identificación de áreas de mejora, así como la comprensión de un campo específico. Por otra parte, Alvarado (2018) afirma que un caso de estudio puede abarcar diferentes sujetos de análisis, como una persona, un grupo de individuos, un aula, un programa, un recurso, un cambio o una institución, destacándose por su singularidad y especificidad en relación con otras realidades. El objetivo central de esta técnica es describir, comprender y

analizar de manera detallada el comportamiento del caso en cuestión, considerando tanto su historia como su contexto, lo que permite anticipar su comportamiento y tomar decisiones informadas para su mejora.

En cuanto a las desventajas de aplicar esta técnica, tanto la validez como la confiabilidad en los estudios de caso dependen en gran medida de la calidad de los datos recopilados. Además, el análisis detallado de los datos puede requerir una cantidad significativa de tiempo y recursos, lo que puede reducir la eficiencia del proceso (Jiménez, 2012).

En este TFG, el caso de estudio se empleó con la finalidad de analizar en profundidad un fenómeno específico dentro del contexto de la calidad del software. Este enfoque fue seleccionado debido a su capacidad para proporcionar una comprensión rica y contextualizada de los procesos y dinámicas que caracterizan a los equipos de trabajo en entornos de desarrollo de software. Al elegir un caso representativo, se buscó ilustrar tanto los desafíos como las oportunidades que enfrentan los equipos reducidos en la detección y solución de defectos. La elección del caso de estudio permitió abordar el fenómeno basado en preguntas de investigación que requieren un análisis detallado y contextual, ya que se centra en una realidad particular y se apoya en múltiples fuentes de información, como entrevistas, observaciones y revisión de documentación, facilitando una exploración más completa del fenómeno, sino que también brindó la oportunidad de captar las percepciones y experiencias de los miembros del equipo, lo que enriquece la interpretación de los resultados.

### ***3.2.4 Evaluación de pares***

Según Bowen (2023), esta técnica permite evaluar los logros obtenidos en diversas actividades propuestas por los estudiantes, siempre y cuando se garantice tanto la validez

como la confiabilidad de las herramientas empleadas, las cuales son consideradas de apoyo. En la misma línea, Bowen (2023) explica que la evaluación por pares ofrece varias ventajas, tales como proporcionar una oportunidad de mejora para el trabajo en su totalidad, ya que al implementarla, contribuye a reducir el sesgo de información, favoreciendo un análisis más objetivo.

El Tecnológico de Monterrey (Innovación Educativa, 2018) define la evaluación de pares como un proceso en el cual una o más personas valoran el trabajo realizado con el objetivo de realizar una revisión externa del mismo. Asimismo, Bowen (2023) señala varias desventajas asociadas a la aplicación de esta técnica. En primer lugar, el tiempo de entrega puede verse afectado por la limitada disponibilidad de los evaluadores. Además, la objetividad del análisis podría verse comprometida si existe algún tipo de relación entre el evaluador y el evaluado, lo que puede influir en la imparcialidad de la evaluación y, en consecuencia, en la calidad del resultado.

Respecto a la implementación de esta técnica en el trabajo de fin de grado (TFG), se utilizó para evaluar la propuesta para optimizar los procesos (QA) en entornos reducidos, con el objetivo de optimizar los procesos, la cual fue presentada a profesionales con experiencia en estas condiciones específicas, quienes como pares evaluaron las mejoras planteadas en este TFG.

### **3.3 Fuentes y sujetos de información**

#### ***3.3.1. Fuentes de información***

En el contexto de la investigación, las fuentes son todos los recursos o materiales en los que se extrae toda la información relevante para el estudio. Estas pueden ser compuestas por

artículos científicos y libros, hasta bases de datos académicas, que son evaluadas por su relevancia y su rigor académico para aportar al conocimiento del tema que se está investigando (Hernández & Mendoza, 2022). De acuerdo con este mismo autor, existen tres tipos de fuentes; sin embargo, en vista de los objetivos de la investigación, las fuentes primarias y secundarias.

- a) **Fuentes primarias:** De acuerdo con Linares et al. (2021), son herramientas diseñadas para recolectar datos que no se obtuvieron de las fuentes secundarias; estas incluyen encuestas, entrevistas, trabajos en equipo, muestreo, análisis de laboratorio, historias de vida y visitas de campo en grupo, entre otras.
- b) **Fuentes secundarias:** Bajo la perspectiva de Linares et al. (2021), menciona que las fuentes secundarias provienen tanto de documentos como de informes, estadísticas o mapas, y que a su vez estos han sido creados con propósitos ajenos a la investigación actual. No obstante, se menciona que estos recursos pueden ser de gran ayuda para alcanzar los objetivos del estudio, e incluso, en ocasiones, pueden brindar información suficiente para realizar el diagnóstico requerido.

### ***3.3.2 Sujetos***

Según menciona Ramírez et al. (2023), el sujeto de investigación son los elementos específicos que participan directamente en la investigación; por ende, son profesionales clave involucrados en el proceso de control de calidad en el desarrollo de software, como lo son el gerente de proyectos, el líder de aseguramiento de calidad (QA Lead), el analista de aseguramiento de calidad senior (Senior QA), el analista de aseguramiento de calidad junior (Junior QA) y el encargado de desarrollo de software (Head of Development), los cuales son

fundamentales para comprender cómo se gestionan los procesos de control de calidad en entornos de equipo reducido.

### **3.4 Población y muestra**

En el siguiente apartado se aborda a detalle la población de estudio y los criterios de selección de la muestra.

#### ***3.4.1. La población***

La población en un estudio se refiere al grupo amplio y general sobre el cual se pretende realizar el análisis y obtener conclusiones significativas. Según Ramírez et al. (2023), puede consistir en personas, objetos, eventos u otros elementos relevantes para el objetivo investigativo. En el contexto de esta investigación, la población está conformada por el conjunto de profesionales de calidad de software (QA) que operan con equipo reducido.

En vista de que la investigación busca desarrollar una propuesta de estrategias aplicables para optimizar los procesos de QA, la población también incluye a las empresas que utilizan estos equipos reducidos, ya que se espera que también sea aplicable y relevante para este tipo de entorno.

#### ***3.4.2. Muestra***

La muestra es un componente crucial en el proceso investigativo, ya que asegura la selección representativa de la población total, por lo cual es fundamental que refleje las características de esta para que los resultados sean válidos y generalizables (Ramírez et al., 2023). Esto concuerda con lo postulado por Arias (2021), quien afirma que se trabaja con una representación de los diversos componentes que conforman una población.

Considerando lo anterior, y en respuesta a las intenciones de estudio, el muestreo más adecuado es el muestreo no probabilístico por conveniencia (Arias, 2021), ya que se seleccionan profesionales del área de aseguramiento de calidad (QA) y empresas de desarrollo de software para el estudio de caso, que proporcionen información relevante sobre los procesos.

Asimismo, el muestreo por conveniencia permite asegurar que la información provenga de participantes que estén disponibles y tengan accesibilidad, por lo cual se han seleccionado 11 profesionales del área de QA, los cuales serán considerados en las técnicas. Este enfoque asegura que la muestra sea práctica y que los datos recopilados sean pertinentes para el análisis de los procesos y la propuesta de estrategias en empresas con equipos de QA reducidos.

## **Capítulo IV. Diagnóstico y análisis de resultados**

En este capítulo se presentan los resultados obtenidos con respecto a la implementación de las técnicas aplicadas dentro de este proyecto final, que incluyen la revisión de la literatura, entrevistas, casos de estudio y evaluación par. La combinación de estas herramientas permite desarrollar una propuesta orientada a optimizar los procesos de control de calidad (QA) en la empresa de desarrollo de software con equipos reducidos.

#### **4.1 Revisión de la literatura**

Con el fin de proporcionar una base teórica sólida para esta investigación, se llevó a cabo una extensa revisión de la literatura con la finalidad de crear un marco teórico en donde se presentan los conceptos clave necesarios para abarcar el estudio que se efectúa, los cuales fueron expuestos en el segundo capítulo de esta tesis. A continuación, se presentan los principales hallazgos derivados de este proceso.

Con el fin de proporcionar una base teórica sólida para esta investigación, se llevó a cabo una exhaustiva revisión de la literatura que permitió construir un marco conceptual que abarca los principales temas y teorías relacionadas con el objeto de estudio. Este proceso resultó en la selección de 40 referencias bibliográficas, las cuales fueron consultadas para sustentar y guiar el análisis de la investigación. Cabe destacar que se consultaron fuentes bibliográficas de alta calidad, extraídas propiamente de sitios académicos especializados y de alta calidad, lo que garantiza la confiabilidad y rigor de la información empleada.

Es relevante señalar que las fuentes bibliográficas consultadas se clasifican en dos tipos: primarias y secundarias. Las referencias primarias son aquellas que provienen directamente de los trabajos de investigación originales, como artículos de revistas científicas y actas de conferencias. Este tipo de fuentes aporta datos y resultados directos de investigaciones previas. En cambio, las fuentes secundarias son aquellas que presentan, interpretan o resumen

los resultados de las investigaciones primarias, como libros, artículos de revisión o sitios web que recogen los hallazgos de otros autores. En este caso, de las 40 referencias empleadas, un porcentaje significativo corresponde a trabajos de investigación primaria, mientras que el resto se conforma de fuentes secundarias que recogen o exponen los resultados de investigaciones previas.

Después de esta aclaración, es necesario señalar que, en esta investigación, el 62.5 % (25) de las fuentes fueron primarias y el 37.5 % fueron secundarias. De esas, el 72.5 % (29) de las referencias provenían de revistas académicas, mientras que el 25 % (10) corresponde a estudios, sitios web y otros artículos de carácter académico. El análisis de antigüedad de las referencias revela que solo el 22.5 % (9) corresponde a fuentes anteriores a 2019; sin embargo, estas se incluyeron debido a su relevancia en el tema de investigación, y el restante sí cumplen con el criterio de ser posterior a los últimos cinco años.

## **4.2 Entrevistas**

Se realizaron un total de 11 entrevistas a profesionales especializados en calidad de software, quienes laboran en entornos de equipos reducidos, con el fin de conocer las problemáticas más recurrentes que afectan estos equipos en la práctica diaria. Para ello, se elaboró una serie de preguntas que permitió guiar la entrevista hacia los aspectos más relevantes de la investigación.

#### **4.2.1 Datos demográficos**

Se realizaron entrevistas con 11 participantes, de los cuales 3 eran mujeres, lo que representaba un poco menos de un tercio de los participantes (27 %) y el resto (73 %) eran hombres.

Los participantes tenían un rango de edades diferentes, donde seis de los participantes (55 %) se encontraban entre los 25 y 34 años, tres (3/11, 27 %) entre 35 y 45 años, y uno (1/11, 9 %) entre 45 y 54 años. En cuanto al nivel educativo, la mayoría de los entrevistados (9/11, 82%) habían completado estudios de bachillerato universitario, mientras que uno (1/11, 9 %) tenía una licenciatura y otro (1/11, 9 %) había obtenido un título de maestría.

Los entrevistados mencionaron desempeñarse en diferentes roles relacionados con el aseguramiento de la calidad en software. Solo un participante (1/11, 9 %) ocupaba el puesto de Líder de Aseguramiento de Calidad (QA Lead). Por otro lado, dos entrevistados (2/11, 18 %) se desempeñan como Encargados de Desarrollo de Software (Head of Development), dos (2/11, 18 %) como Gerentes de Proyectos, y cuatro (4/11, 36 %) como analistas de aseguramiento de calidad senior (Senior QA). Este último grupo indica una alta prioridad en los equipos respecto a los procesos de calidad.

#### **4.2.2 Análisis de datos**

En el análisis de resultados, se presenta un resumen de los datos obtenidos a través de las preguntas de entrevista sobre la comunicación, colaboración y eficiencia en equipos reducidos de calidad.

**Tabla 2***Datos de entrevista*

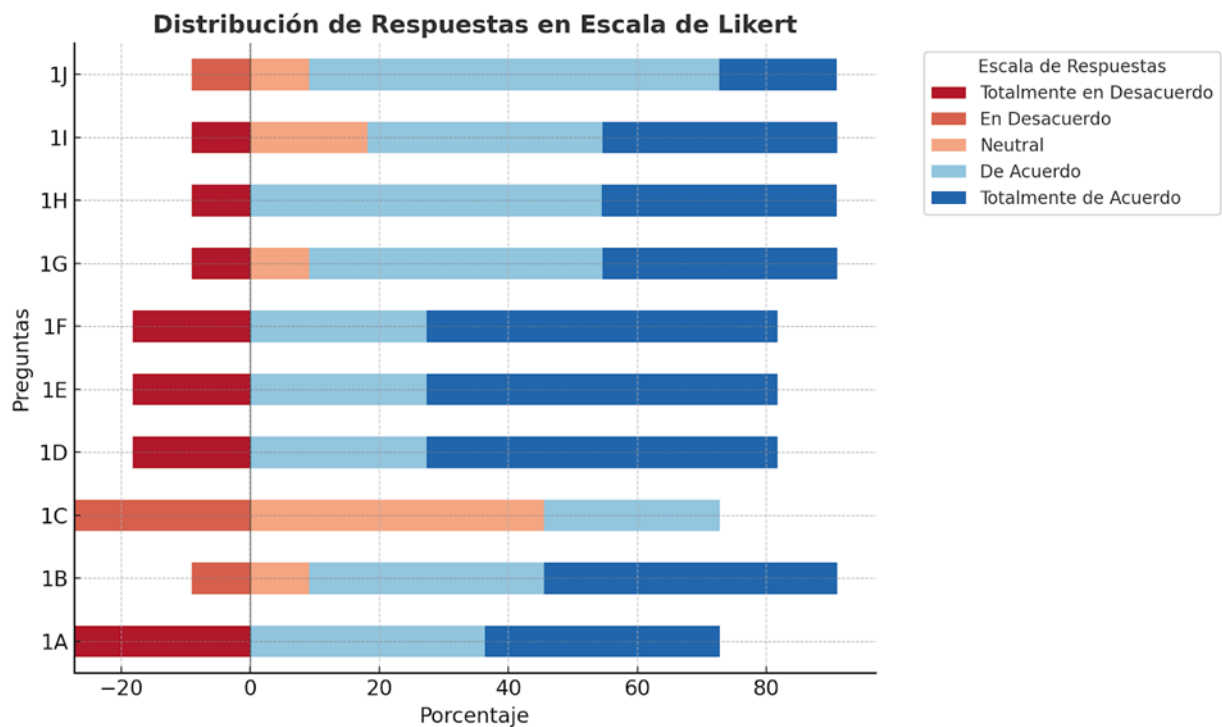
<b>Referencia gráfica</b>	<b>Totalmente en desacuerdo</b>	<b>En desacuerdo</b>	<b>Neut ral</b>	<b>De acuerdo</b>	<b>Totalmente de acuerdo</b>
1A	3 (27.3%)	0 (0.0%)	0 (0.0%)	4 (36.4%)	4 (36.4%)
1B	0 (0.0%)	1 (9.1%)	1 (9.1%)	4 (36.4%)	5 (45.5%)
1C	0 (0.0%)	3 (27.3%)	5 (45.5%)	3 (27.3%)	0 (0.0%)
1D	2 (18.2%)	0 (0.0%)	0 (0.0%)	3 (27.3%)	6 (54.5%)
1E	2 (18.2%)	0 (0.0%)	0 (0.0%)	3 (27.3%)	6 (54.5%)
1F	2 (18.2%)	0 (0.0%)	0 (0.0%)	3 (27.3%)	6 (54.5%)
1G	1 (9.1%)	0 (0.0%)	1 (9.1%)	5 (45.5%)	4 (36.4%)
1H	1 (9.1%)	0 (0.0%)	0 (0.0%)	6 (54.5%)	4 (36.4%)
1I	1 (9.1%)	0 (0.0%)	2 (18.2%)	4 (36.4%)	4 (36.4%)
1J	0 (0.0%)	1 (9.1%)	1 (9.1%)	7 (63.6%)	2 (18.2%)

A continuación, se presentan los datos de la encuesta sobre los niveles de acuerdo respecto a la efectividad y distribución de roles en equipos reducidos de aseguramiento de calidad. Cada

referencia gráfica (1A, 1B, etc.) representa un aspecto específico del proceso de asignación y colaboración en QA, con los porcentajes detallados para cada nivel de respuesta: desde "Totalmente en desacuerdo" hasta "Totalmente de acuerdo".

**Figura 1**

*Gráfico de distribución de respuestas en Escala de Likert*



#### 4.2.2.1 Eficiencia de los equipos reducidos de calidad de software

En el contexto de la investigación, la pregunta 1A cuestionó a los entrevistados sobre su percepción respecto a la eficiencia de los equipos reducidos, específicamente sobre su capacidad para detectar y solucionar defectos, a pesar de contar con un número limitado de miembros en el equipo; los datos obtenidos revelan una distribución equilibrada entre los entrevistados, puesto que del total un 36.4 % de los participantes estuvo totalmente de acuerdo, mientras que otro 36.4 % estuvo de acuerdo con la afirmación, surgiendo que una

mayoría percibe a los equipos reducidos como eficientes en la detección y solución de defectos. No obstante, un 27.3 % de los entrevistados se mostró neutral, lo que indica una falta de consenso o una evaluación más cautelosa sobre la efectividad de estos equipos en la práctica.

Es relevante indicar que, a pesar de esta diversidad de opiniones, prevalece una percepción positiva, ya que se justifica bajo el hecho de que los equipos reducidos destacan en temas como la comunicación y la agilidad de procesos, lo cual concuerda con la idea propuesta por Bumbuc (2024), quien afirma que en estos equipos, la comunicación es más directa, ya que existen menos niveles jerárquicos, lo que reduce parcialmente la distorsión y falta de comunicación. Asimismo, Ferrarini y Enzo (2023) enfatizan que esta eficiencia es evidente, ya que se traduce en una mejora significativa en el flujo de información, subrayando con ello la efectividad de los equipos reducidos en la identificación y resolución de problemas de una manera más dinámica, lo cual es crucial en un entorno de desarrollo de software (Bumbuc, 2024; Chukwuranh & Aderemi 2024).

No obstante, no se puede obviar el porcentaje significativo de respuestas neutrales, que subrayan la noción de que también estos equipos pueden enfrentar desafíos, considerando la sobrecarga de tarea, las limitaciones de tiempo; esta percepción neutral es concordante con lo estipulado por Bumbuc (2024), quien sugiere que, aunque los equipos pequeños pueden ser eficaces en ciertos aspectos, la eficiencia no está garantizada, no cuando las tareas son complejas o los miembros tienen múltiples responsabilidades.

#### **4.2.2.2 *Comunicación entre equipos***

En la pregunta 1B, se evaluó la percepción de los entrevistados sobre la comunicación y colaboración en equipos reducidos de calidad. Los resultados muestran una variabilidad de

opiniones, ya que el 45.5 % de los participantes estaba totalmente de acuerdo y el 36.4 % de acuerdo con la efectividad de la comunicación. Sin embargo, un 9.1 % se mantuvo neutral y otro 9.1 % en desacuerdo.

La percepción positiva sobre la comunicación y colaboración en equipos de trabajo reducidos que reflejan los datos del estudio está en línea con lo señalado en la literatura sobre la importancia de estos aspectos para el funcionamiento eficaz de un equipo, ya que facilitan la resolución de problemas y mejoran la coordinación entre los miembros (Dietrichson et al., 2022). Esto concuerda también con la idea propuesta por Machuca et al. (2020), quienes argumentan que, a medida que aumenta el tamaño del equipo, mantener una comunicación efectiva puede volverse más desafiante, lo que afecta la productividad, reforzando la idea de que los equipos más pequeños tienden a beneficiarse de una comunicación más ágil y eficiente, como se refleja en los datos del estudio. Lo mismo, lo cual también es planteado por John et al. (2024), las colaboraciones más grandes están relacionadas con un aumento en los problemas reportados, especialmente en lo que respecta a la coordinación.

Sin embargo, la existencia de una porción de participantes 18.2 % que expresó reservas o insatisfacción sugiere que, a pesar de los beneficios reconocidos, la comunicación en estos equipos no siempre es óptima, lo cual concuerda con la postura de Cabrera et al. (2024), quienes advierten que incluso en equipos reducidos puede haber desafíos significativos en la comunicación, como malentendidos, falta de claridad en los roles y responsabilidades, o una ausencia de retroalimentación constructiva, los cuales son elementos que dentro del control de calidad son fundamentales, ya que permitirán solucionar y detectar defectos de manera oportuna (Dietrichson et al., 2022).

Por último, es fundamental reconocer que la comunicación es un proceso dinámico que trasciende la simple transmisión de información, ya que juega un papel crucial en el

crecimiento y progreso organizacional, por lo que requiere de la interacción social y la construcción de confianza entre los miembros del equipo (Vilca et al., 2021). Por lo tanto, a pesar de que los datos sugieren una mayoría satisfecha, es fundamental abordar las preocupaciones de los disidentes que expresan disconformidad para fomentar una comunicación más efectiva y colaborativa.

#### ***4.2.2.3 Influencia del tamaño de los equipos en el cumplimiento de estándares de calidad***

Respecto a la pregunta 1C, se consultó a los entrevistados sobre la consideración del tamaño reducido de los equipos y su posible impacto en la capacidad para cumplir con los estándares de calidad asignados. De los once participantes, el 45.5 % adoptó una postura neutral, sugiriendo que el tamaño del equipo puede no ser un factor determinante en el cumplimiento de estos estándares. Sin embargo, el 27.3 % estuvo de acuerdo en que el tamaño del equipo es relevante, indicando la necesidad de un equilibrio en las cargas de trabajo y tiempos de entrega. Por otro lado, un 27.3 % se mostró desacuerdo con esta afirmación, enfatizando que la calidad del trabajo individual y la experiencia de los profesionales son más importantes que la cantidad de miembros en el equipo. Esto destaca la necesidad de un enfoque balanceado en la asignación de recursos humanos para asegurar la calidad de los procesos.

Los hallazgos sugieren que la relación entre el tamaño del equipo y la eficiencia en el cumplimiento de estándares de calidad es compleja y multifacética. La postura neutral de muchos entrevistados indica que, aunque el tamaño puede tener cierta influencia, no es necesariamente un determinante de la efectividad del trabajo. Esta idea se alinea con lo planteado por Miyurangana et al. (2023), quienes argumentan que la calidad del trabajo en

equipo depende más de la dinámica interna y la cooperación entre los miembros que de la cantidad de personas involucradas.

No obstante, es importante considerar la perspectiva del 27.3% de los entrevistados que sostiene la influencia del tamaño del equipo. Esta opinión está en consonancia con las advertencias de Dietrichson et al. (2022), quienes señalan que en equipos más grandes, los problemas de coordinación y comunicación tienden a escalar, lo que puede resultar en una disminución de la eficiencia. Este análisis sugiere que, si bien un número reducido de expertos puede resultar más efectivo en la consecución de estándares de calidad, la falta de una estructura clara y la comunicación efectiva son elementos que no se pueden pasar por alto, independientemente del tamaño del equipo.

Por lo tanto, es esencial evaluar detenidamente el número de personas asignadas a las tareas para garantizar la calidad de los procesos vinculados al cumplimiento de los estándares de calidad. Aunque, según las opiniones de los profesionales, la cantidad de miembros puede no ser una limitante en este contexto, es crucial contar con una estimación adecuada del personal encargado de estas labores. Solo así se puede asegurar una correcta implementación y eficacia en la consecución de los objetivos de calidad.

#### ***4.2.2.4 Influencia del tamaño de los equipos en el cumplimiento de estándares de calidad***

En cuanto a la pregunta 1D, se consultó a los entrevistados sobre la consideración del tamaño reducido de los equipos y su posible impacto en la capacidad para cumplir con los estándares de calidad asignados. De los once participantes, el 45.5 % adoptó una postura neutral, sugiriendo que el tamaño del equipo puede no ser un factor determinante en el cumplimiento

de estos estándares. Sin embargo, el 27.3 % estuvo de acuerdo en que el tamaño del equipo es relevante, indicando la necesidad de un equilibrio en las cargas de trabajo y tiempos de entrega. Por otro lado, un 27.3 % se mostró de acuerdo con esta afirmación, enfatizando que la calidad del trabajo individual y la experiencia de los profesionales son más importantes que la cantidad de miembros en el equipo. Esto destaca la necesidad de un enfoque balanceado en la asignación de recursos humanos para asegurar la calidad de los procesos.

Los hallazgos sugieren que la relación entre el tamaño del equipo y la eficiencia en el cumplimiento de estándares de calidad es compleja y multifacética. La postura neutral de muchos entrevistados indica que, aunque el tamaño puede tener cierta influencia, no es necesariamente un determinante de la efectividad del trabajo. Esta idea se alinea con lo planteado por Miyurangana et al. (2023), quienes argumentan que la calidad del trabajo en equipo depende más de la dinámica interna y la cooperación entre los miembros que de la cantidad de personas involucradas.

No obstante, es importante considerar la perspectiva del 27.3% de los entrevistados que sostiene la influencia del tamaño del equipo. Esta opinión está en consonancia con las advertencias de Dietrichson et al. (2022), quienes señalan que en equipos más grandes, los problemas de coordinación y comunicación tienden a escalar, lo que puede resultar en una disminución de la eficiencia. Este análisis sugiere que, si bien un número reducido de expertos puede resultar más efectivo en la consecución de estándares de calidad, la falta de una estructura clara y la comunicación efectiva son elementos que no se pueden pasar por alto, independientemente del tamaño del equipo.

Por lo tanto, es esencial evaluar detenidamente el número de personas asignadas a las tareas para garantizar la calidad de los procesos vinculados al cumplimiento de los estándares de calidad. Aunque, según las opiniones de los profesionales, la cantidad de miembros puede

no ser una limitante en este contexto, es crucial contar con una estimación adecuada del personal encargado de estas labores. Solo así se puede asegurar una correcta implementación y eficacia en la consecución de los objetivos de calidad.

#### ***4.2.2.5 Importancia del personal capacitado para la detección de errores***

En este apartado, se exploró, mediante la pregunta 1E, la percepción de los entrevistados respecto a la importancia de contar con personal capacitado para la detección de errores en el desarrollo de sistemas. Los resultados muestran que un 54.5 % de los encuestados considera fundamental disponer de profesionales con conocimientos técnicos especializados en equipos de calidad de software reducidos. Este hallazgo pone de relieve la necesidad, para las empresas en este sector, de incorporar personal experto que garantice una detección temprana y precisa de errores a lo largo del ciclo de desarrollo del software, contribuyendo así a la calidad y eficiencia en los procesos de aseguramiento de calidad.

Con esto se evidencia que la experiencia en detección y diagnóstico de errores es fundamental para sacar el máximo provecho de las herramientas automáticas, lo que ayuda a maximizar la calidad del producto y reducir la incidencia de defectos en el código fuente. TechComm (2020) subraya la relevancia de métodos de capacitación prácticos, como el "entrenamiento con ruedas", que han demostrado ser efectivos en la mejora de la detección y resolución de errores. La implementación de estrategias de formación adecuadas permite un desarrollo de habilidades en los profesionales, que se traduce en una mayor competencia y agilidad en los procesos de aseguramiento de calidad.

Un 27.3 % de los entrevistados expresó también su acuerdo con la necesidad de experiencia y competencia técnica. No obstante, esta postura reconoce implícitamente la importancia de

una formación continua para los miembros del equipo, dado que se desenvuelven en un entorno en constante cambio, en el que las metodologías, herramientas y lenguajes de programación evolucionan con rapidez. Este punto refuerza la idea de que el aprendizaje y la adaptación son esenciales para que el equipo mantenga una alta eficiencia y precisión en la detección de errores.

Por otro lado, un 18.2 % de los encuestados no considera que la capacitación avanzada sea estrictamente necesaria para la detección de errores. Este enfoque crítico sugiere la posibilidad de que un equipo pueda operar eficazmente sin una base técnica sólida, lo cual contrasta con la perspectiva de MIT Press (2024). Este autor argumenta que la dependencia excesiva en herramientas de detección automáticas puede limitar las habilidades de evaluación y juicio humano, esenciales en escenarios complejos que requieren una toma de decisiones informada y personalizada. Este contraste evidencia un posible riesgo en la automatización y la formación técnica limitada, ya que la falta de conocimientos técnicos en los equipos podría resultar en dependencias excesivas que, aunque eficientes en el corto plazo, pueden poner en peligro la calidad y robustez del software a largo plazo.

La diversidad destaca un dilema clave entre la capacitación y la eficiencia. Aunque es crucial que los equipos sean ágiles y capaces de adaptarse rápidamente a las demandas cambiantes del mercado, este objetivo no debe comprometer el conocimiento profundo y técnico necesario para garantizar un desarrollo de calidad. En conclusión, esta discusión sugiere que una combinación equilibrada de personal capacitado y herramientas automáticas, junto con una cultura de aprendizaje continuo, es fundamental para lograr una mejora sostenida en los procesos de prueba y control de calidad en el desarrollo de software.

#### **4.2.2.6 Relevancia de experiencia previa para la detección de errores**

En el marco de esta investigación, se consultó al grupo de expertos participantes mediante la pregunta 1F sobre la importancia de la experiencia previa en el ámbito de QA para el desempeño efectivo de los equipos reducidos. Los resultados revelan que el 54.5 % de los participantes afirma estar totalmente de acuerdo con esta afirmación, mientras que el 27.3 % se declara de acuerdo, reconociendo así que la experiencia previa en equipos limitados puede ser determinante. Contar con miembros que hayan enfrentado situaciones similares parece contribuir a la resolución más eficiente y efectiva de procesos, permitiéndoles aplicar herramientas y estrategias previamente validadas.

Este hallazgo concuerda con la propuesta de Boukhelif et al. (2023), quienes sostienen que la experiencia previa es fundamental para la detección de errores, ya que quienes tienen un bagaje en esta área pueden identificar defectos con mayor precisión, lo que reduce tanto el costo como el tiempo de corrección de errores durante todo el ciclo de desarrollo. Además, Cabable y Troilo (2021) afirman que el personal experimentado aporta un valor significativo al equipo, contribuyendo en múltiples áreas y enfrentando desafíos de forma proactiva, lo que optimiza el desempeño colectivo.

Sin embargo, un 18.2 % de los encuestados expresa desacuerdo con la relevancia de la experiencia previa, lo que sugiere una falta de consenso respecto al papel que juega la experiencia en QA. Esta disparidad podría ser reflejo de trayectorias profesionales diversas y de expectativas diferenciadas respecto al desempeño en el trabajo. En este sentido, la observación de Howden (2011) resulta pertinente, ya que enfatiza que, a pesar de contar con profesionales altamente capacitados, estos también son susceptibles a incurrir en errores. Además, señala que las empresas de desarrollo de software no tienen la posibilidad de

capacitar a sus colaboradores de manera que piensen y actúen como expertos en la materia. Aquellos que no valoran la experiencia como un factor determinante pueden estar argumentando desde la perspectiva de que un enfoque fresco y nuevas ideas pueden ser igualmente, si no más, valiosos en un entorno tan cambiante como el del desarrollo de software.

Es importante reconocer que la percepción de la experiencia previa no puede abordarse de manera unidimensional; la relevancia de la experiencia depende de otros factores, como la dinámica del equipo, la cultura organizacional y el contexto específico de desarrollo. La diversidad de opiniones indica que, aunque la experiencia es un activo valioso, no debería considerarse el único criterio para evaluar la efectividad en equipos de QA.

#### ***4.2.2.7 El personal experimentado contribuye a la mejora de procesos de prueba***

La experiencia previa del personal en el ámbito del aseguramiento de la calidad es un factor crucial para el éxito de los procesos de pruebas en la organización. En este punto se consultó a los entrevistados mediante la pregunta 1G, donde se observó que un 54.5 % del total de entrevistados respondió de manera positiva al estar totalmente de acuerdo con esta afirmación, reconociendo que este es un factor clave en el desempeño en la mejora de procesos de prueba, considerando que al contar con un número reducido de integrantes, la necesidad de brindar procesos de calidad y el personal debe de contar con experiencia profesional ayuda notablemente al proceso en general de calidad, argumentando que la experiencia en equipos de QA argumentando que la experiencia permite a los profesionales detectar y corregir errores de manera más ágil, lo cual es vital en equipos reducidos donde cada integrante tiene un impacto significativo en la calidad del proceso. Este respaldo es consistente con estudios como el de Álvarez (2024), quien asegura que la experiencia no solo

permite una detección temprana de errores, sino que también reduce el tiempo y costo de correcciones en etapas avanzadas del desarrollo, lo cual es crucial para la competitividad en un mercado altamente exigente.

Por otro lado, un 27.3 % de los encuestados expresa acuerdo, aunque con matices, destacando que la experiencia profesional en QA aporta conocimiento acumulado que optimiza tanto la eficiencia como la eficacia de los procesos. Pariona & Vargas (2024) respaldan esta visión, señalando que el conocimiento experto permite prever puntos críticos y gestionar mejor los riesgos, lo cual resulta fundamental para la mejora continua en proyectos con alta complejidad. Sin embargo, como advierte Álvarez (2024), el énfasis exclusivo en la experiencia puede conducir a una visión rígida que impida innovaciones necesarias en la práctica de QA, sugiriendo que un equipo debe equilibrar la experiencia con una apertura a nuevas metodologías y tecnologías que evolucionan constantemente.

Sin embargo, un 18.2 % de los entrevistados se mostró en desacuerdo con la necesidad de experiencia previa como factor indispensable para la mejora de los procesos de prueba. Este grupo plantea una visión crítica, sugiriendo que un equipo sin altos niveles de experiencia podría funcionar de manera eficaz si se le proporcionan las herramientas y metodologías adecuadas. De acuerdo con MIT Press (2024), los enfoques basados en la automatización y el aprendizaje asistido pueden, en algunos casos, sustituir parcialmente la experiencia humana, permitiendo que profesionales menos experimentados alcancen niveles de desempeño similares a los de sus pares con mayor recorrido. No obstante, esta perspectiva también presenta riesgos; por ejemplo, la dependencia de la automatización puede limitar el criterio humano en la toma de decisiones críticas, lo cual podría repercutir en la calidad final del producto en escenarios complejos.

#### **4.2.2.8 Dependencia en la adaptación de nuevas herramientas y tecnologías**

La pregunta 1H buscaba analizar la capacidad de adaptarse a nuevas herramientas y metodologías, la cual es crucial para evaluar la eficiencia y efectividad de los equipos de los QA. En un sondeo realizado sobre esta cuestión, el 45.5 % de los encuestados se mostró de acuerdo, mientras que el 36.4 % estuvo totalmente de acuerdo con la afirmación de que la optimización de procesos depende en gran medida de esta adaptabilidad. Sin embargo, un 9.1 % se declaró neutral y otro 9.1 % expresó su desacuerdo, lo que plantea dudas sobre la generalización de esta tendencia y sugiere una diversidad de opiniones respecto a la importancia de la adaptación en este contexto.

La mayoría de los encuestados (81.9 %) coincide en la importancia de esta adaptabilidad, lo cual sugiere que existe un consenso amplio sobre la necesidad de mantenerse al día con las tecnologías emergentes. Esta postura encuentra respaldo en la literatura; autores como Álvarez (2024) destacan que la implementación ágil de nuevas herramientas permite a los equipos pequeños responder con mayor eficacia a los desafíos cambiantes del mercado y a las crecientes demandas de calidad en el desarrollo de software.

No obstante, esta postura mayoritaria no está exenta de críticas. Un 9.1 % de los encuestados expresó desacuerdo, mientras que otro 9.1 % se mostró neutral, lo que indica una diversidad de opiniones respecto a la dependencia de herramientas tecnológicas. Esta discrepancia sugiere que el entusiasmo por la rápida adaptación a nuevas tecnologías podría acarrear problemas de eficiencia y limitar la flexibilidad de los equipos a largo plazo. Pargaonkar (2021) respalda esta preocupación al advertir que la dependencia excesiva de herramientas modernas puede resultar en la pérdida de capacidad crítica y habilidades fundamentales dentro de los equipos, convirtiéndolos en meros usuarios de tecnología en lugar de

innovadores activos en el proceso de calidad. Esta dependencia podría complicar aún más la cobertura de pruebas exhaustivas y la identificación de puntos de falla potenciales en sistemas complejos, lo que pone en riesgo la integridad del producto final.

Además, el hecho de que una minoría de los encuestados cuestione la adaptación frecuente a nuevas tecnologías sugiere que ciertos equipos podrían priorizar la experiencia y la estabilidad de herramientas probadas en lugar de adoptar cada novedad tecnológica.

Si bien la adaptación a nuevas herramientas y metodologías es vista como una necesidad en el ámbito de la calidad de software, la implementación de esta adaptación debe ser selectiva y contextual, ya que la evidencia sugiere que un enfoque equilibrado, que valore tanto la experiencia consolidada como la capacidad de innovación, puede ser más efectivo que una dependencia ciega a cada novedad tecnológica (Pargaonkar, 2021; Álvarez, 2024).

#### ***4.2.2.9 Asignación de roles y su efectividad para la optimización de procesos de control de calidad***

La asignación de roles dentro de un equipo reducido de aseguramiento de calidad (QA) es un tema crucial para la optimización de los procesos de control de calidad. Según los resultados obtenidos, un 54.5 % de los encuestados está de acuerdo con la afirmación de que la asignación de roles es efectiva para optimizar estos procesos, mientras que un 36.4 % se manifiesta totalmente de acuerdo. Por otro lado, un 9.1 % expresa un desacuerdo total. Estos datos evidencian una tendencia mayoritaria que respalda la idea de que una correcta asignación de roles puede impactar positivamente en la efectividad de los procesos de control de calidad.

La percepción general de que la asignación de roles contribuye a la optimización de los procesos de control de calidad sugiere que, en equipos reducidos, la claridad en las responsabilidades individuales puede reducir la confusión y mejorar la comunicación. Este punto es respaldado por Robles et al. (2023), quien señala que una asignación clara de roles en los equipos permite no solo una mejor distribución de las tareas, sino también un uso más eficaz de las habilidades y competencias de cada miembro. Lo anterior es crucial, ya que con recursos limitados, la eficiencia es vital; además, la claridad en los roles contribuye a la satisfacción laboral y a la motivación de los empleados, lo que a su vez puede conducir a una mayor productividad, ya que puede fomentar un sentido de pertenencia y responsabilidad, elementos que son esenciales para la moral del equipo y la efectividad del proceso de aseguramiento de calidad.

No obstante, el 9.1 % de los encuestados que se manifestaron en desacuerdo plantea interrogantes sobre la universalidad de la efectividad de esta práctica, y esta discrepancia puede estar vinculada a la resistencia al cambio o a la experiencia previa de los participantes en la implementación de metodologías ágiles.

La diversidad de opiniones observadas en la encuesta resalta la importancia de adoptar un enfoque crítico respecto a la asignación de roles en equipos de QA; mientras que la mayoría aboga por la claridad en los roles, el desacuerdo de una parte del grupo podría sugerir la necesidad de explorar formas alternativas de estructurar el trabajo en equipo, lo cual es particularmente relevante en contextos donde la innovación y la adaptación son esenciales.

#### ***4.2.2.10 Automatización de pruebas como compensación de recurso humano***

La automatización de pruebas es un recurso vital para el funcionamiento de los equipos de calidad de software, especialmente en aquellos con personal reducido; para ello, se evaluó mediante la pregunta 11 a los entrevistados sobre ello. Las respuestas de los entrevistados revelaron que un 36.4 % de la muestra expresó un total acuerdo con esta afirmación, mientras que un 36.4 % se mostró de acuerdo, lo que sugiere que la implementación de pruebas dentro de los proyectos es altamente beneficiosa para optimizar los procesos y la detección de errores, lo cual concuerda con lo mencionado por Sindhu (2019), quien enfatiza que las pruebas permiten que ciertos procesos, que anteriormente requerían de una intervención humana, sean ejecutados de manera automática, lo que permite a los profesionales de QA enfocarse en tareas de mayor valor agregado. Además, las pruebas automatizadas no solo reducen el esfuerzo manual, sino que también permiten a los equipos de calidad de software generar pruebas exhaustivas y repetitivas de manera más eficiente (IEEE, 2019).

Asimismo, se encontró que un 18.2 % de los encuestados adoptó una postura neutral, lo cual sugiere una falta de conocimiento respecto a los procesos que se ejecutan durante un ciclo de automatización de pruebas. Además, es importante destacar que un 9.1 % indicó estar en desacuerdo, lo que subraya que este tipo de pruebas demanda conocimientos técnicos especializados para su correcta implementación. Esto implica que las pruebas deben estar en contraste con lo señalado por Shin et al. (2011), quienes afirman que la automatización de prueba reduce las cargas de trabajos manuales. Su eficiencia depende en gran medida de configuraciones y revisiones realizadas por humanos para garantizar su funcionamiento adecuado.

#### ***4.2.2.11 Satisfacción en los procesos de control de calidad en su equipo de trabajo reducido***

Como parte del análisis, la pregunta 1J buscaba analizar la satisfacción de los equipos de calidad de software en el contexto individual de su funcionamiento. Los resultados obtenidos indican que la percepción general es positiva, ya que del total de encuestados, un 18.2 % manifestó estar totalmente de acuerdo con este aspecto, mientras que un 63.6 % expresó estar de acuerdo. Esta tendencia sugiere que, a pesar de tener número limitado de integrantes, estos logran cumplir con los procesos de calidad establecidos dentro de la empresa.

No obstante, es crucial considerar que el 9.1 % de los encuestados adoptó una postura neutral; puede interpretarse como una señal de cautela en el funcionamiento de estos equipos; y esta neutralidad refleja una falta de información o una percepción ambivalente sobre la eficacia de los procesos implementados.

Por último, la encuesta también evidenció que un 9.1 % de los entrevistados está totalmente en desacuerdo con la aseveración, evidenciando que el trabajo dentro de estos equipos reducidos no es la mejor opción para garantizar la calidad de los procesos del software.

Es esencial indicar que, aunque existe una disparidad de opciones respecto al grado de satisfacción, persiste una tendencia general que refleja una percepción positiva sobre el desempeño de los equipos de QA. Esta dualidad sugiere que, si bien algunos miembros pueden presentar reservas ante ellos, la mayoría valora positivamente la capacidad de estos equipos para cumplir con los objetivos, a pesar de los desafíos asociados a su tamaño reducido y tiempos de entrega.

### **4.3 Casos de estudio**

Se desarrolló un estudio de caso en una empresa de desarrollo de software con el objetivo de examinar el funcionamiento de los equipos de calidad (QA), analizando su estructura, cargas de trabajo y estrategias de mejora continua. Además, se investigó cómo estos equipos gestionan su participación en múltiples proyectos de manera simultánea.

#### ***4.3.1. Contexto general del caso de estudio: Descripción de la empresa***

Este caso de estudio se centra en Golabs Tech, una destacada empresa costarricense especializada en el desarrollo de software. Situada en la zona norte del país, específicamente en la provincia de Alajuela, en Ciudad Quesada, Golabs Tech.

De acuerdo con Golabs Tech (2024), cuenta con más de 10 años de experiencia y se dedica a proporcionar soluciones tecnológicas innovadoras y personalizadas a empresas en Estados Unidos y Canadá. Su modelo de negocio se centra en el outsourcing de servicios de software, permitiendo a sus clientes acceder a un equipo altamente capacitado sin necesidad de incurrir en los costos asociados con el reclutamiento y la gestión de personal interno.

#### ***4.3.2. Objetivo del caso de estudio: Justificación de la elección de la empresa***

Golabs Tech fue seleccionada para este caso de estudio debido a su perfil como empresa nacional y su enfoque en mercados internacionales. En particular, se destaca el análisis detallado de su departamento de calidad de software, ya que cuenta con un equipo reducido de profesionales dedicados a la calidad, lo cual la vuelve altamente atractiva para evaluar y optimizar los procesos de control de calidad.

Además, Golabs Tech dispone de métricas significativas que respaldan su enfoque en la calidad y ofrece un entorno propicio para implementar estrategias de mejora en este ámbito, lo que optimiza la eficiencia operativa y fomenta la innovación dentro del equipo de QA.

Asimismo, uno de los puntos significativos que justifican la elección responde a su capacidad de adaptación de metodologías y herramientas a las exigencias específicas de sus clientes, lo que resalta en flexibilidad y compromiso en el tema de satisfacción al cliente.

#### ***4.3.3. Descripción general de Golabs Tech***

Como se mencionó con anterioridad, Golabs Tech es una empresa que se especializa en servicios de outsourcing para el mercado norteamericano. Actualmente cuenta con un equipo aproximado de 80 colaboradores distribuidos en diversas regiones de Latinoamérica. La empresa ofrece tres principales servicios, los cuales se detallan a continuación:

**a) Staff augmentation:** Como primer servicio, Golabs Tech tiene como objetivo proporcionar personal altamente calificado que se integre de manera efectiva a los equipos de trabajo de los clientes, brindando apoyo en las tareas diarias.

**b) Time and materials:** En este segundo servicio, Golabs Tech ofrece una mano de obra a los clientes a través de contratos basados en horas de trabajo, lo cual resulta ventajoso en muchos escenarios donde las empresas requieren asistencia puntual y específica en diferentes áreas, permitiendo una adaptación ágil a las necesidades particulares de cada proyecto.

**c) Custom development:** En el marco del tercer servicio, la empresa se dedica a desarrollar productos a medida, colaborando con startups para la creación de versiones iniciales de productos que, en caso de éxito, evolucionan hacia contratos más amplios.

#### **4.3.4. Equipos de QA: Roles, tamaños y responsabilidades**

En el contexto de la compañía, se ha determinado que su departamento de calidad de software está constituido por un total de cinco profesionales, quienes están clasificados en tres roles principales. Estos roles, a su vez, se subdividen en tres niveles específicos. Los roles identificados son: Ingenieros de Calidad de Software Junior, Ingenieros de Calidad de Software Mid e Ingenieros de Calidad de Software Senior. Cabe destacar que para cada uno de estos niveles existen tres subniveles; por ejemplo, dentro del nivel Junior se encuentran los puestos de Junior I, Junior II y Junior III, los cuales deben alcanzarse antes de progresar al nivel de Ingeniero de Calidad de Software Mid, y de manera similar para los niveles Mid y Senior. Este enfoque no solo garantiza un desarrollo progresivo de competencias técnicas, sino que también fomenta un ambiente de trabajo que prioriza el aprendizaje continuo y la mejora de habilidades, lo cual concuerda con la postura de Machuca et al. (2020), quien afirma que la estructura es fundamental para la especialización en QA, permitiendo a los profesionales adquirir experiencia de manera efectiva.

Adicionalmente, en el marco de este caso de estudio, se han delineado las principales responsabilidades que cumplen estos profesionales, que se detallan en los siguientes apartados: Gestión de casos de prueba (Test Case Management), Planificación de pruebas (Test Planning), Automatización de pruebas (Web y APIs) y Control de calidad de datos (Data Quality).

Como se observa, la estructura del departamento de QA en Golabs Tech no solo refuerza la importancia de cada rol dentro del proceso de aseguramiento de la calidad, sino que también destaca la necesidad de un enfoque sistemático y bien definido para abordar los desafíos en la calidad del software.

#### ***4.3.5. Procesos de QA: Metodologías y herramientas***

En este apartado se aborda el enfoque que adopta la empresa en relación con las metodologías y herramientas de aseguramiento de la calidad (QA). De acuerdo con los datos obtenidos, Golabs Tech adapta tanto las metodologías como las herramientas a los requerimientos específicos de sus clientes. Sin embargo, cuando los clientes no cuentan con las herramientas necesarias, la empresa colabora en su implementación, siguiendo metodologías ágiles como BDD (Behavior Driven Development), TDD (Test Driven Development), CI/CD (Continuous Integration/Continuous Deployment), Scrum y Kanban. Esta flexibilidad en la adaptación de metodologías y herramientas permite a Golabs Tech satisfacer las necesidades particulares de cada cliente, garantizando así la efectividad de los procesos de QA.

#### ***4.3.6. Desafíos previos de la compañía***

En este apartado se analizan los principales desafíos que enfrenta Golabs Tech en la actualidad. Un reto significativo es la dificultad para encontrar profesionales altamente cualificados en el ámbito del aseguramiento de la calidad, particularmente en roles especializados como Senior QA Automation, los cuales requieren de experiencia en el manejo de datos; lo significativo de esta problemática es que ha sido un obstáculo persistente para la compañía, dado que la demanda de profesionales con estas competencias supera considerablemente la oferta disponible en el mercado laboral local.

Aunado a ello, la escasez de talento en este sector no solo limita la capacidad de la compañía para satisfacer la demanda, sino que también obstaculiza su crecimiento potencial y, por ende, su competitividad en un entorno empresarial, que cada día se vuelve más exigente.

#### ***4.3.7. Revisión de la documentación***

La revisión documental realizada en Golabs Tech ha revelado que la disponibilidad de documentación es limitada y se centra en un Career Path estándar (Ver Anexo 3). Este enfoque sugiere que, aunque existen directrices para el desarrollo profesional de los colaboradores, carece de soporte documental adicional que respalde de manera más integral los procesos y procedimientos realizados por la compañía.

Lo relevante de la carencia de documentación tiene implicaciones significativas, ya que puede obstaculizar la efectividad de la formación y el desarrollo personal. Asimismo, dificulta la comprensión de mejores prácticas, así como la transferencia de conocimientos entre los miembros del equipo.

#### ***4.3.8. Flujo de trabajo y actividades diarias***

En lo que respecta al flujo de trabajo y las actividades diarias que se realizan en Golabs Tech, se ha observado que las operaciones se gestionan de manera dinámica en función de las demandas específicas de los clientes. La empresa utiliza una plataforma interna denominada Gacela, que permite a cada colaborador registrar su trabajo de forma semanal, lo cual no solo facilita el mapeo de las actividades y el desempeño individual de los empleados, sino que también contribuye a una gestión más eficiente de los recursos y del tiempo.

El enfoque centrado en las necesidades del cliente asegura que las prioridades de trabajo estén alineadas con los requerimientos del mercado, lo que permite a Golabs Tech responder de manera ágil y efectiva a las exigencias de sus clientes. Además, la capacidad de rastrear el desempeño de cada colaborador a través de la plataforma interna proporciona información

valiosa que puede ser utilizada para la evaluación del rendimiento, la identificación de áreas de mejora y el desarrollo profesional de los empleados.

Cuando Golabs Tech colabora con clientes externos, también implementa otras herramientas de gestión de proyectos, como ClickUp, Jira y Asana, que permiten una planificación más detallada y un seguimiento en tiempo real de las tareas asignadas, demostrando con ello la versatilidad del equipo y su compromiso con la excelencia en la calidad del servicio ofrecido.

#### ***4.3.9. Planes de capacitación***

Golabs Tech, además de la estructura de Career Path estándar, se ha implementado un enfoque proactivo hacia la capacitación mediante la oferta de cursos bajo demanda en áreas técnicas relacionadas con la calidad de software. Para este propósito, la empresa colabora con plataformas reconocidas como lo es Coursera, con el objetivo de fomentar el desarrollo de competencias específicas entre sus colaboradores.

Cómo se logra observar mediante el análisis realizado, Golabs Tech ha logrado gestionar eficazmente su equipo pequeño de QA mediante un enfoque estructurado y estratégico, ya que logra optimizar los recursos al implementar metodologías ágiles, automatización y capacitación de sus miembros, lo que le permite mantener un estándar de calidad en los procesos, sin necesidad de contar con un gran número de miembros en sus equipos de calidad de software. Esta gestión flexible y adaptativa maximiza la eficiencia y asegura la satisfacción de sus clientes.

#### **4.4 Evaluaciones pares**

Por último, la investigación fue sometida a un proceso de evaluación por pares con el fin de validar los resultados obtenidos y medir el impacto de la propuesta de optimización de procesos de control de calidad en empresas de desarrollo de software con equipos reducidos. La prueba de revisión fue enviada a tres expertos en control de calidad, quienes poseían el grado de maestría en la disciplina, lo que asegura un alto nivel de conocimiento y experiencia en el tema tratado.

El propósito de este proceso fue evaluar la propuesta y proporcionar retroalimentación constructiva para mejorar la calidad del documento. La evaluación se centró en aspectos clave, tales como la efectividad de la propuesta para el control de calidad, la satisfacción del equipo y la eficiencia, los tiempos de entrega, la calidad del producto en general y las metodologías empleadas, ya que la intencionalidad fue optimizar la propuesta, asegurando que sea adecuada y aplicable para mejorar los procesos en equipos reducidos de control de calidad.

Para llevar a cabo la evaluación, se utilizó la escala de Likert, que va desde “totalmente en desacuerdo” hasta “totalmente de acuerdo”, permitiendo a los expertos evaluar cada aspecto de la propuesta de manera detallada, otorgando un puntaje que refleja el grado de acuerdo o desacuerdo con las afirmaciones presentadas en el documento. Además, los expertos tuvieron la posibilidad de agregar comentarios adicionales en cada sección, lo que permitió obtener retroalimentación cualitativa; las mismas se observan en el siguiente capítulo.

## **Capítulo V. Propuesta de Solución**

El siguiente capítulo muestra la solución planteada para el problema, la cual se detalla a continuación. Esta propuesta se fundamenta en la aplicación de un conjunto de técnicas y un enfoque adecuado. Inicialmente, se realizó una investigación exhaustiva de fuentes bibliográficas y se complementa con una entrevista realizada a profesionales del área de calidad de software. Posteriormente, se llevó a cabo un estudio de casos con el propósito de identificar los factores clave que influyen en los equipos de QA dentro de entornos reducidos, lo que permitió construir una propuesta en función de la optimización de procesos. Adicionalmente, se consultaron estándares de calidad, como los establecidos por la Organización Internacional de Normalización (ISO), para respaldar la formulación de dicha propuesta.

### **5.1 Propuesta de solución**

La propuesta sobre optimización de procesos de QA en equipos reducidos de empresas de desarrollo de software, Versión 1.1, se desarrolló de manera integral, pensando en la necesidad de ofrecer soluciones a equipos de QA reducidos. En la primera parte se encuentra la sección introductoria, exponiendo la importancia de estos equipos a lo largo del desarrollo del ciclo de vida del software; también se consideran los desafíos a los que se enfrentan, como la limitación de recursos, la organización y distribución de tareas.

Es fundamental enfatizar que el propósito primordial de estos equipos es gestionar de manera eficiente la calidad del software, garantizando que cumpla con los requisitos técnicos establecidos y las expectativas del usuario final, asegurando así la fiabilidad y el rendimiento del producto.

A continuación, y partiendo de las definiciones y objetivos clave de la propuesta, se han planificado siete puntos estratégicos que constituyen la base para la optimización de los procesos de QA en equipos reducidos. En primer lugar, se aborda la estrategia general de

optimización, que tiene en cuenta los roles clave dentro de un equipo de QA, como senior de QA, QA Automation Engineer y los testers, así como la asignación eficiente de tareas y responsabilidades. A través de una estructura organizada y flexible, se busca maximizar el rendimiento del equipo sin sobrecargar a sus miembros, promoviendo un entorno de trabajo colaborativo y altamente enfocado en los objetivos de calidad.

Seguidamente, dentro de este mismo marco, se plantean dos fases cruciales para la optimización de los procesos de QA: la planificación y el diseño de pruebas, seguidas de la ejecución y el seguimiento de pruebas.

En la fase de planificación y diseño de pruebas, se establecen los objetivos específicos a evaluar, considerando tanto los aspectos técnicos como funcionales del software. Aquí, el equipo de QA debe identificar los casos de prueba prioritarios, diseñar las estrategias de prueba adecuadas, gestionar los recursos y seleccionar las herramientas necesarias. Esta etapa es esencial para garantizar que las pruebas cubran los escenarios más relevantes y los posibles riesgos del producto, optimizando el tiempo y los recursos disponibles en equipos reducidos. Una vez definida la estrategia de pruebas, se pasa a la fase de ejecución y seguimiento de pruebas, donde el equipo lleva a cabo las pruebas de acuerdo con los casos diseñados. Para ello, se emplean herramientas especializadas que facilitan la gestión, ejecución y monitoreo de las pruebas, como TestRail y Jira. Estas herramientas permiten realizar un seguimiento en tiempo real del estado de las pruebas, identificando errores o fallos y asegurando que se cumplan los requisitos técnicos y funcionales del producto. Además, el seguimiento continuo permite detectar posibles áreas de mejora y ajustarlas de forma ágil, lo cual es particularmente relevante en equipos reducidos que necesitan adaptarse rápidamente a cambios y optimizar sus esfuerzos.

Seguidamente, como cuarto componente de la propuesta, se abordan las metodologías ágiles, como Scrum o Kanban, que facilitan la adaptación a cambios y el seguimiento de tareas, promoviendo ciclos de trabajo iterativos y enfoques continuos. Posteriormente, se presentan los protocolos de QA, los cuales se dividen en dos áreas claves: ejecución de pruebas y protocolos de validación y aprobación; la intencionalidad fue aportar directrices claras y sistemáticas para ejecutar pruebas.

Un aspecto adicional fundamental de la propuesta es el análisis de los beneficios económicos de un departamento de QA, resaltando cómo este contribuye a una mayor satisfacción del cliente. Otro punto relevante es la importancia y colaboración interdepartamental en QA, donde se plantea que la colaboración continua facilita una comunicación fluida, el intercambio de información y la toma de decisiones.

En cuanto al tercer apartado, contempla los elementos propios en relación con la implementación de la propuesta, detallando las frases claves del proceso, así como los cronogramas, los costos asociados y los beneficios esperados.

La propuesta concluye con la conclusión y la bibliografía, donde se enumeran las fuentes consultadas, proporcionando el respaldo académico y técnico necesario para la validez y la fundamentación de la propuesta de soluciones.

## 5.2 Procesos de prueba y resultados

La propuesta de optimización de procesos de control de QA en empresas de desarrollo de software con equipos reducidos fue sometida a una evaluación de pares, con el fin de determinar la calidad de dicha propuesta; se consultó a tres profesionales, todos en el área de calidad y desarrollo de software y con niveles académicos de máster. Respecto al primer evaluador, José Miguel Chacón Araya, Senior QA Analyst, clasificó como “totalmente de acuerdo” con la mayoría de los atributos, destacando que mejorará la comunicación entre los equipos de control de calidad. Sin embargo, fue neutral respecto a la afirmación de que la propuesta mantiene una coherencia lógica entre los diversos componentes, incluidas las metodologías. En cuanto a la sugerencia, indicó

En cuanto a las sugerencias, propuso eliminar el rol de QA Lead y reemplazarlo por un Senior QA, dado que en muchas empresas los equipos de QA suelen integrarse al equipo de desarrollo y reportar directamente al líder de estos equipos.

El segundo evaluador, Lawrence Fowks Peña, staff software engineer y docente universitario, evaluó los apartados entre “totalmente de acuerdo” y “de acuerdo”, mostrando solo neutralidad en cuanto a los ciclos de prueba. Como sugerencias, destacó la importancia de evidenciar los beneficios económicos que representa un departamento de QA dentro de las empresas de desarrollo de software, y la colaboración interdepartamental en QA, enfatizando la necesidad de una comunicación abierta no solo entre los integrantes del equipo de QA, sino también con otros equipos, como el de desarrollo.

El tercer evaluador, Alex Daniel Villegas Carranza, profesor e investigador de la Universidad Nacional (UNA) de Costa Rica, consultor e ingeniero de software y MSc. en ciberseguridad, expresó estar “totalmente de acuerdo” con los apartados claves de la propuesta, destacando

la neutralidad en que la propuesta no mejora la comunicación, ni muestra los pasos para crear un departamento de QA. Sugirió ampliar las métricas presentadas para permitir una evaluación más precisa del éxito de la propuesta, incorporando comparaciones detalladas de los indicadores clave a mejorar. En particular, propuso establecer mediciones enfocadas en la eficiencia y los tiempos de ejecución de las pruebas, tanto en la fase de ejecución como en el ciclo de pruebas en su conjunto. Además, recomendó incluir métricas de comparación entre el estado previo y posterior a la implementación de la propuesta. Por otro lado, este evaluador sugirió agregar indicadores específicos para comprender cómo la propuesta contribuirá a la mejora de la calidad del software.

En términos generales, la evaluación demostró ser positiva, pero también reveló áreas críticas que requieren ser mejoradas en una reestructuración futura de esta evaluación, ya que, a medida que la propuesta evolucione, será fundamental considerar estos comentarios para afinar los detalles y de esta forma asegurar que el modelo propuesto no solo optimice los procesos, sino que también proporcione una base sólida para evaluar su efectividad a lo largo del tiempo. A este punto es necesario indicar que las evaluaciones integras se pueden observar en los apéndices 4,5 y 6.

## **Capítulo VI. Análisis financiero**

Dentro de este capítulo, se aborda el análisis financiero del trabajo final de graduación, que busca evaluar la viabilidad económica que representa una propuesta de optimización de procesos de control de calidad en empresas de desarrollo de software con un número reducido de personas en sus equipos, con el objetivo de identificar costos, beneficios y, además, el potencial retorno de la inversión (ROI). Además, este capítulo plantea un modelo replicable para consultorías privadas, destacando la implementación de dicha propuesta en empresas con recursos limitados.

## **6.1 Análisis financiero**

El presente análisis financiero busca evaluar la viabilidad económica de la implementación de procesos optimizados para equipos de QA reducidos en empresas de desarrollo de software, con el objetivo de identificar los costos, beneficios y el potencial retorno de la inversión (ROI). Además, este análisis plantea un modelo, el cual puede ser replicado para temas de consultoras privadas, destacando su aplicabilidad en empresas con recursos limitados.

## **6.2 Costos de implementación**

### ***6.2.1 Recursos Humanos***

Para abordar el recurso necesario para llevar a cabo esta propuesta de optimización, se ha determinado la necesidad de contar con tres roles indispensables, los cuales son QA Lead, QA Automation Engineer y Manual QA Tester, y para los cuales se han estimado Los roles abarcan jornadas de 8 horas diarias 5 días a la semana.

Los siguientes costos anuales:

**Tabla 3***Costos de recurso humano del departamento de QA*

<b>Rol</b>	<b>Salario por hora</b>	<b>Costos Anual</b>	<b>Cantidad</b>	<b>Total</b>
	(₡)	(₡)		(₡)
QA Lead	₡11,700.26	₡22,477,939.00	1	₡22,477,939.00
QA Automation Engineer	₡9,248.00	₡17,757,984.00	1	₡17,757,984.00
Manual QA Tester	₡6,738.00	₡12,937,420.00	1	₡12,937,420.00
<b>Total de Recurso Humano</b>				<b>₡53,173,343.00</b>

Fuente: Los datos fueron tomados del ERI (*Economic Research Institute*)

### **6.2.2 Herramientas tecnológicas**

Además de los recursos humanos, es fundamental contar con herramientas tecnológicas que permitan gestionar y automatizar las pruebas, los planes de prueba y la administración de proyectos. Estas herramientas son esenciales para optimizar el equipo de QA y gestionar de manera eficiente todo el ciclo de vida del proyecto. Cabe destacar que estas herramientas requieren una inversión en compra, licencias o suscripciones, las cuales pueden variar en función de su modalidad (mensual o anual). Para este análisis, se han considerado los costos anuales de las siguientes herramientas:

**Tabla 4***Costos de herramientas empleadas a por el departamento de QA*

<b>Herramientas</b>	<b>Costo mensual por Usuario</b>	<b>Costo Anual por Usuario</b>	<b>Usuarios</b>	<b>Total</b>
TestRail	\$35.00	\$420.00	3	\$1,260.00
Jira	\$7.53	\$90.36	3	\$271.08
GitHub	\$21.00	\$252.00	3	\$756.00
Burp Suite	\$37.50	\$450.00	3	\$1,350.00
Asana	\$11.00	\$132.00	3	\$396.00
Google Space	\$12.00	\$144.00	3	\$432.00
Microsoft Teams	\$4.00	\$48.00	3	\$144.00
<b>Total Herramientas</b>	<b>\$128.03</b>	<b>\$1,536.36</b>		<b>\$4,609.08</b>

Como se observa en la tabla anterior, los costos totales mensuales por usuario son \$128.23. Mientras que los costos anuales para la implementación de estas en tres usuarios son de \$4.609,08.

### **6.2.3 Costos fijos**

Para este análisis financiero, no se tomarán en cuenta este tipo de gastos, ya que se parte de la hipótesis de que cada colaborador de QA inicialmente continúa trabajando de manera remota, lo cual, por supuesto, abre la posibilidad de ampliar en versiones futuras un estudio financiero que abarque estos costos de una manera en la cual tome en cuenta lo que representa el trabajo de los colaboradores en una oficina.

### **6.2.4 Capacitaciones y automatizaciones**

Otro de los rubros importantes a valorar en el costo de la propuesta es el tema de la formación continua de los equipos, ya que es esencial para asegurar que el equipo de QA se mantenga

actualizado sobre las mejores prácticas, herramientas y metodologías emergentes. Es importante indicar que el costo de capacitación incluirá el valor de la adquisición en diversas plataformas de formación, así como el costo asociado a cursos especializados en herramientas y tecnologías claves.

**Tabla 5**

*Costos de capacitaciones*

<b>Capacitaciones y Automatizaciones</b>	<b>Costos</b>
Capacitación Cruzada	€500,000
Automatización de pruebas	€1,200,000

Como se observa en la tabla anterior, los costos totales de las capacitaciones y las automatizaciones ascienden a €1,700,000 anuales. Cabe destacar que las automatizaciones implican una implementación única.

### **6.3 Ingresos esperados respecto a la implementación de la propuesta**

En relación con los ingresos generados por la implementación de la propuesta de optimización de procesos de QA en empresas de desarrollo de software con equipos reducidos, se identificaron tres aspectos clave, determinados según los beneficios asociados a los puntos a optimizar. A continuación, se detallan puntualmente los ingresos que obtienen las empresas:

#### **6.3.1 Reducción de costos por detección temprana de errores**

Según un estudio realizado por IEEE (2020), la detección temprana de defectos es un factor determinante para minimizar los costos en proyectos de desarrollo de software. La implementación de herramientas predictivas permite a las empresas identificar problemas potenciales en las etapas iniciales del desarrollo, lo que evita que estos errores lleguen a los

ambientes de producción, donde los costos asociados a su corrección serían significativamente mayores. Además, el estudio resalta que estos modelos predictivos no solo son efectivos para reducir defectos críticos en producción, sino que también optimizan el uso de recursos y mejoran la eficiencia operativa del equipo. Como resultado, estas optimizaciones generan un ahorro aproximado de \$3,800 mensuales.

### ***6.3.2 Retención de clientes***

El desarrollo de software de alta calidad tiene un impacto directo en los clientes, fomentando su fidelidad. Según la IEEE (2022), la implementación de buenas prácticas como la mejora continua no solo asegura que los productos desarrollados sean de alta calidad, sino que también incrementa la satisfacción del cliente, un factor crucial para su retención. Además, la mejora constante de los equipos mediante metodologías ágiles y herramientas predictivas influye positivamente en la percepción que los clientes tienen de la empresa. Este estudio estima que, gracias a estas prácticas, las empresas obtienen ingresos iniciales de aproximadamente \$3,000 mensuales.

### ***6.3.3 Reducción de tiempos de entrega***

La optimización de procesos a través de metodologías ágiles y la automatización de pruebas tiene un impacto positivo y significativo en la reducción de los tiempos de entrega. De acuerdo con IEEE (2020), estas estrategias permiten a las empresas acelerar los ciclos de trabajo y responder con mayor rapidez a las demandas del mercado. La correcta implementación de la integración continua, acompañada de la automatización de pruebas, reduce los tiempos de entrega y facilita la detección de errores, generando un ahorro y

ganancias aproximadas de \$5,000 mensuales debido a la mejora en la eficiencia operativa y la rapidez en los ciclos de desarrollo.

#### **6.4 Flujo de caja**

Para el desarrollo del flujo de caja, se consideraron tanto los ingresos como los egresos mensuales que tendrá una empresa que implemente la propuesta de optimización de procesos de QA en empresas de desarrollo de software con equipos reducidos. Los egresos se calcularon a partir de los salarios de los profesionales a cargo de estos procesos, que son: un QA Lead con un salario mensual de \$3,600.08, un QA Automation Engineer con un salario mensual de \$2,845.53 y un Manual QA Tester con un salario mensual de \$2,073.20. Adicionalmente, se contemplan herramientas necesarias para la propuesta, cuyo costo total es de \$384.10 mensuales, así como las capacitaciones y automatizaciones, que suman un total de \$1,346.14.

Para la sección de ingresos, se tomaron en cuenta los estudios mencionados en el apartado de este análisis, titulado “*Ingresos esperados respecto a la implementación de la propuesta*”, que enumeran los siguientes puntos:

- a) Reducción de costos por detección temprana de errores, con un total de \$3,800.
- b) Retención de clientes, que suma \$3,000 mensuales.
- c) Reducción de tiempos de entrega, con una aportación de \$5,000 mensuales.

A continuación, se detalla el flujo de caja del proyecto en cuestión. En relación con esta propuesta, se aplicó una tasa de descuento del 10 %, ya que, dentro de la industria de desarrollo de software, las tasas de descuento tienden a ser más altas debido a la volatilidad y los rápidos cambios tecnológicos. Según Damodaran (2021), una tasa entre 10 % y 15 %

podría ser adecuada para reflejar el riesgo del sector. A continuación se detalla el flujo de caja mensual:

**Tabla 6**

*Detalles del flujo de caja*

<b>Detalles de flujo de caja</b>	<b>Mes 1</b>	<b>Mes 2</b>	<b>Mes 3</b>	<b>Mes 4</b>	<b>Mes 5</b>	<b>Mes 6</b>
<b>INGRESOS</b>	\$11,800.00	\$11,800.00	\$11,800.00	\$11,800.00	\$11,800.00	\$11,800.00
<b>EGRESOS</b>	\$10,249.08	\$10,249.08	\$10,249.08	\$10,249.08	\$10,249.08	\$10,249.08
<b>Flujo Neto</b>	\$1,550.92	\$1,550.92	\$1,550.92	\$1,550.92	\$1,550.92	\$1,550.92

**6.4.1. Resultados del Valor Actual Neto (VAN) y Tasa Interna de Retorno (TIR)**

De acuerdo con el análisis del flujo de caja realizado, se obtuvieron los siguientes parámetros correspondientes al VAN y la TIR, considerando una tasa de descuento del 10 %, según las referencias aplicables en la industria de desarrollo de software.

**Tabla 7**

*Resultados del Valor Actual Neto (VAN) y Tasa Interna de Retorno (TIR)*

<b>Parámetros</b>	<b>Valor</b>
<b>VAN</b>	\$ 199.91
<b>TIR</b>	10 %

**6.4.2. Recuperación de la Inversión**

La recuperación de la inversión se refiere al tiempo que le toma a la empresa recobrar el capital invertido para implementar la propuesta. Como se observa en la siguiente tabla, al cuarto mes de implementación, la empresa estaría generando ganancias y habría cubierto las deudas asociadas a la propuesta.

**Tabla 8***Recuperación de la Inversión*

<b>Periodo</b>	<b>Mes 1</b>	<b>Mes 2</b>	<b>Mes 3</b>	<b>Mes 4</b>	<b>Mes 5</b>	<b>Mes 6</b>
Flujo Neto en efectivo	\$4,809.78	\$4,809.78	\$4,809.78	\$4,817.78	\$4,817.78	\$4,817.78
Flujo de caja acumulado	-\$15,953.10	-\$11,143.32	-\$6,333.54	\$4,819.11	\$9,636.89	\$14,454.67

Como se observa, la implementación de procesos optimizados para equipo de QA reducido en empresa de desarrollo de software presenta una oportunidad estratégica para mejorar la eficiencia operativa y la calidad del producto, a la vez que reduce costos a largo plazo. Los costos asociados con la capacitación, automatización y herramientas tecnológicas son significativos, pero proyectados, tales como la reducción de costos operativos, el aumento de la productividad y la mejora en la satisfacción del cliente, superan ampliamente la inversión inicial.

Además, la integración de metodologías ágiles y la automatización de pruebas se presentan como componentes clave para maximizar la eficiencia de los equipos de QA sin comprometer la calidad. Los beneficios económicos estimados, junto con la mejora en la colaboración y el uso más efectivo de los recursos, consolidan la viabilidad de la propuesta, proporcionando un retorno de inversión positivo y sostenible.

## **Capítulo VII. Conclusiones y recomendaciones**

## **7.1 Conclusiones**

Este trabajo final tuvo como objetivo la creación de una propuesta para la optimización de los procesos de control de calidad en empresas de desarrollo de software con equipos reducidos. Durante el proceso para la confección de dicha propuesta, se implementaron diversas técnicas para garantizar su fiabilidad.

A continuación, se presentan las conclusiones obtenidas en relación con cada uno de los objetivos planteados en la investigación.

### ***7.1.1 Conclusión general***

La propuesta de optimización de procesos de control de calidad en empresas de desarrollo de software con equipos reducidos permite mejorar los procesos de control de calidad implementados, incluso en escenarios donde los equipos de este departamento operan con recursos limitados. Integra un análisis detallado de casos de éxito previos, lo que respalda la estrategia con un marco teórico sólido. Además, esta propuesta ofrece una solución adaptada a las necesidades y restricciones específicas de este tipo de entornos, fomentando un trabajo eficiente y efectivo en los procesos de QA.

### ***7.1.2 Conclusiones específicas***

Respecto a la revisión sistemática de la literatura, esta permitió construir una base sólida que sustenta este trabajo final de graduación, permitiendo identificar y analizar los factores determinantes en la gestión de la calidad de software. A través de la exploración de estudios previos, enfoques metodológicos y estrategias efectivas, se ha logrado comprender en profundidad las dinámicas que influyen en el desempeño de los equipos de calidad, particularmente en aquellos con recursos limitados.

El análisis de la literatura ha sido fundamental para identificar las mejores prácticas aplicables a equipos de tamaño reducido, lo cual es crucial para optimizar los procesos de control de calidad sin comprometer la efectividad del trabajo. Las lecciones extraídas de los estudios revisados no solo han servido para sustentar la propuesta de mejora de este trabajo, sino que también han asegurado que dicha propuesta esté basada en conocimientos actualizados y pertinentes para la industria. En este sentido, la revisión de la literatura no solo ha proporcionado un respaldo teórico sólido, sino que también ha ofrecido valiosos lineamientos y estrategias que pueden implementarse de manera efectiva en empresas con equipos pequeños dedicados a la calidad en entornos de desarrollo de software. De esta manera, se garantiza que la propuesta no solo sea relevante, sino también viable, alineada con las necesidades actuales del sector y capaz de enfrentar los desafíos propios de estos equipos en la industria del software.

En cuanto al segundo objetivo que versaba sobre el diagnóstico situacional revisión de la gestión de equipos de calidad en entornos reducidos de desarrollo de software, a través de un diagnóstico situacional, ha permitido evidenciar las prácticas actuales y los principales desafíos que enfrentan las empresas de desarrollo de software. El análisis realizado concluyó que, a pesar de las limitaciones en recursos, estas empresas logran mantener un alto estándar de calidad mediante el uso de metodologías ágiles, herramientas de automatización y una planificación estratégica de la capacitación; pero no se puede obviar, que están también enfrentan retos como lo son las altas cargas de trabajo, la necesidad de contar con personal multifuncional y la disposición de recursos limitados, los cuales dificultan la gestión y optimización de procesos de calidad, destacando la importancia de contar con una estructura más formalizada, mejores prácticas documental y un flujo de trabajo más organizado.

En relación con el objetivo de desarrollar una propuesta de estrategia para optimizar los procesos de control de calidad en equipos reducidos, se logró integrar estrategias que aborden de manera integral las necesidades y dificultades que enfrentan estos equipos, tomando en consideración las mejores prácticas identificadas a partir de revisión literaria y el caso de estudio, considerando la integración de metodologías ágiles, la automatización de pruebas, la revisión y validación y un enfoque proactivo que permita capacitar a su personal, adaptando a las restricciones de su propio equipo, como la escasez de recursos y las altas cargas laborales. Asimismo, la propuesta presupone una serie de herramientas y plataformas que facilitan la gestión de tareas y el seguimiento del rendimiento, buscando con ello no solo mejorar la calidad del producto final, sino también incrementar la eficiencia operativa de estos equipos, reduciendo tiempos de entrega y mejorando la capacidad de respuesta ante las necesidades cambiantes del cliente. En conclusión, la propuesta tiene el potencial de generar un impacto significativo en la eficiencia y éxito de los equipos de control de calidad de tamaño reducido, al estar diseñada específicamente para mejorar la productividad, la calidad y la adaptabilidad en entornos de desarrollo de software, lo que contribuirá a la sostenibilidad y competitividad de las organizaciones.

Por último, como parte integral de este TFG, el cuarto objetivo fue someter la propuesta a una revisión de pares con el fin de depurarla mediante la retroalimentación de expertos en el área. Tras realizar la evaluación de pares, se encontró una aceptación positiva por parte de los tres expertos; sin embargo, los evaluadores sugirieron implementar ciertos cambios o mejoras con el propósito de complementar de forma más integral la propuesta, como se explicó en el capítulo V, titulado **Propuesta**.

## 7.2 Recomendaciones

En este apartado se exponen las recomendaciones finales derivadas de la aplicación de los diversos instrumentos y técnicas utilizadas durante el desarrollo de la investigación:

1. En vista de que las empresas de desarrollo de software suelen enfrentar limitaciones de recursos, se recomienda que el equipo de QA reducido sea lo más multifuncional y capacitado que se pueda en las diversas áreas, como pruebas automatizadas, metodologías ágiles y gestión de proyectos, por lo que una capacitación en los diversos enfoques y la flexibilidad de funciones se traducirá en un alto nivel de desempeño.
2. Se recomienda a las empresas promover el fortalecimiento de las capacidades técnicas de sus colaboradores a través de programas de formación continua, tanto en herramientas como en metodologías para el control de la calidad, dando énfasis a la implementación de tecnologías automatizadas, lo cual permitirá una maximización de la eficiencia y minimizará el impacto creado por la limitación de recursos humanos.
3. Se propone a las empresas que deseen implementar la propuesta planificada en el TFG la destinación de presupuesto a la capacitación continua y a la obtención de certificación por parte de sus miembros, ya que se espera que estos posean conocimientos y habilidades actualizadas, que no solo mejoren la calidad del trabajo, sino que también aseguren que el equipo esté preparado para enfrentar los desafíos que surjan durante el ciclo de vida del producto.
4. Se sugiere a los equipos de QA reducidos establecer ciclos determinados para la retroalimentación entre el equipo de QA y los desarrolladores, con la finalidad de poder identificar posibles áreas de mejora y, además, poder hacer reajustes en caso

de ser necesario. Este tipo de situaciones fortalecerá la comunicación, la cual es un factor importante para reducir errores recurrentes dentro de los ciclos de trabajo.

## **Capítulo VIII. Análisis retrospectivo**

## **8.1 Análisis retrospectivo**

El análisis retrospectivo de este TFG consiste en una revisión crítica de las limitaciones encontradas durante su desarrollo, con el fin de destacar las principales dificultades experimentadas, las cuales se relacionan tanto con factores externos como internos.

En primer lugar, una de las limitaciones más significativas fue la falta de tiempo del investigador, lo que impactó de manera directa en el desarrollo, ya que afectó ocasionalmente la posibilidad de realizar un análisis más exhaustivo.

La segunda limitación estuvo relacionada con la dificultad para obtener informantes adecuados para la recolección de datos mediante encuestas, debido a la coordinación de tiempos y la disponibilidad de los participantes, lo que restringió la cantidad de respuestas y afectó la representatividad de los datos recopilados.

La tercera limitante se relacionó con el análisis financiero, ya que la falta de una empresa con la cual implementar la propuesta dificulta la evaluación de viabilidad económica, limitando la posibilidad de determinar si la propuesta financieramente era sostenible o beneficiosa.

Otra limitación significativa estuvo vinculada a la dificultad para contar con expertos con grado académico de máster disponibles para la revisión de áreas clave de la propuesta, limitando la obtención de una retroalimentación suficientemente profunda y especializada, lo que afectó la calidad y el alcance de la validación experta.

Se insta al programa y a la Universidad Nacional a considerar los resultados de TFG como un referente de consulta para futuros investigadores que buscan explorar la implementación de propuestas de optimización de procesos de QA en equipos reducidos, ya que esta investigación puede funcionar como base para el diseño y evaluación de estos procedimientos.

## **Referencias Bibliográficas**

- Alanoca, O. W. M. (2021). Modelo de Pruebas de Regresión Automatizadas en Procesos de Integración Continua en Sistemas Web. *INF-FCPN-PGI Revista PGI*, 143-148. [https://ojs.umsa.bo/ojs/index.php/inf\\_fcpn\\_pgi/article/download/72/59](https://ojs.umsa.bo/ojs/index.php/inf_fcpn_pgi/article/download/72/59)
- Aleem, S., Capretz, L. F., & Ahmed, F. (2016). Game development software engineering process life cycle: a systematic review. *Journal of Software Engineering Research and Development*, 4, 1-30. <https://link.springer.com/content/pdf/10.1186/s40411-016-0032-7.pdf>
- Alvarado, J. C. O. (2018). El Marco Metodológico en la investigación cualitativa. Experiencia de un trabajo de tesis doctoral. *Revista Científica Estelí*, (27), 25-37. <https://revistas.unan.edu.ni/index.php/Cientifica/article/download/2256/3490>
- Álvarez Torres, H. J. (2024). *Análisis de adopción de Metodologías Ágiles (Scrum, Kanban) para mejorar la eficiencia y calidad en el desarrollo de software en la Compañía Ingeniería Integrasayox SA* (Bachelor's thesis, Babahoyo: UTB-FAFI. 2024).
- American Society for Quality. (2021). *History & Evolution of Total Quality Management (TQM) / ASQ*. <https://asq.org/quality-resources/total-quality-management/tqm-history>
- Araújo, C. A., Delamaro, M. E., Maldonado, J. C., & Vincenzi, A. M. (2016). Correlating automatic static analysis and mutation testing: towards incremental strategies. *Journal of Software Engineering Research and Development*, 4, 1-32. <https://link.springer.com/content/pdf/10.1186/s40411-016-0031-8.pdf>
- Barraood, S., Haslina, H., & Baharom, F. (2021, febrero 1). *A Comparison Study of Software Testing Activities in Agile Methods*.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M.,

- Grenning, J., Highsmith, J., Hunt, A., & Jeffries, R. (2018). *Manifesto for agile software development*. [https://ai-learn.it/wp-content/uploads/2019/03/03\\_ManifestoofAgileSoftwareDevelopment-1.pdf](https://ai-learn.it/wp-content/uploads/2019/03/03_ManifestoofAgileSoftwareDevelopment-1.pdf)
- Boukhelif, M., Hanine, M., & Kharmoum, N. (2023). A decade of intelligent software testing research: A bibliometric analysis. *Electronics*, 12(9), 2109.
- Bowen Mendoza, L. E. (2023). *Modelo difuso para la evaluación por pares*. <http://helvia.uco.es/xmlui/handle/10396/25876>
- Cababie, P., & Troilo, F. (2021). *Metodologías ágiles en equipos de operaciones del área de tecnología de la información (TI)* (No. 783). Serie Documentos de Trabajo.
- Cabrera Vigil, L. E., Landa Rodríguez, I. E., & Vindel Figueroa, J. R. (2024). *Estudio y análisis sobre la creación e implementación de un plan en el proceso de pruebas de software, basado en la Norma ISO/IEC/IEEE 29119 apartados: 1: 2022, 2: 2021; 3: 2021; 4: 2021, aplicado al área de control de calidad en la empresa system out of the box, el salvador* (Doctoral dissertation, Universidad Don Bosco).
- Cabrera Vigil, L. E., Landa Rodríguez, I. E., & Vindel Figueroa, J. R. (2024). *Estudio y análisis sobre la creación e implementación de un plan en el proceso de pruebas de software, basado en la Norma ISO/IEC/IEEE 29119 apartados: 1: 2022, 2: 2021; 3: 2021; 4: 2021, aplicado al área de control de calidad en la empresa system out of the box, el salvador* (Doctoral dissertation, Universidad Don Bosco).
- Chukwurah, E. G., & Aderemi, S. (2024). Elevating team performance with scrum: insights from successful US technology companies. *Engineering Science & Technology Journal*, 5(4), 1357-1371.
- Chukwurah, E. G., & Aderemi, S. (2024). Elevating team performance with scrum: insights from successful US technology companies. *Engineering Science & Technology*

*Journal*, 5(4), 1357-1371.

Conboy, K., Dennehy, D., & O'Raghallaigh, P. (2020). *Agile IS Development: The Role of Organizational Culture in Facilitating Agility*.

Damodaran, A. (2021). *Corporate finance*. Hoboken: Wiley.

de Farias Junior, I. H., Torcate, A., & Gois, D. (2021). Melhores Práticas de Comunicação em Desenvolvimento Distribuído de Software: Uma Revisão Sistemática da Literatura. *Revista dos Mestrados Profissionais*, 10(1), 86-108.

Díaz-Bravo, L., Torruco-García, U., Martínez-Hernández, M., & Varela-Ruiz, M. (2013). La entrevista, recurso flexible y dinámico. *Investigación en educación médica*, 2(7), 162-167.

Dietrichson, J., Gudmundsson, J., & Jochem, T. (2022). Why don't we talk about it? Communication and coordination in teams. *Journal of Economic Behavior & Organization*, 197, 257-278.

Dietrichson, J., Gudmundsson, J., & Jochem, T. (2022). Why don't we talk about it? Communication and coordination in teams. *Journal of Economic Behavior & Organization*, 197, 257-278.

Erich, F. M. A., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. *Journal of Software: Evolution and Process*, 29(6), e1885.  
<https://doi.org/10.1002/smr.1885>

Esterkin, V., & Pons, C. (2017). Evaluación de calidad en el desarrollo de software dirigido por modelos. *Ingeniare. Revista chilena de ingeniería*, 25(3), 449-463.

Ferrarini, L., & Enzo, E. L. (2023). *Maestría en gestión estratégica*.  
[http://bibliotecadigital.econ.uba.ar/download/tpos/1502-2881\\_FerrariniELE.pdf](http://bibliotecadigital.econ.uba.ar/download/tpos/1502-2881_FerrariniELE.pdf)

FERRARINI, L., & ENZO, E. L. (2023). MAESTRÍA EN GESTIÓN ESTRATÉGICA.

- Flores-Cerna, F., Sanhueza-Salazar, V.-M., Valdés-González, H.-M., & Reyes-Bozo, L. (2022). Metodologías ágiles: Un análisis de los desafíos organizacionales para su implementación. *Revista científica*, 43(1), 38-49.
- Folgueiras Bertomeu, P. (2016). La entrevista. *Documents de treball / Informes (Mètodes d'Investigació i Diagnòstic en Educació)*.  
<https://diposit.ub.edu/dspace/handle/2445/99003>
- García, J., Pérez, M., & López, A. (2020). *Quality Management in Software Development Projects*.
- García-Peñalvo, F. J. (2017). *Revisión sistemática de literatura en los Trabajos de Final de Máster y en las Tesis Doctorales*.
- Grant, D. (2023, enero 18). *Software quality in 2023 – trends to watch*. Jam Blog.  
<https://strawberryjam.ghost.io/quality-engineering-trends-to-watch-in-2023/>
- Green Arias, P. (2023). *Diseño de una propuesta de implementación del departamento de SQA en una empresa de software*. Universidad Nacional de Costa Rica.
- Herbold, S. (2020). On the costs and profit of software defect prediction. *IEEE Transactions on Software Engineering*, 47(11), 2617-2631.
- Hernández-Sampieri, R., & Mendoza, C. (2020). *Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta*.  
<https://www.academia.edu/download/64591365/Metodolog%C3%ADvestigaci%C3%B3n.%20Rutas%20cuantitativa,%20cualitativa%20y%20mixta.pdf>
- Hoda, R., Noble, J., & Marshall, S. (2012). Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering*, 39(3), 422-444.  
<https://dspace.usthb.dz/bitstream/handle/123456789/8973/nl3076.pdf?sequence=1&isAllowed=y>

- Hoda, R., Noble, J., & Marshall, S. (2012). Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering*, 39(3), 422-444.
- Howden, W. E. (2011, July). Error-based software testing and analysis. In *2011 IEEE 35th Annual Computer Software and Applications Conference Workshops* (pp. 161-167). IEEE.
- Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Pearson Education.
- ISO. (2019). *ISO/IEC/IEEE 15026-1:2019*. <https://www.iso.org/standard/73567.html>
- ISO/IEC 25010. (2023). *ISO/IEC 25010:2023*. <https://www.iso.org/standard/78176.html>
- Jens, Dietrichson., Jens, Gudmundsson., Torsten, Jochem. (2022). 2. Why don't we talk about it? Communication and coordination in teams. *Journal of Economic Behavior and Organization*, doi: 10.1016/j.jebo.2022.02.018
- Jiménez Chaves, V. E. (2012). El estudio de caso y su implementación en la investigación. *Revista Internacional de Investigación en Ciencias Sociales*, 8(1), 141-150.
- John, P., Walsh., Nancy, G., Maloney. (2006). 5. Collaboration Structure, Communication Media, and Problems in Scientific Work Teams. *Journal of Computer-Mediated Communication*, doi: 10.1111/J.1083-6101.2007.00346.X
- John, P., Walsh., Nancy, G., Maloney. (2006). 5. Collaboration Structure, Communication Media, and Problems in Scientific Work Teams. *Journal of Computer-Mediated Communication*, doi: 10.1111/J.1083-6101.2007.00346.X
- Khatri, F. (2022, diciembre 20). What is Continuous Testing? *LambdaTest*. <https://www.lambdatest.com/blog/what-is-continuous-testing/>
- Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology*

*Organizations*. IT Revolution.

Lalband, N., & Kavitha, D. (2020). Software Development Technique for the Betterment of End User Satisfaction using Agile Methodology. *TEM Journal*, 9, 992-1002. <https://doi.org/10.18421/TEM93-22>

Leong, J., May Yee, K., Baitsegi, O., Palanisamy, L., & Ramasamy, R. K. (2023). Hybrid project management between traditional software development lifecycle and agile based product development for future sustainability. *Sustainability*, 15(2), 1121.

Linares Guerra, E. M., Díaz Aguirre, S., González Pérez, M. M., Pérez Rodríguez, E., & Córdova Vázquez, V. (2021). Metodología para el diagnóstico ambiental comunitario con fines investigativos desde el posgrado académico. *Revista Universidad y Sociedad*, 13(4), 309-319. [http://scielo.sld.cu/scielo.php?pid=S2218-36202021000400309&script=sci\\_arttext&tlng=pt](http://scielo.sld.cu/scielo.php?pid=S2218-36202021000400309&script=sci_arttext&tlng=pt)

Lloyd, C. (2018). *FOR SOCIOLOGY SENIOR THESES*.

Machuca-Villegas, L., Gasca-Hurtado, G. P., Restrepo Tamayo, L. M., & Morillo Puente, S. (2020). Social and human factor classification of influence in productivity in software development teams. In *Systems, Software and Services Process Improvement: 27th European Conference, EuroSPI 2020, Düsseldorf, Germany, September 9–11, 2020, Proceedings 27* (pp. 717-729). Springer International Publishing.

Machuca-Villegas, L., Gasca-Hurtado, G. P., Restrepo Tamayo, L. M., & Morillo Puente, S. (2020). Social and human factor classification of influence in productivity in software development teams. In *Systems, Software and Services Process Improvement: 27th European Conference, EuroSPI 2020, Düsseldorf, Germany, September 9–11, 2020, Proceedings 27* (pp. 717-729). Springer International Publishing.

Mahmoud, A. N., & Santos, V. (2021). Statistical analysis for revealing defects in software

- projects: systematic literature review. *International Journal of Advanced Computer Science and Applications*, 12(11).
- Marsicano, D. V. Pereira, F. Q. B. da Silva and C. França, "Team Maturity in Software Engineering Teams," *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Toronto, ON, Canada, 2017, pp. 235-240. <https://ieeexplore.ieee.org/document/8170109/authors#authors>
- Mishra, A., & Otaiwi, Z. (2020). DevOps and software quality: A systematic mapping. *Computer Science Review*, 38, 100308. <https://doi.org/10.1016/j.cosrev.2020.100308>
- Nogueira, E., Moreira, A., Lucrédio, D., Garcia, V., & Fortes, R. (2016). Issues on developing interoperable cloud applications: Definitions, concepts, approaches, requirements, characteristics and evaluation models. *Journal of Software Engineering Research and Development*, 4(1), 7. <https://doi.org/10.1186/s40411-016-0033-6>
- Pargaonkar, S. (2021). Unveiling the Challenges, A Comprehensive Review of Common Hurdles in Maintaining Software Quality. *Journal of Science & Technology*, 2(1), 85-94.
- Pariona Ccoyllar, B. D., & Vargas Marin, C. D. (2024). *Propuesta de una guía para minimizar el tiempo y costo en la etapa de diseño de proyectos de escuelas bicentenario del paquete 02-lima del peip-eb en san juan de lurigancho utilizando los contratos nec-4 tipo c.*
- Pascarella, L., Palomba, F., & Bacchelli, A. (2019). Fine-grained just-in-time defect prediction. *Journal of Systems and Software*, 150, 22-36. <https://doi.org/10.1016/j.jss.2018.12.001>
- Rahman, M., Helms, M., & Raut, R. (2019). *Information Systems Development Through Agile Methodology: A Case Study of Blue Cross Blue Shield of Michigan.*

- Ramírez, D. M. (2024). La investigación cualitativa: definiendo otra gran fuerza de indagación científica. *Rhombus*, 4(1), 77-93  
<https://revistas.ulacit.ac.cr/index.php/rhombus/article/download/169/132>
- Robayo, J. D. J. (2023). Calidad en el desarrollo de software desde una perspectiva bibliométrica en el periodo 2018 -2022. *INNOVAR Research Journal*, 8(3), 173-188.  
<https://revistas.uide.edu.ec/index.php/innova/article/view/2299/2104>
- Robles, R. D. B., Cruz, O. D. C., Medina, J. L. F., Valdiviezo, Y. G., Mauricio, L. A. S., & Rodríguez, C. J. S. (2023). Auditoría y control de calidad en proyectos de ingeniería.
- Santos, P. de O., & de Carvalho, M. M. (2022). Exploring the challenges and benefits for scaling agile project management to large projects: A review. *Requirements Engineering*, 27(1), 117-134. <https://doi.org/10.1007/s00766-021-00363-3>
- SEI. (2018). *Capability Maturity Model Integration (CMMI)® for Development, Version 2.0*.  
<https://www.sei.cmu.edu/>
- Severo, R.N., & Andrade, A.P. (2024). COMUNICAÇÃO E COORDENAÇÃO EM EQUIPES VIRTUAIS DE DESENVOLVIMENTO DISTRIBUÍDO DE SOFTWARE. *Revista Gestão em Análise*.
- Shahin, M., Ali Babar, M., & Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*, 5, 3909-3943. IEEE Access.  
<https://doi.org/10.1109/ACCESS.2017.2685629>
- Sielis, G. A., Tzanavari, A., & Papadopoulos, G. A. (2017). ArchReco: A software tool to assist software design based on context aware recommendations of design patterns. *Journal of Software Engineering Research and Development*, 5(1), 2.  
<https://doi.org/10.1186/s40411-017-0036-y>

- Sindhu, B.D . (2019). Artificial intelligence in software test automation: A systematic literature review. *International Journal of Emerging Technologies and Innovative Research (www.jetir.org/ UGC and issn Approved), ISSN, 2349-5162.*
- Sommerville, I. (2016). *Software engineering* (Tenth edition, global edition). Pearson.
- Spiegler, S. V., Heinecke, C., & Wagner, S. (2021). An empirical study on changing leadership in agile teams. *Empirical Software Engineering, 26*(3), 41. <https://doi.org/10.1007/s10664-021-09949-5>
- Stumbles, T. (2021). *Software Development Methodologies timeline.* <https://www.officetimeline.com/blog/software-development-methodologies-timeline>
- Sutherland, K. S. & J. (2020). *La Guía Definitiva de Scrum: Las Reglas del Juego.* <https://repositorio.uvm.edu.ve/handle/123456789/59>
- Tecnológico de Monterrey, Innovación Educativa (Director). (2018, enero 11). *Evaluación de pares* [Video recording]. <https://www.youtube.com/watch?v=g3Mo7f7KTC0>
- Trippensee, L., & Remané, G. (2021). Practices for Large-Scale Agile Transformations: A Systematic Literature Review. *AMCIS 2021 Proceedings.* [https://aisel.aisnet.org/amcis2021/it\\_projmgmt/it\\_projmgmt/5](https://aisel.aisnet.org/amcis2021/it_projmgmt/it_projmgmt/5)
- Van Casteren, W. (2017). *The Waterfall Model and the Agile Methodologies: A comparison by project characteristics - short.* <https://doi.org/10.13140/RG.2.2.10021.50403>
- Vical, Núñez, M. E. D. L. C., Tantapoma, M. E. V., & Jave, A. C. L. (2021). Influencia de la comunicación interna en el clima organizacional dentro de una empresa. *SCIÉENDO, 24*(3), 147-153.

## **Apéndice**

## **Apéndice 1: Entrevista**

### **Optimización de Procesos de Control de Calidad (QA) en Empresa de Desarrollo de Software con Equipos Reducidos**

#### **I. Parte Introducción:**

Mi nombre es Carlos Andrés Obando Jiménez, profesional en el área de Ingeniería en Sistemas de Información de la Universidad Nacional de Costa Rica, y actualmente estoy cursando una Maestría en Administración de Tecnologías de la Información (MATI) con Énfasis en Administración de Proyectos.

Este instrumento forma parte de un proyecto de investigación del trabajo final de graduación que se enfoca en Optimización de Procesos de Control de Calidad (QA) en Empresa de Desarrollo de Software con Equipos Reducidos, para esto le rogamos de su colaboración para lograr contestar una serie de preguntas que fueron hechas con el fin de conocer aspectos puntuales relacionados con el tema de investigación.

Desde ya le damos las gracias por su atención y por su colaboración que es invaluable para poder desarrollar los objetivos de esta investigación. Personalmente, me comprometo con usted a hacerle llegar los resultados finales del proyecto final de graduación a una vez haya sido presentado y aprobado para lo cual opcionalmente le agrego un campo de registro de su correo electrónico por medio del cual le haría llegar el documento, en caso de que no desee recibir este documento omita el campo de llenar correo electrónico.

Por último, quiero mencionar que los datos que usted proporcione en este instrumento serán únicos, exclusiva y confidencialmente, con fines del desarrollo de esta investigación y sus datos no serán utilizados para ningún otro fin.

## **II. Parte preguntas**

### **1. ¿Con cuál de los siguientes géneros se identifica usted?**

Masculino  Femenino  Otro  Prefiero no decir

### **2. ¿En cuál de los siguientes rangos de edad se encuentra usted?**

Menos de 25 años  25-34 años  35-44 años  45-54 años  55 años o más

### **3. ¿Cuál es su nivel académico actualmente?**

Secundaria  Bachillerato Universitario  Licenciatura  Maestría  Doctorado

### **4. ¿Cuántos años de experiencia tiene usted en el área del aseguramiento de la calidad?**

Menos de 1 año  1-3 años  4-6 años  7-10 años  Más de 10 años

### **5. ¿Cuál es el rol que usted actualmente desempeña dentro de la empresa de desarrollo de software?**

Gerente de Proyectos  Líder de Aseguramiento de Calidad (QA Lead)

Analista de Aseguramiento de Calidad Senior (Senior QA)  Analista de aseguramiento de Calidad Junior (Junior QA)  Encargado de Desarrollo de Software (Head of Development)

**6. ¿Qué tan de acuerdo estás con la siguiente afirmación? En un equipo de calidad de software, a pesar de ser un equipo reducido, se demuestra eficiencia en la detección y en la solución de defectos en el software.**

Totalmente en desacuerdo  En desacuerdo  Neutral  De acuerdo  Totalmente de acuerdo

**7. ¿Qué tan satisfecho está usted con la comunicación y colaboración dentro del equipo reducido de calidad?**

Totalmente en desacuerdo  En desacuerdo  Neutral  De acuerdo  Totalmente de acuerdo

**8. ¿En qué medida considera usted que el tamaño reducido de los equipos de calidad afecta la capacidad de cumplir con los estándares de calidad trazados por la organización?**

Totalmente en desacuerdo  En desacuerdo  Neutral  De acuerdo  Totalmente de acuerdo

**¿Qué tan de acuerdo está usted con la siguiente afirmación? "Contar con personas experimentadas dentro de estos equipos reducidos de calidad es crucial para garantizar la eficiencia en la detección de defectos en el software."**

Totalmente en desacuerdo  En desacuerdo  Neutral  De acuerdo  Totalmente de acuerdo

**9. ¿Qué tan importante considera usted la experiencia previa en calidad de software para el desempeño efectivo en un equipo de calidad de software reducido?**

Totalmente en desacuerdo  En desacuerdo  Neutral  De acuerdo  Totalmente de acuerdo

**10. ¿En qué medida está usted de acuerdo con la siguiente afirmación: 'Contar con personal experimentado en un equipo de calidad de software reducido contribuye a la mejora de los procesos de prueba'?**

Totalmente en desacuerdo  En desacuerdo  Neutral  De acuerdo  Totalmente de acuerdo

**11. ¿En qué medida está usted de acuerdo con la siguiente afirmación: 'La optimización de procesos en un equipo de calidad de software reducido depende en gran parte de la capacidad del equipo para adaptarse a nuevas herramientas y metodologías'?**

Totalmente en desacuerdo  En desacuerdo  Neutral  De acuerdo  Totalmente de acuerdo

**12. ¿En qué medida está usted de acuerdo con la siguiente afirmación: 'La asignación de roles dentro de un equipo reducido de QA es efectiva para optimizar los procesos de control de calidad'?**

Totalmente en desacuerdo  En desacuerdo  Neutral  De acuerdo  Totalmente de acuerdo

**13. ¿En qué medida está usted de acuerdo con la siguiente afirmación: 'La automatización de pruebas puede compensar las limitaciones de recursos dentro de los equipos de calidad de software reducidos'?**

Totalmente en desacuerdo  En desacuerdo  Neutral  De acuerdo  Totalmente de acuerdo

**14. ¿Qué tan satisfecho está usted con la actual implementación de los procesos de control de calidad en un equipo reducido de su organización?**

Totalmente en desacuerdo  En desacuerdo  Neutral  De acuerdo  Totalmente de acuerdo

*¡Muchas gracias por su participación!*

---

Por último, quiero expresar mi sincero agradecimiento por participar en esta entrevista, parte de mi trabajo final de graduación. Aprecio su tiempo y valiosos aportes, los cuales confío en que contribuirán significativamente a mi investigación sobre la optimización de procesos de control de calidad (QA) en empresas de

desarrollo de software con equipos reducidos. Valoro mucho su colaboración y disposición para compartir su experiencia profesional.

Atentamente,

Carlos Andrés Obando Jiménez

## Apéndice 2: Caso de estudio

<b>Caso de estudio de Golabs Tech</b>	
<b>Contexto general del caso de estudio: Descripción de la empresa</b>	Golabs Tech es una empresa costarricense dedicada al desarrollo de software, ubicada en la zona norte del país, en la provincia de Alajuela, específicamente en Ciudad Quesada. Con más de 10 años de experiencia en la industria, se especializa en la provisión de servicios tecnológicos para compañías en Estados Unidos y Canadá, consolidándose como un referente en outsourcing de soluciones de software.
<b>Objetivo del caso de estudio: Justificación de la elección de la empresa</b>	Golabs Tech fue seleccionada para este caso de estudio debido a su perfil de empresa nacional, su enfoque en mercados internacionales, y, en particular, por el análisis detallado de su departamento de calidad de software. El equipo reducido de profesionales dedicados a la calidad de software en la compañía la convierte en un escenario ideal para evaluar y optimizar procesos de control de calidad. La empresa cuenta con métricas relevantes y un entorno propicio para implementar estrategias de mejora en este campo, lo cual permite generar aprendizajes aplicables a otras empresas del sector.
<b>Definición del caso de estudio</b>	
<b>Descripción general:</b>	Golabs Tech es una compañía especializada en servicios de outsourcing para el mercado estadounidense, con un equipo aproximado de 80 colaboradores. Ofrece tres líneas de servicio clave: Staff Augmentation: Personal especializado que se integra en equipos de clientes. Time and Materials: Contratación por horas de servicios. Custom Development: Desarrollo de productos a medida, donde se colabora con startups para construir versiones iniciales de productos que, en caso de éxito, evolucionan hacia contratos más amplios.
<b>Equipo de QA: Roles, tamaño y responsabilidades</b>	El equipo de calidad de software en Golabs Tech está compuesto por cinco ingenieros, clasificados en los niveles Junior, Mid y Senior. Sus responsabilidades principales incluyen: Gestión de casos de prueba (Test case management) Planificación de pruebas (Test Planning) Automatización de pruebas (Web y APIs) Control de calidad de datos (Data Quality)

<b>Procesos de QA: Metodologías y herramientas</b>	En la mayoría de los casos, Golabs Tech adapta sus metodologías y herramientas a los requerimientos específicos de sus clientes. No obstante, cuando el cliente carece de estas herramientas, la empresa colabora en su implementación, siguiendo metodologías ágiles como BDD, TDD y CI/CD.
<b>Contexto y problemática</b>	
<b>Desafíos previos:</b>	Uno de los principales desafíos que enfrenta la empresa es la dificultad para encontrar profesionales altamente calificados en aseguramiento de la calidad, particularmente en roles especializados como Senior QA Automation con experiencia en manejo de datos. Esta situación ha representado un obstáculo recurrente, ya que la demanda de profesionales con estas habilidades excede la oferta disponible en el mercado local.
<b>Recolección de datos</b>	
<b>Revisión de documentación</b>	La revisión documental en Golabs Tech es limitada y se circunscribe a un Career Path estándar, sin mayor soporte documental adicional.
<b>Flujo de trabajo y actividades diarias</b>	Las actividades operativas diarias se gestionan en función de las demandas del cliente, utilizando una plataforma interna en la que cada colaborador debe registrar su trabajo semanalmente. Esto permite a la empresa mapear la actividad y desempeño de su personal de manera eficiente.
<b>Planes de capacitación</b>	Además del Career Path estándar, Golabs Tech ofrece cursos bajo demanda en temas técnicos relacionados con la calidad de software, utilizando plataformas como Coursera para fomentar el desarrollo de competencias específicas dentro de su equipo.

Observaciones finales:

---



---

### **Apéndice 3: Implementación de propuesta**

# **Optimización de procesos de QA en equipos reducidos de empresas de desarrollo de software**

Versión 1.1

Noviembre -2024

## Tabla de contenido

Tabla de contenido .....	2
Introducción .....	4
Definiciones y acrónimos .....	5
Objetivos de la propuesta.....	6
Estrategias Generales de Optimización de QA.....	6
Organización del Equipo de QA .....	6
Estructura de Roles .....	6
<i>Distribución de tareas</i> .....	8
Actividades esenciales .....	8
Planificación y diseño de pruebas.....	9
Ejecución y seguimiento de pruebas.....	9
Metodologías ágiles .....	10
Protocolos de QA.....	10
Protocolos de Ejecución de Pruebas para Equipos Reducidos .....	10
Protocolos de Validación y aprobación.....	12
Implementación de propuesta .....	13
Cronograma de actividades.....	13
Costos y beneficios .....	14
Conclusiones.....	16
Biografía .....	17

## **Introducción**

Los equipos de QA en empresas de desarrollo de software son parte integral del ciclo de vida del desarrollo de un producto, ya que se enfocan en asegurar los resultados de alta calidad a través de diversas metodologías (procesos) y dinámicas de equipo. Estos equipos no solo son los responsables del aseguramiento, sino que también juegan un papel crucial en el fomento de la colaboración y la comunicación, lo cual es esencial para lograr la calidad del software. Es por ello por lo que contar con estos equipos desde las primeras etapas del desarrollo puede reducir defectos, disminuir costos y mejorar la satisfacción del cliente (Alphonse, 2024).

La presente propuesta es fruto de una revisión exhaustiva de literatura especializada, del conocimiento acumulado de profesionales del sector y de la experiencia de una empresa líder en QA, ya que se han identificado varios desafíos clave que afectan la eficiencia. Entre ellos, destacan la organización y asignación de tareas, así como la limitación de recursos para la documentación y registros de procesos, ofreciendo soluciones orientadas a la optimización de procesos, la eficiencia operativa, la mejora de la gestión de recursos y el aseguramiento del flujo de trabajo continuo (Navarro,2023).

Se espera que, al implementar esta propuesta, los equipos de QA puedan enfrentar los desafíos derivados del trabajo con recursos humanos limitados, sin comprometer la calidad del producto final. Por esta razón, la propuesta abarca un enfoque integral, abordando la implementación de metodologías ágiles, la automatización de pruebas, el fortalecimiento de la comunicación interna y el desarrollo continuo de habilidades resolutivas. Estos enfoques están diseñados para optimizar los recursos y maximizar el impacto de los equipos de QA en la calidad del software y la satisfacción del cliente.

## **Definiciones y acrónimos**

Como parte fundamental de esta propuesta, se presenta un conjunto de conceptos y acrónimos clave que forman parte del departamento de control de calidad. Estos términos se explican para proporcionar un contexto claro y facilitar la comprensión de los procesos, metodologías y herramientas que se abordarán en el documento.

**QA:** Quality Assurance (QA) es un proceso clave en el desarrollo de productos de software, el cual está encargado de garantizar que se cumplan los estándares de calidad requeridos. El principal enfoque de QA se encuentra en las pruebas para asegurar que el producto que se desarrolla funcione correctamente, tal como se espera (Goericke, 2019).

**Calidad:** Según la Organización Internacional de Normalización (ISO; 2015), la calidad se define como el nivel en el cual un conjunto de atributos propios de un producto, servicio o proceso logra satisfacer las expectativas del cliente o de las partes involucradas.

**Scrum:** Hernández (2024) define Scrum como "(...) un marco de trabajo para la gestión y desarrollo de productos complejos" (p.14).

**Automatización de pruebas:** En el sitio web de Atlassian, Rehkopf (s. f.) explica que la automatización de pruebas es un método que emplea herramientas de software para simular el proceso manual de revisión y validación de un producto de software. En lugar de depender de la intervención humana en cada fase, este enfoque automatiza las tareas repetitivas, lo que resulta en una detección temprana más ágil de errores. Además, garantiza que el software cumpla con los estándares de calidad esperados.

## **Objetivos de la propuesta**

La propuesta de optimización de procesos de control de calidad en equipos reducidos de empresa de desarrollo de software tiene los siguientes objetivos:

- a) Mejorar los procesos de QA, maximizando el uso de recursos limitado sin comprometer los estándares de calidad.
- b) Introducir metodologías ágiles, como scrum para facilitar la gestión de tareas, la comunicación y colaboración de los equipos.
- c) Proponer estrategias de optimización de proceso de pruebas automatizadas y CI/CD que reduzcan la carga de trabajo manual de la empresa, para asegurar la alineación de los objetivos y mejorar la calidad.

## **Estrategias Generales de Optimización de QA**

### **d) Organización del Equipo de QA**

De acuerdo con Navarro (2023), uno de los puntos claves en la optimización de procesos de equipos reducidos depende de una estructura organizacional, ya que al contar con una organización bien definida va a permitir una distribución equitativa, mejorar la comunicación y agilizar la toma de decisiones. Considerando lo anterior, es necesario no obviar que un equipo de QA reducido debe mantener claro los roles de cada uno de sus miembros.

### **e) Estructura de Roles**

En equipo reducido, los roles y responsabilidades deben estar bien definidas, ya que estas deben indicar a cada uno de los miembros, las actividades y responsabilidades requeridas para el desarrollo del proyecto, ya que el tener esto claro, se mejora la eficiencia, se reduce las redundancias y fomenta las responsabilidades (Navarro, 2023).

**Tabla 9***Equipo de QA en empresas de software*

<b>Rol</b>	<b>Descripción</b>	<b>Responsabilidades</b>	<b>Habilidades</b>
<b>QA Lead</b>	También conocido como Líder de QA, y este es el encargado de la coordinación, planificación y supervisión de actividades de calidad a lo largo del ciclo de vida del proyecto (Baron, 2022). Su rol es estratégico en el equipo de QA, ya que asigna los recursos técnicos y humanos que son necesarios para resolver los problemas que surjan.	Responsabilidad QA Lead: Coordinación del equipo de QA Planificación de las estrategias de pruebas Revisión de los resultados Comunicación con otros equipos Implementa estándares de calidad	Conocimientos técnicos en pruebas de software. Habilidad en gestión y manejo de equipos. Comunicación efectiva. Experiencia en planificación y gestión de estrategias de prueba. Manejo de herramientas de integración continua DevOps Gestión de riesgos
<b>QA Automation Engineer</b>	De acuerdo con Baron (2022), es el encargado de diseñar, ejecutar, planificar y reportar pruebas en el desarrollo del software, sin necesidad de pruebas manuales repetitivas. Responsabilidad del QA Automation Engineer:	Diseña e implementa script de pruebas automatizadas. Seleccionar las herramientas de automatización a implementar Colaborar con el equipo de desarrollo Analizar los resultados de las pruebas	Conocimiento en herramientas de automatización de pruebas. Dominio en pruebas de Software. Conocimiento de herramientas relacionadas a la Integración Continua. Conocimiento en Frameworks de prueba. Documentación y reporte de las pruebas realizadas. Conocimiento en Arquitecturas del Software y API's
<b>Manual QA Tester</b>	Es el responsable de llevar a cabo las revisiones relacionadas, tanto de los procesos actuales como de los futuros, y se encarga de la ejecución de pruebas, ya sean automatizadas o manuales, para asegurar que el producto cumpla con los estándares de calidad (Baron, 2022).	Diseño y ejecución de casos de prueba. Identificación y Registro de Defectos (Bugs) Pruebas funcionales Pruebas de regresión Automatización de pruebas	Dominio de técnicas de prueba de software. Conocimiento de los ciclos de vida del desarrollo de software. Atención a los detalles. Gestión de documentación y reporte de errores. Resolución de problemas. Conocimiento para trabajar con metodologías ágiles. Experiencia de programación y bases de datos.

### ***Distribución de tareas***

La distribución de tareas debe estar orientada a la optimización del uso de los recursos disponibles, maximizando la eficiencia operativa y el desarrollo de habilidades dentro del equipo, ya que una correcta asignación de responsabilidades no solo beneficia el tema de productividad, sino que también optimiza el aprendizaje y la colaboración de los miembros (Navarro, 2023). En este punto, los equipos reducidos de QA deben tener ciertos elementos en cuenta para lograr una distribución óptima de las tareas, entre los cuales resaltan:

- a) **La especialización en áreas técnicas:** Es una de las claves fundamentales en equipos de QA reducidos, ya que permiten a los miembros el desarrollo de conocimientos profundos y enfocados en áreas específicas, lo cual no solo incrementa su competencia técnica, sino que también mejora la eficiencia en la ejecución de tareas complejas, al contar con expertos ampliamente capacitados para resolver problemas claves (Laoyan,2024).
- b) **La colaboración y la comunicación fluida entre los miembros del equipo:** Es esencial, ya que el éxito del trabajo en equipos reducidos depende en gran medida de una interacción constante y efectiva. Esto se logra a través de la revisión conjunta de avances, donde cada miembro del equipo presenta los resultados y avances durante el sprint, con la intención de identificar mejoras o ajustes para el siguiente sprint, garantizando un correcto alineamiento de los objetivos del proyecto.

### **Actividades esenciales**

El proceso de gestión de actividades de un equipo de QA debe tomar como base la planificación y el diseño de pruebas en función del aseguramiento de calidad, ya que estas van a definir las bases para la ejecución a lo largo del ciclo de vida del software (Ahmad, 2022).

#### **f) Planificación y diseño de pruebas**

En esta etapa el equipo debe identificar y definir los objetivos y pruebas a realizar, determinando las prioridades del producto. Esta planificación implica no solo la evaluación de aspectos técnicos, sino también la parte funcional del mismo; acá se establecen los recursos humanos y herramientas tecnológicas y los tiempos. Mientras que en la parte de diseño de pruebas es donde se priorizan y diseñan los casos de prueba en función de las necesidades del producto. A continuación, se detallan los pasos fundamentales para la planificación y priorización de las pruebas:

- a) Identificación de objetivos de prueba.
- b) Priorización de casos de prueba.
- c) Diseño de casos de prueba
- d) Gestión de riesgos
- e) Planificación de recursos y herramientas.

#### **g) Ejecución y seguimiento de pruebas**

Las pruebas se ejecutan a través de herramientas como TestRail, donde el QA puede crear y gestionar casos de prueba, permitiendo la ejecución de estos, con el fin de monitorizar los segmentos del producto. Una de las ventajas de la herramienta es su capacidad para generar reportes del estado de la prueba, lo que permite obtener información sobre los resultados, incluyendo la identificación de errores o si todo funciona correctamente. Además, la centralización de procesos de gestión en una herramienta, a la cual todos los miembros del equipo tienen acceso, favorece la transparencia y conocimiento en tiempo real, lo que permite una gestión eficiente (Ahmad, 2022).

## **h) Metodologías ágiles**

La implementación de metodologías ágiles como Scrum o Kanban en equipos reducidos es fundamental porque permite una entrega periódica, la cual se adapta a los cambios y pruebas durante el desarrollo del producto, ya que se trabaja mediante ciclos cortos, que integran reuniones diarias y revisiones generales de los procesos. Esto facilita la retroalimentación de los miembros del equipo en función de la calidad del producto (Benavides et al., 2024). Algunas de las metodologías ágiles más empleadas en la actualidad por los equipos de desarrollo y calidad de software en las empresas son Scrum y Kanban.

## **Protocolos de QA**

En el departamento de QA, los protocolos son fundamentales para garantizar que los procesos de calidad se lleven a cabo de manera consistente; estos contienen las normas, procedimientos y mejores prácticas que guían a los equipos.

## **i) Protocolos de Ejecución de Pruebas para Equipos Reducidos**

En un equipo de QA reducido, es esencial enfocar los esfuerzos en los casos de prueba más críticos, priorizando las funcionalidades de mayor impacto en el software. Dado el número limitado de miembros, se debe asegurar que se cubran los aspectos más importantes del producto para cumplir con las expectativas de calidad. A pesar de la naturaleza breve de los procesos, estos deben ser claros, bien definidos y accesibles para que todos los miembros del equipo puedan ejecutarlos de manera eficiente sin necesidad de explicaciones adicionales que retrasen el proceso (Song y Park, 2015).

**Tabla 10***Protocolos de Ejecución de Pruebas para Equipos Reducidos*

<b>Fase</b>	<b>Descripción</b>	<b>Tareas y herramientas</b>
<b>1. Enfoque en Pruebas Prioritarias</b>	Se debe concentrar los esfuerzos en los casos de prueba críticos, maximizando la eficiencia y la funcionalidad del software que mide la frecuencia de uso, complejidades técnicas y el grado de impacto del error en el usuario final.	Matriz de impacto de frecuencia (Asana o Jira): Evaluar funcionalidad según, grado de impacto, uso y complejidad. Clasificación de casos de prueba por su frecuencia y complejidad. Priorizar casos de prueba (Google Sheets): Analiza y enfocar esfuerzos en casos más importantes
<b>2. Estandarización de Protocolos</b>	Desarrollar protocolos claros y uniformes que aseguren que los miembros puedan ejecutar pruebas de manera consistente y sin ambigüedad.	Documentación clara y concisa (Confluence): Crear guías paso a paso para cada tipo de pruebas. Plantillas reutilizables (Google Docs o Microsoft Word): Crear formatos estandarizados para los casos de pruebas. Creación de checklist: (Excel o Google Sheets): Verificar las configuraciones y condiciones antes de ejecutar cada prueba.
<b>3. Automatización y Herramientas</b>	Implementar herramientas de automatización para pruebas repetitivas y de regresión.	Identificación de pruebas críticas (Cypress): Configuración de pruebas automatizadas en cada ciclo del desarrollo (Jenkins o GitHub). Monitoreo y reporte de estado de pruebas automatizadas (Jenkins o GitHub)
<b>4. Capacitación Cruzada</b>	Asegura que los miembros del equipo puedan cubrir diversas áreas de pruebas a través de capacitación interna y rotación.	Rotación de roles entre los miembros (Asana o ClickUp): Exponer a todos los miembros a diferentes áreas del software. Sesiones de capacitación internas (Google Meet o Microsoft): Organizar formación continua. Documentación compartida (Google drive): Asegura el acceso a guías y protocolos centralizados.

## j) **Protocolos de Validación y aprobación**

Como su nombre los indica estos buscan garantizar que el software que se ha desarrollado cumpla con los requisitos preestablecidos con la calidad establecida antes de que el producto sea lanzado en producción. Estos protocolos aseguran que las pruebas realizadas sean efectivas y que los resultados obtenidos sean revisados de manera objetivo y formal.

**Tabla 11**

*Protocolos de valoración y aprobación de equipos reducidos*

<b>Fase</b>	<b>Descripción</b>	<b>Tareas y herramientas</b>
<b>1. Verificación</b>	Proceso de asegurarse de que el software cumple con los requisitos técnicos y funcional definidos.	Verificación de requisitos (Jira): Revisa si el software se ajusta a los requisitos previos. Revisión de diseño y arquitectura (GitHub): Busca asegurar que sean coherentes con las especificaciones. Pruebas de integración (Jenkins o GitHub Actions): Verifica que los distintos módulos del software y su integración en el sistema completo.
<b>2. Validación</b>	Este busca confirmar que el software cumpla con las expectativas y necesidades del usuario final.	Pruebas de validación funcional (TestRail): Asegura que las funciones del software trabajen según lo esperado. Pruebas de regresión (Selenium o Cypress ): Verifica que las modificaciones no afectan las funciones. Validación de desempeño (JMeter): Busca confirmar que el producto funcione bien Validación de seguridad (OWASP ZAP, Burp Suite): Realiza pruebas para asegurar que el software sea seguro y esté protegido contra vulnerabilidades conocidas. Pruebas de aceptación de usuario (TestRail, Jira): Validación final por parte del usuario para verificar que el producto cumple con sus necesidades y expectativas.

## Implementación de propuesta

Una vez establecido los puntos clave de la propuesta, es fundamental definir de manera clara el proceso de implementación de esta en empresas de desarrollo de software que cuente con entornos de equipos reducidos.

**Tabla 12**

### *Fases de implementación*

<b>Fase</b>	<b>Nombre</b>	<b>Descripción</b>
Fase I.	Planificación inicial	En esta fase se definen los objetivos específicos de la propuesta, se asignan los recursos necesarios y se establece la estructura para su implementación, dentro de esta se incluyen la asignación de roles, la identificación de tareas clave, así como la selección de herramientas y metodologías adecuadas.
Fase II.	Implementación	Durante esta fase se lleva a cabo la ejecución de la propuesta, integrando las herramientas seleccionadas y aplicando los procesos establecidos en la fase de planificación, lo cual busca garantizar una implementación fluida y eficiente.
Fase III.	Revaluación y ajustes de propuesta	Una vez implementada la propuesta, se realiza una evaluación exhaustiva de los resultados obtenidos, analizando su efectividad y eficiencia. A partir de esta evaluación, se realizan los ajustes necesarios para optimizar el proceso y asegurar el cumplimiento de los objetivos.

### k) **Cronograma de actividades**

El cronograma de actividades debe ser adaptado según los recursos disponibles y las especificidades del proyecto. A continuación, se presenta un cronograma sugerido que debe ajustarse según las necesidades del equipo y el contexto particular de la empresa.

**Figura 2**

*Cronogramas de actividades*

Actividades	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5
<b>Fase I: Planificación Inicial</b>					
Definición de objetivos y recursos					
Asignación de roles y tareas					
Selección de herramientas necesarias					
<b>Fase II: Implementación</b>					
Ejecución de la propuesta					
Integración de herramientas					
Despliegue de entornos de trabajo					
<b>Fase III: Reevaluación y ajustes</b>					
Evaluación de resultados y ajustes de resultados					

### Costos y beneficios

La implementación de la propuesta para optimizar los procesos de QA en equipos reducidos en empresas de desarrollo de software implica una serie de costos asociados con la ejecución del plan, así como una variedad de beneficios tangibles e intangibles que justifican la inversión y se distribuyen principalmente entre los salarios del personal involucrado y las herramientas necesarias para llevar a cabo las pruebas, así como las actividades de automatización y evaluación.

En cuanto a los salarios, las estimaciones de los costos anuales están basadas en el costo promedio por hora para cada rol dentro del equipo de QA en Costa Rica, considerando una jornada laboral estándar de 8 horas diarias, y se calcularon con base en la información suministrada por el ERI Economic Research Institute (2023). A continuación, se presenta un resumen de los costos anuales de los roles clave dentro del equipo de QA:

**Tabla 13**

*Costos por hora*

Rol	Salario Promedio por hora	Costo Anual
QA Lead	₡ 11,700.26	₡ 22,477,939
QA Automation Engineer	₡9,248	₡17,757,984

<b>Manual QA Tester</b>	€ 6,738	€12,937,420
-------------------------	---------	-------------

Elaborado a partir de ERI Economic Research Institute, 2023.

Estos costos reflejan el salario anual de cada miembro del equipo, pero es importante señalar que los valores pueden variar dependiendo de la cantidad de miembros y la composición del equipo, la cual no es fija. En el caso de un equipo reducido, que inicialmente puede estar compuesto por 4 personas, los costos salariales totales se calcularían multiplicando los valores anteriormente mencionados por el número de personas en cada rol, lo que permitirá obtener una estimación más precisa en función de los roles asignados en cada equipo; es decir, el costo total de los salarios dependerá de cómo se distribuyan los roles (por ejemplo, si un equipo cuenta con más o menos QA Leads o QA Automation Engineers), lo que generará variaciones en el total anual estimado para los salarios del equipo.

Como parte de la propuesta, también deben evaluarse los costos asociados a las herramientas necesarias para realizar las pruebas de calidad. Las herramientas que se emplearán en el proceso de aseguramiento de la calidad son fundamentales para la gestión de pruebas, el seguimiento de los errores, la automatización y la colaboración dentro del equipo de QA. A continuación, se detallan los precios aproximados por usuario para las herramientas más relevantes que se utilizarán en el equipo de QA:

**Tabla 14** *Costos de herramientas*

<b>Herramienta</b>	<b>Costo por Usuario al mes</b>
TestRail	\$35. 00
Jira	\$7.53. 00
GitHub	\$ 21. 00
Burp Suite	\$ 37.50
Asana	\$ 11. 00
Google Space	\$ 12. 00

Microsoft Teams	\$ 4. 00
<b>Total</b>	<b>\$ 128.03</b>

Es importante considerar que estos costos fueron tomados de los sitios web de los diversos proveedores de herramientas, y estos pueden variar dependiendo del plan contratado, las condiciones de suscripción, o incluso variar dependiendo del plazo del pago, si este es mensual o anual.

Es fundamental señalar que no todas las herramientas mencionadas se implementarán necesariamente en la propuesta, ya que se brindan varias opciones a elegir según las necesidades particulares de los equipos de QA y las prioridades del proyecto. En función de los requerimientos específicos de cada fase del ciclo de vida del desarrollo de software, se optará por las herramientas que mejor se alineen con los objetivos de calidad, eficiencia en la ejecución de pruebas y optimización de recursos. De este modo, se busca no solo cumplir con los estándares de calidad establecidos, sino también maximizar la eficiencia operativa y el retorno de la inversión.

### **Conclusiones**

La implementación de esta propuesta de optimización de procesos de QA en equipos reducidos de empresas de desarrollo de software tiene como objetivo generar un impacto positivo en organizaciones con equipos de trabajo pequeños, promoviendo la eficiencia, efectividad y calidad en los productos desarrollados. A través de una planificación y organización estratégica de los equipos de QA, se busca asignar roles clave y responsabilidades que aseguren el cumplimiento de los objetivos del producto.

Además, se destacó la importancia de la división de tareas durante los procesos de control de calidad, con el fin de equilibrar las cargas de trabajo entre los miembros de estos equipos y lograr así una mayor eficiencia laboral en los profesionales encargados de la calidad del software. En este sentido, fue crucial la elaboración de actividades esenciales, como la planificación y diseño de pruebas, así como la ejecución y seguimiento de las mismas, abordando aspectos clave para el desarrollo de la propuesta.

Otro aspecto determinante es la incorporación de metodologías ágiles en entornos de trabajo con equipos reducidos, ya que facilitan la flexibilidad y la mejora continua de los procesos de QA, permitiendo que cada miembro se adapte a los cambios inesperados que surjan durante el desarrollo del software. Esto asegura que la calidad y consistencia se mantengan a lo largo del ciclo de vida del proyecto.

La propuesta pretende ser una estructura sólida y eficiente para los equipos reducidos de QA, optimizando tanto los procesos como los costos y garantizando un producto de calidad, lo que contribuye al éxito y la sostenibilidad a lo largo del proyecto en desarrollo.

## Biografía

- Ahmad, N. (2022). QA Metrics Tutorial: A Comprehensive Guide With Examples And Best Practices. Lambdatest. <https://www.lambdatest.com/learning-hub/qa-metrics>
- Alphonse, M. (2024). Enhancing Software Quality through Early-Phase of Software Verification and Validation Techniques. Available at SSRN 4611404. <https://papers.ssrn.com/sol3/Delivery.cfm?abstractid=4611404>
- Barón. R.B. (2022). Definición del proceso de QA para la ejecución de pruebas de software de la OAS.
- Benavides Vásquez, J. J., Ortiz Parejo, Á. V., & Martínez Bacca, A. V. (2024). *Plan para la aplicación de sinergias metodológicas entre PMI y Scrum en el desarrollo de proyectos de tecnología* (Bachelor's thesis, Especialización en Gerencia de Proyectos-Virtual).
- Farchi, E., & Route, S. (2023). Quality Engineering for Agile and DevOps on the Cloud and Edge. *arXiv preprint arXiv:2302.03651*. <https://arxiv.org/pdf/2302.03651>
- Goericke, S. (2020). *The future of software quality assurance* (p. 257). Springer Nature. <https://library.oapen.org/bitstream/handle/20.500.12657/22847/1/1007314.pdf>
- Laoyan, S. (6 de febrero de 2024). *Cómo mejorar la eficiencia operativa*. Asana. <https://asana.com/es/resources/operational-efficiency>
- López, J. O. H. (2024). *Scrum Master: Fundamentos Actuales*. Jesus Omar Hernandez Lopez.
- Melgarejo Solis, J. M., & Plasencia Olivas, C. I. Las metodologías ágiles y la gestión del conocimiento.
- Navarro-Martínez, F. (2023). Propuesta de mejora del proceso de aseguramiento de la calidad del software para un Grupo Financiero.

Organización Internacional de Normalización (ISO). (2015). ISO 9001:2015. Sistemas de gestión de la calidad — Requisitos. <https://www.iso.org/obp/ui/#iso:std:iso:9001:ed-5:v1:es>

Romeroa, V., Gilb, M., & Panachc, J. I. (2023) Enseñando Scrum desde Diversas Perspectivas de la Ingeniería Informática. En libro de actas: IX Congreso de Innovación Educativa y Docencia en Red. Valencia

Semerano, V. Z., & De Oliveira, L. F. (2024). O Papel Estratégico do Analista de Qualidade (QA) em Equipes SCRUM, Kanban e Scrumban no Desenvolvimento de Software Ágil. *Advances in Global Innovation & Technology*, 2(2), 32-45.

Song, J., Kim, H., & Park, S. (2015). Enhancing Conformance Testing Using Symbolic Execution for Network Protocols. *IEEE Transactions on Reliability*, 64(3), 1024-1037. <https://doi.org/10.1109/TR.2015.2443392>

## Apéndice 4: Revisión de José Miguel Chacón

### Instrucciones para completar el instrumento de revisión por pares

1. Lea el documento titulado: *Optimización de procesos de QA en equipos reducidos Vers- I*
2. Complete la información solicitada en el cuadro de dictamen general.
3. Rellene el cuadro de "Temas por evaluar", clasificando cada uno utilizando la escala de Likert, donde:

1: Totalmente en desacuerdo

2: En desacuerdo

3: Neutral

4: De acuerdo

5: Totalmente de acuerdo

4. Si lo desea, puede agregar comentarios directamente en el documento; sin embargo, esto no es obligatorio.

---

### Instrumento de revisión por pares

#### Propósito del instrumento

Este instrumento está diseñado para evaluar la propuesta titulada Optimización de procesos de QA en equipos reducidos. El objetivo de esta evaluación es identificar posibles debilidades para mejorar la calidad del documento y garantizar su utilidad, viabilidad y realismo.

La propuesta fue enviada junto con este instrumento el mismo día.

<b>DICTAMEN GENERAL DESPUÉS DE LA REVISIÓN (MARQUE UNA X):</b>
--

<input type="checkbox"/> Se recomienda aprobar la propuesta sin correcciones.
---

<input checked="" type="checkbox"/> Se recomienda aprobar la propuesta con correcciones.
--

<input type="checkbox"/> Se recomienda no aprobar la propuesta.
---

#### Temas por evaluar

Evalúe cada aspecto de la propuesta utilizando la escala de Likert (1 a 5). Además, puede incluir observaciones específicas relacionadas con los enunciados evaluados:

### 1. Efectividad para el Control de Calidad

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
Considera que al implementar esta propuesta, los procesos de control de calidad reducirán significativamente las tasas de defecto.					X
Los procesos de control de calidad estarán bien documentados y su trazabilidad será fácil de seguir.					X
Las pruebas manuales repetitivas disminuirán debido a las mejoras en el proceso.					X
Menos errores críticos serán reportados después del lanzamiento, en comparación con proyectos anteriores.					X

### 2. La satisfacción dentro del equipo y la eficiencia

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
Mejorará la comunicación entre los equipos de control de calidad.				X	
Dentro del equipo existirán pautas claras y consistentes respecto a la entrega de informes de errores.					X
Se fomentará el intercambio de conocimientos y el desarrollo de habilidades dentro de los equipos reducidos de control de calidad.					X
Herramientas como Jira, Slack o Microsoft Teams respaldarán de manera eficaz las actividades de control de calidad.					X

### 3. Tiempos de entrega y calidad del producto general

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
Los ciclos de pruebas serán completados dentro de los plazos especificados					X
La retroalimentación de control de calidad se integrará a los ciclos de desarrollo de manera oportuna.					X

La calidad del producto entregado cumplirá con las expectativas del cliente.					X
Las partes interesadas estarán satisfechas con los resultados de los procesos de control de calidad.					X
La optimización de los procesos de control de calidad generará un impacto positivo en la satisfacción general del cliente.					X

#### 4. Metodologías

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
La propuesta presenta de manera clara y detallada los pasos necesarios para crear un departamento funcional de QA en una empresa con equipo reducido.					X
Las metodologías propuestas son apropiadas para los equipos de QA en empresas de desarrollo de software, considerando las limitaciones.					X
La propuesta mantiene una coherencia lógica entre los diversos componentes, incluidas las metodologías.		X			
Aporta soluciones innovadoras y creativas, o bien mejora significativamente los procesos existentes.					X
La propuesta es escalable y puede adaptarse de manera efectiva a equipos más grandes o a diferentes contextos.					X

**Si lo considera pertinente, por favor, incluya sus observaciones, sugerencias o comentarios respecto a la propuesta. Estos pueden abordar cualquier aspecto que considere relevante para mejorar la calidad, la viabilidad o la claridad del documento:**

En la parte de Roles, en equipos de metodologías ágiles se elimina algunas veces el líder de QA, los QA pasan a ser parte del equipo de desarrollo y le reportan directamente al líder del equipo de desarrollo. Existen los roles de QA senior, intermedio y junior. Donde el senior se encarga de enseñar a los demás los procesos actuales, pero no es el jefe como tal. En la parte de actividades yo agregaría la creación de Estrategia de pruebas con su respectiva revisión con los otros QA antes de la creación de Pruebas.

**Nombre completo del revisor:** José Miguel Chacón Araya

**Cargo actual del revisor:** Senior QA Analyst

## Apéndice 5: Revisión de Lawrence Fowks Peña

### Instrucciones para completar el instrumento de revisión por pares

1. Lea el documento titulado: *Optimización de procesos de QA en equipos reducidos Vers- I*
2. Complete la información solicitada en el cuadro de dictamen general.
3. Rellene el cuadro de "Temas por evaluar", clasificando cada uno utilizando la escala de Likert, donde:
  - 1: Totalmente en desacuerdo
  - 2: En desacuerdo
  - 3: Neutral
  - 4: De acuerdo
  - 5: Totalmente de acuerdo
4. Si lo desea, puede agregar comentarios directamente en el documento; sin embargo, esto no es obligatorio.

---

### Instrumento de revisión por pares

#### Propósito del instrumento

Este instrumento está diseñado para evaluar la propuesta titulada Optimización de procesos de QA en equipos reducidos. El objetivo de esta evaluación es identificar posibles debilidades para mejorar la calidad del documento y garantizar su utilidad, viabilidad y realismo.

La propuesta fue enviada junto con este instrumento el mismo día.

#### **DICTAMEN GENERAL DESPUÉS DE LA REVISIÓN (MARQUE UNA X):**

- Se recomienda aprobar la propuesta sin correcciones.
- Se recomienda aprobar la propuesta con correcciones.
- Se recomienda no aprobar la propuesta.

#### Temas por evaluar

Evalúe cada aspecto de la propuesta utilizando la escala de Likert (1 a 5). Además, puede incluir observaciones específicas relacionadas con los enunciados evaluados:

## 1. Efectividad para el Control de Calidad

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
Considera que al implementar esta propuesta, los procesos de control de calidad reducirán significativamente las tasas de defecto.				✓	
Los procesos de control de calidad estarán bien documentados y su trazabilidad será fácil de seguir.					✓
Las pruebas manuales repetitivas disminuirán debido a las mejoras en el proceso.					✓
Menos errores críticos serán reportados después del lanzamiento, en comparación con proyectos anteriores.				✓	

## 2. La satisfacción dentro del equipo y la eficiencia

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
Mejorará la comunicación entre los equipos de control de calidad.					✓
Dentro del equipo existirán pautas claras y consistentes respecto a la entrega de informes de errores.					✓
Se fomentará el intercambio de conocimientos y el desarrollo de habilidades dentro de los equipos reducidos de control de calidad.					✓
Herramientas como Jira, Slack o Microsoft Teams respaldarán de manera eficaz las actividades de control de calidad.			•	✓	

## 3. Tiempos de entrega y calidad del producto general

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
Los ciclos de pruebas serán completados dentro de los plazos especificados			✓		
La retroalimentación de control de calidad se integrará a los ciclos de desarrollo de manera oportuna.				✓	
La calidad del producto entregado cumplirá con las expectativas del cliente.				✓	
Las partes interesadas estarán satisfechas con los resultados de los procesos de control de calidad.					✓

La optimización de los procesos de control de calidad generará un impacto positivo en la satisfacción general del cliente.					✓
--	--	--	--	--	---

#### 4. Metodologías

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
La propuesta presenta de manera clara y detallada los pasos necesarios para crear un departamento funcional de QA en una empresa con equipo reducido.				✓	
Las metodologías propuestas son apropiadas para los equipos de QA en empresas de desarrollo de software, considerando las limitaciones.				✓	
La propuesta mantiene una coherencia lógica entre los diversos componentes, incluidas las metodologías.				✓	
Aporta soluciones innovadoras y creativas, o bien mejora significativamente los procesos existentes.				✓	
La propuesta es escalable y puede adaptarse de manera efectiva a equipos más grandes o a diferentes contextos.					✓

**Si lo considera pertinente, por favor, incluya sus observaciones, sugerencias o comentarios respecto a la propuesta. Estos pueden abordar cualquier aspecto que considere relevante para mejorar la calidad, la viabilidad o la claridad del documento:**

En el ciclo de desarrollo de software, si bien los aspectos mencionados como objetivos el departamento de QA está involucrado, otros departamentos y factores pueden afectar estos objetivos, puesto que el software es un aspecto integral. Es necesario también aparte de implementar una propuesta para QA, extender normas o acciones de interacción con las otras áreas involucradas y que esto también esté documentado de alguna manera. Ayudando así con mayor transparencia y comunicación entre las partes y a su vez mejorando la aceptación de QA como rol involucrado en el proceso.

Por otra parte, es importante demostrar por que el rol de QA es tan relevante, muchas veces existe una resistencia al cambio en aspectos de QA en empresas no tan maduras, por consecuencia limitan mucho el presupuesto para este tipo de talento humano. Se debe manifestar en la propuesta con datos tangibles por que un departamento de QA altamente

integrado y maduro con la cantidad de recursos necesarios reducirá significativamente los costos futuros que un software en producción podría generar y demostrar como la calidad del mismo es altamente confiable.

**Nombre completo del revisor:** Lawrence Fowks Peña

**Cargo actual del revisor:** Staff Software Engineer / Docente Universitario

## Apéndice 5: Revisión de Alex Daniel Villegas Carranza

### Instrucciones para completar el instrumento de revisión por pares

1. Lea el documento titulado: *Optimización de procesos de QA en equipos reducidos Vers- I*
2. Complete la información solicitada en el cuadro de dictamen general.
3. Rellene el cuadro de "Temas por evaluar", clasificando cada uno utilizando la escala de

Likert, donde:

- 1: Totalmente en desacuerdo
  - 2: En desacuerdo
  - 3: Neutral
  - 4: De acuerdo
  - 5: Totalmente de acuerdo
4. Si lo desea, puede agregar comentarios directamente en el documento; sin embargo, esto no es obligatorio.

---

### Instrumento de revisión por pares

#### Propósito del instrumento

Este instrumento está diseñado para evaluar la propuesta titulada Optimización de procesos de QA en equipos reducidos. El objetivo de esta evaluación es identificar posibles debilidades para mejorar la calidad del documento y garantizar su utilidad, viabilidad y realismo.

La propuesta fue enviada junto con este instrumento el mismo día.

#### **DICTAMEN GENERAL DESPUÉS DE LA REVISIÓN (MARQUE UNA X):**

- Se recomienda aprobar la propuesta sin correcciones.
- Se recomienda aprobar la propuesta con correcciones.
- Se recomienda no aprobar la propuesta.

## Temas por evaluar

Evalúe cada aspecto de la propuesta utilizando la escala de Likert (1 a 5). Además, puede incluir observaciones específicas relacionadas con los enunciados evaluados:

### 1. Efectividad para el Control de Calidad

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
Considera que al implementar esta propuesta, los procesos de control de calidad reducirán significativamente las tasas de defecto.					X
Los procesos de control de calidad estarán bien documentados y su trazabilidad será fácil de seguir.					X
Las pruebas manuales repetitivas disminuirán debido a las mejoras en el proceso.					X
Menos errores críticos serán reportados después del lanzamiento, en comparación con proyectos anteriores.				X	

### 2. La satisfacción dentro del equipo y la eficiencia

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
Mejorará la comunicación entre los equipos de control de calidad.			X		
Dentro del equipo existirán pautas claras y consistentes respecto a la entrega de informes de errores.					X
Se fomentará el intercambio de conocimientos y el desarrollo de habilidades dentro de los equipos reducidos de control de calidad.					X
Herramientas como Jira, Slack o Microsoft Teams respaldarán de manera eficaz las actividades de control de calidad.					X

### 3. Tiempos de entrega y calidad del producto general

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
Los ciclos de pruebas serán completados dentro de los plazos especificados				X	
La retroalimentación de control de calidad se integrará a los ciclos de desarrollo de manera oportuna.				X	
La calidad del producto entregado cumplirá con las expectativas del cliente.				X	
Las partes interesadas estarán satisfechas con los resultados de los procesos de control de calidad.				X	
La optimización de los procesos de control de calidad generará un impacto positivo en la satisfacción general del cliente.					X

#### 4. Metodologías

	Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo
La propuesta presenta de manera clara y detallada los pasos necesarios para crear un departamento funcional de QA en una empresa con equipo reducido.			X		
Las metodologías propuestas son apropiadas para los equipos de QA en empresas de desarrollo de software, considerando las limitaciones.			X		
La propuesta mantiene una coherencia lógica entre los diversos componentes, incluidas las metodologías.				X	
Aporta soluciones innovadoras y creativas, o bien mejora significativamente los procesos existentes.		X			
La propuesta es escalable y puede adaptarse de manera efectiva a equipos más grandes o a diferentes contextos.				X	

**Si lo considera pertinente, por favor, incluya sus observaciones, sugerencias o comentarios respecto a la propuesta. Estos pueden abordar cualquier aspecto que considere relevante para mejorar la calidad, la viabilidad o la claridad del documento:**

“Me parece que faltan métricas claras en esta propuesta para entender cómo se estará midiendo la optimización, para entender si la propuesta es exitosa o no al implementarse debe hacerse algún tipo de comparación (detallar los indicadores claves que se estarán proponiendo a mejorar). Por ejemplo:

**Eficiencia, en términos de tiempo de ejecución de pruebas**

**Tiempo promedio necesario para completar todas las pruebas de un ciclo.**

(Objetivo: Reducir este tiempo a través de automatización).

**Comparación del tiempo del equipo de desarrollo antes, y después de la aplicación de la propuesta.**

Así mismo se podrían incluir más indicadores, por ejemplo en cuales aspectos se mejorará la calidad, etc.

Me parece que la propuesta podría replantearse de otra forma, tal vez como **Propuesta de un**

**Estándar para Optimizar Procesos de QA en Equipos de Desarrollo Reducidos.**

De esta forma, el título describe que las prácticas que van a proponer, son la suma del rigor académico respecto a las mejores prácticas sobre QA y respaldadas por la experticia de equipos de desarrollo pequeños que hayan generado estos indicadores y puedan proveer claridad sobre las mejoras aplicadas, en términos de indicadores clave que mejoran luego de haber sido implementados. Todos estos procesos que mencionan en la propuesta ya son ampliamente aceptados en la industria y han sido puestos a prueba y demostrado mejorar a equipos de desarrollo, proponer la optimización a través de estas prácticas directamente no es innovador, en su lugar una propuesta de un estándar para optimización de procesos, que promueva la detección de defectos en etapas tempranas del ciclo de desarrollo de forma proactiva, que ejercite pruebas considerando las limitaciones de los equipos, que sea flexible a cambios de los requerimientos y necesidades de cada equipo. Ese estándar podría ser utilizado como documento de referencia para fundar equipos de desarrollo pequeños con QA que a pesar de sus limitaciones sea bastante efectivo.”

**Nombre completo del revisor:** Alex Daniel Villegas Carranza

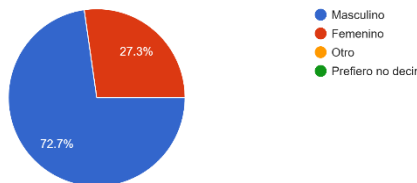
**Cargo actual del revisor:** Profesor e Investigador de la Universidad Nacional (UNA) - Costa Rica/  
Consultor e Ingeniero de Software / M.Sc. en Ciberseguridad.

## **Anexos**

## Anexo 1: Respuestas del cuestionario

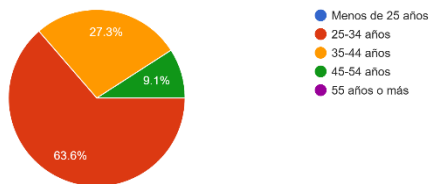
¿Con cuál de los siguientes géneros se identifica usted?

1. ¿Con cuál de los siguientes géneros se identifica usted?  
11 respuestas



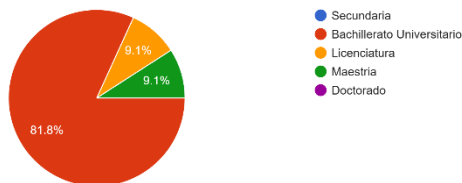
¿En cuál de los siguientes rangos de edad se encuentra usted?

2. ¿En cuál de los siguientes rangos de edad se encuentra usted?  
11 respuestas



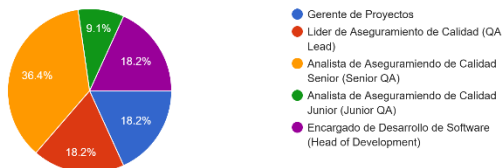
¿Cuál es su nivel académico actualmente?

3. ¿Cuál es su nivel académico actualmente?  
11 respuestas



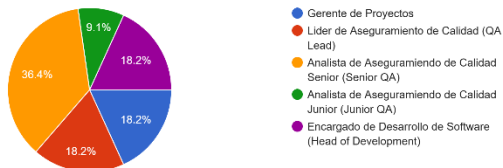
¿Cuántos años de experiencia tiene usted en el área del aseguramiento de la calidad?

5. ¿Cuál es el rol que usted actualmente desempeña dentro de la empresa de desarrollo de software?  
11 respuestas



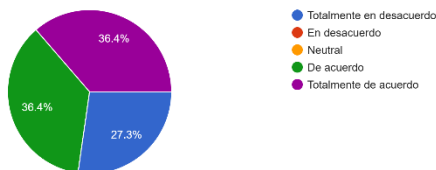
¿Cuál es el rol que usted actualmente desempeña dentro de la empresa de desarrollo de software?

5. ¿Cuál es el rol que usted actualmente desempeña dentro de la empresa de desarrollo de software?  
11 respuestas



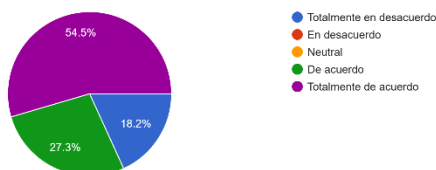
¿Qué tan de acuerdo estás con la siguiente afirmación? En un equipo de calidad de software, a pesar de ser un equipo reducido, se demuestra eficiencia en la detección y en la solución de defectos en el software.

6. ¿Qué tan de acuerdo estás con la siguiente afirmación? En un equipo de calidad de software, a pesar de ser un equipo reducido, se demuestra eficiencia en la detección y en la solución de defectos en el software.  
11 respuestas



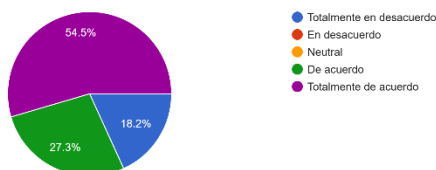
¿Qué tan importante considera usted la experiencia previa en calidad de software para el desempeño efectivo en un equipo de calidad de software reducido?

10. ¿Qué tan importante considera usted la experiencia previa en calidad de software para el desempeño efectivo en un equipo de calidad de software reducido?  
11 respuestas



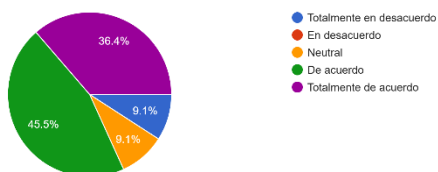
¿En qué medida está usted de acuerdo con la siguiente afirmación: 'Contar con personal experimentado en un equipo de calidad de software reducido contribuye a la mejora de los procesos de prueba'?

11. ¿En qué medida está usted de acuerdo con la siguiente afirmación: 'Contar con personal experimentado en un equipo de calidad de software ...ntribuye a la mejora de los procesos de prueba'?  
11 respuestas



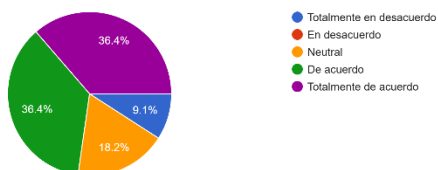
¿En qué medida está usted de acuerdo con la siguiente afirmación: 'La optimización de procesos en un equipo de calidad de software reducido depende en gran parte de la capacidad del equipo para adaptarse a nuevas herramientas y metodologías'?

12. ¿En qué medida está usted de acuerdo con la siguiente afirmación: 'La optimización de procesos en un equipo de calidad de software reducido...daptarse a nuevas herramientas y metodologías'?  
11 respuestas



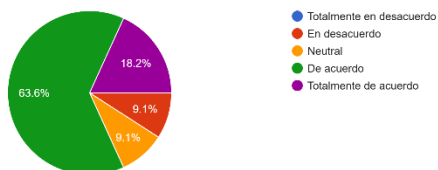
¿En qué medida está usted de acuerdo con la siguiente afirmación: 'La automatización de pruebas puede compensar las limitaciones de recursos dentro de los equipos de calidad de software reducidos'?

14. ¿En qué medida está usted de acuerdo con la siguiente afirmación: 'La automatización de pruebas puede compensar las limitaciones de recur...e los equipos de calidad de software reducidos'?  
11 respuestas



¿Qué tan satisfecho está usted con la actual implementación de los procesos de control de calidad en un equipo reducido de su organización?

15. ¿Qué tan satisfecho está usted con la actual implementación de los procesos de control de calidad en un equipo reducido de su organización?  
11 respuestas



# Anexo 2: Career Path

The image displays a career path matrix for Software Testing roles, organized into four rows representing different levels of expertise: QA, QA, QA, and Senior. Each row contains four columns representing different skill areas: Agile Methodologies, Automated Testing, Bug Life Cycle, and Quality Assurance. Each cell in the matrix provides a detailed description of the role's responsibilities, evaluation methods, and relevant documentation links.

**Row 1: QA**

- Agile Methodologies:** Includes SCRUM and Kanban. Evaluation Methodology: Candidates must know about the Agile methodologies.
- Automated Testing:** Includes Programming languages and Automation frameworks. Evaluation Methodology: Candidates must know basic concepts about automated tests and at least the principal tools.
- Bug Life Cycle:** Includes Writing bug reports, Know best for bug management, and Bug Life Cycle. Evaluation Methodology: Candidates must know, analyze and document bugs including all relevant information (logs, screenshots, and detailed steps to reproduce it) in order to get testing results.
- Quality Assurance:** Includes Checkpoint, Plan, Test, Implement, Monitor, and Problem solving. Evaluation Methodology: Candidates must know the basic definitions of Software Quality.

**Row 2: QA**

- Agile Methodologies:** Includes SCRUM and Kanban. Evaluation Methodology: Candidates must know about the Agile methodologies.
- Automated Testing:** Includes Programming languages and Automation frameworks. Evaluation Methodology: Candidates must know basic concepts about automated tests and at least the principal tools.
- Bug Life Cycle:** Includes Writing bug reports, Know best for bug management, and Bug Life Cycle. Evaluation Methodology: Candidates must know, analyze and document bugs including all relevant information (logs, screenshots, and detailed steps to reproduce it) in order to get testing results.
- Quality Assurance:** Includes Checkpoint, Plan, Test, Implement, Monitor, and Problem solving. Evaluation Methodology: Candidates must know the basic definitions of Software Quality.

**Row 3: QA**

- Agile Methodologies:** Includes SCRUM and Kanban. Evaluation Methodology: Candidates must know about the Agile methodologies.
- Automated Testing:** Includes Programming languages and Automation frameworks. Evaluation Methodology: Candidates must know basic concepts about automated tests and at least the principal tools.
- Bug Life Cycle:** Includes Writing bug reports, Know best for bug management, and Bug Life Cycle. Evaluation Methodology: Candidates must know, analyze and document bugs including all relevant information (logs, screenshots, and detailed steps to reproduce it) in order to get testing results.
- Quality Assurance:** Includes Checkpoint, Plan, Test, Implement, Monitor, and Problem solving. Evaluation Methodology: Candidates must know the basic definitions of Software Quality.

**Row 4: Senior**

- Agile Methodologies:** Includes SCRUM and Kanban. Evaluation Methodology: Candidates must know about the Agile methodologies.
- Automated Testing:** Includes Programming languages and Automation frameworks. Evaluation Methodology: Candidates must know basic concepts about automated tests and at least the principal tools.
- Bug Life Cycle:** Includes Writing bug reports, Know best for bug management, and Bug Life Cycle. Evaluation Methodology: Candidates must know, analyze and document bugs including all relevant information (logs, screenshots, and detailed steps to reproduce it) in order to get testing results.
- Quality Assurance:** Includes Checkpoint, Plan, Test, Implement, Monitor, and Problem solving. Evaluation Methodology: Candidates must know the basic definitions of Software Quality.