

Mejoramiento del plan estratégico de los procesos de calidad en el ciclo de vida del desarrollo de software en el área de pruebas y certificación para la empresa PDT; por medio de las herramientas de la Guía del PMBOK®

Sindy Vanessa Vergel Lafaurie

Milton Alfredo Gomez

Asesor

Docente. Denis Manuel Roa García

Universidad Nacional Abierta y a Distancia – UNAD

Escuela de Ciencias, Básicas, Tecnología e Ingeniería ECBTI

Gerencia de Proyectos

2025

Nota de aceptación:

Firma de Jurado 1

Firma de Jurado 2

Resumen

La presente investigación tuvo como propósito central la optimización del plan estratégico del área de pruebas y certificación en una empresa tecnológica colombiana, mediante la aplicación rigurosa de los principios y herramientas establecidos en la Guía del PMBOK® (2021). Se evidenció que la aplicación de metodologías y procesos relacionados con el aseguramiento de calidad del software no se realizó de forma completa ni estructurada, generando diversas deficiencias operativas, tales como una insuficiente planificación estratégica, ausencia recurrente de documentación actualizada en los repositorios institucionales y una sistemática pérdida de lecciones aprendidas. Lo anterior trajo como consecuencia directa el incremento significativo de errores críticos en producción, sobrecostos operativos derivados del mantenimiento correctivo, y afectaciones profundas en la confiabilidad de los productos tecnológicos entregados a los usuarios finales. Se aplicó un enfoque híbrido de ejecución combinando marcos de trabajo ágiles como Scrum con elementos del modelo tradicional en cascada, buscando así una gestión flexible, proactiva y adaptativa a las necesidades organizacionales y técnicas del proyecto. Como resultado de la implementación efectiva de estas estrategias, se logró fortalecer considerablemente la trazabilidad de defectos, reducir la pérdida del conocimiento técnico, optimizar la documentación operativa, disminuir la rotación del talento humano, y garantizar entregas tecnológicas alineadas con altos estándares internacionales de calidad, seguridad y rendimiento, elevando la competitividad y sostenibilidad operativa de la empresa.

Palabras clave: calidad, certificaciones, gestión de proyectos, Guía del PMBOK®, pruebas.

Abstract

The main purpose of this research was the optimization of the strategic plan of the testing and certification area in a Colombian technology company, through the rigorous application of the principles and tools established in the PMBOK® Guide (2021). It was evident that the application of methodologies and processes related to software quality assurance was not carried out in a complete or structured way, generating various operational deficiencies, such as insufficient strategic planning, recurrent absence of updated documentation in institutional repositories and a systematic loss of lessons learned. This directly resulted in a significant increase in critical errors in production, operating cost overruns derived from corrective maintenance, and profound effects on the reliability of technological products delivered to end users. A hybrid approach to execution was applied, combining agile frameworks such as Scrum with elements of the traditional waterfall model, thus seeking flexible, proactive and adaptive management to the organizational and technical needs of the project. As a result of the effective implementation of these strategies, it was possible to considerably strengthen the traceability of defects, reduce the loss of technical knowledge, optimize operational documentation, reduce the turnover of human talent, and guarantee technological deliveries aligned with high international standards of quality, safety and performance, increasing the competitiveness and operational sustainability of the company.

Keywords: quality, certifications, project management, PMBOK® Guide, tests.

Tabla de Contenido

Resumen.....	3
Abstract.....	4
Introducción	17
Definición del Problema	20
Preguntas Problema	23
Justificación	25
Objetivos.....	29
Objetivo General.....	29
Objetivos Específicos.....	29
Marco Teórico.....	31
Fundamentos de la Calidad en el Desarrollo de Software	32
Concepto de Calidad en Software.....	32
Evolución Histórica de la Calidad en Software	37
Relación entre Calidad de Software y Competitividad en la Industria Tecnológica	39
Principales Modelos y Estándares para la Gestión de Calidad en Software.....	40
Pruebas y Certificación de Calidad en Software.....	43
Metodologías Especializadas en Pruebas y Certificación de Software	45
Marcos de trabajo Ágiles en la Gestión de Calidad de Software.....	47
Gestión de Proyectos en el Desarrollo de Software.....	49
Gestión de Proyectos Tradicional (Ciclo en Cascada).....	49
Metodologías Ágiles (Scrum, Kanban, DevOps)	50

Herramientas Ágiles (Según Guía del PMBOK® 2021)	52
Enfoque Híbrido en Proyectos Tecnológicos	54
La Guía del PMBOK® y su Aplicación al Aseguramiento de Calidad en Software	55
Estructura Conceptual de la Guía del PMBOK®	56
Dominios de Desempeño de la Guía del PMBOK®	57
Herramientas en la Toma de Decisiones.....	58
Estrategias de Optimización en Procesos de Calidad de Software	61
Estado del Arte: Diagnóstico de la Empresa PDT	63
Marco Legal y Normativo en Calidad de Software	65
Metodología	69
Enfoque de Trabajo.....	69
Diseño de Investigación.....	70
Tipo de Estudio.....	72
Fases del Marco de Trabajo	73
Técnicas, Herramientas y Artefactos de la Guía del PMBOK®.....	74
Población y Muestra	79
Técnicas de Recolección de Datos.....	81
Encuestas Estructuradas.....	81
Entrevistas Semiestructuradas	82
Revisión de Documentación Interna, Procesos y Lecciones Aprendidas	82
Sesiones Colaborativas y Talleres con los Equipos de QA y Desarrollo	83
Validación de la Encuesta.....	83
Cálculo del Alfa de Cronbach.....	85

Análisis de Resultados	87
Diagnóstico de las Deficiencias Operacionales del Área de Pruebas de Software	87
Procesos de Calidad Dentro del Ciclo de Vida del Desarrollo de Software	126
Estrategias de Optimización para los Procesos de Calidad	135
Evidencia de Optimización	145
Socialización de Resultados con los Involucrados	146
Discusión.....	157
Contribuciones	161
Bibliografía	163
Apéndices.....	173

Lista de Tablas

Tabla 1	<i>Evolución del Concepto de Calidad en el Desarrollo de Software</i>	35
Tabla 2	<i>Herramientas Ágiles en Gestión de Calidad de Software</i>	53
Tabla 3	<i>Dominios de Desempeño</i>	57
Tabla 4	<i>Estrategias de Optimización en Procesos de Calidad de Software</i>	62
Tabla 5	<i>Marco Normativo en Calidad de Software</i>	66
Tabla 6	<i>Aplicación de la Guía del PMBOK® por Fase del Proyecto</i>	75
Tabla 7	<i>Relación entre Objetivos Específicos, Fases del Proyecto</i>	78
Tabla 8	<i>Flujo de Validación de la Encuesta</i>	84
Tabla 9	<i>Nivel de Fiabilidad de Alfa de Cronbach</i>	85
Tabla 10	<i>Roles de los Colaboradores Encuestados</i>	88
Tabla 11	<i>Matriz de Prácticas Generales en el Área de Pruebas de Software de PDT</i>	89
Tabla 12	<i>Preguntas Asociadas a la Encuesta</i>	94
Tabla 13	<i>Factores Clave para el Éxito de los Proyectos de Software</i>	95
Tabla 14	<i>Principales Problemas Identificados en los Proyectos de Software</i>	97
Tabla 15	<i>Principales Problemas Asociados a las Fábricas de Software (Proveedores)</i>	99
Tabla 16	<i>Problemas Identificados en la Fase de Planeación del Proceso de Calidad</i>	101
Tabla 17	<i>Problemas Identificados en la Fase de Ejecución del Proceso de Calidad</i>	104
Tabla 18	<i>Matriz DOFA</i>	124
Tabla 19	<i>Análisis de Frecuencia de Problemas Críticos</i>	132
Tabla 20	<i>Matriz MEFÉ (Matriz de Evaluación de Factores Externos)</i>	133
Tabla 21	<i>Matriz MEFI (Matriz de Evaluación de Factores Internos)</i>	134
Tabla 22	<i>Estrategias</i>	136

Tabla 23 <i>Matriz de Impacto vs. Esfuerzo</i>	139
Tabla 24 <i>Roadmap de Implementación por Trimestres</i>	141
Tabla 25 <i>Resumen de Acta de Socialización de Resultados</i>	152

Lista de Figuras

Figura 1	<i>Evolución de la Calidad en Software</i>	38
Figura 2	<i>Relación entre Calidad del Software y Competitividad en la Industria Tecnológica..</i>	42
Figura 3	<i>Metodologías de Pruebas de Software</i>	46
Figura 4	<i>Modelo de EDT</i>	48
Figura 5	<i>Los 12 Principios de la Guía del PMBOK® Séptima Edición.....</i>	56
Figura 6	<i>Modelos, Métodos, Artefactos de la Guía del PMBOK®</i>	59
Figura 7	<i>Fases del Marco de Trabajo</i>	73
Figura 8	<i>Segmentación de Equipos por Roles</i>	87
Figura 9	<i>Mapa de Interesados - Reunión de Apertura del Proyecto</i>	93
Figura 10	<i>¿Qué Tan Importante es un Proyecto?.....</i>	95
Figura 11	<i>¿Cuáles son los Principales Problemas de un Proyecto?.....</i>	97
Figura 12	<i>¿Cuáles son los Problemas de las Fábricas de Software (Proveedores)?.....</i>	99
Figura 13	<i>¿Cuáles son los Problemas del Proceso de Calidad de Software - Planeación?</i>	101
Figura 14	<i>¿Cuáles son los Problemas del Proceso de Calidad de Software - Ejecuciones? ...</i>	103
Figura 15	<i>Mapa Relacional de Códigos</i>	107
Figura 16	<i>Alineación entre Roles.....</i>	108
Figura 17	<i>Barreras para la Calidad.....</i>	111
Figura 18	<i>Cultura de Calidad.....</i>	113
Figura 19	<i>Documentación y Trazabilidad</i>	115
Figura 20	<i>Impacto en la Rotación del Personal</i>	117
Figura 21	<i>Métodos y Herramientas Utilizadas.....</i>	119
Figura 22	<i>Planificación de Pruebas</i>	121

Figura 23 <i>Tormenta de Ideas</i>	127
Figura 24 <i>Diagrama de Afinidad</i>	128
Figura 25 <i>Diagrama de Ishikawa</i>	129
Figura 26 <i>Diagrama de Pareto</i>	131
Figura 28 <i>Matriz de Impacto - Esfuerzo de las Estrategias</i>	138
Figura 29 <i>Mapa de Impacto vs. Esfuerzo de las Alternativas</i>	140
Figura 30 <i>Roadmap de Implementación</i>	143
Figura 31 <i>Optimización del Plan Estratégico</i>	146
Figura 32 <i>Poster - Conferencia Empresarial</i>	149
Figura 33 <i>Conferencia - Expotech 2025</i>	150
Figura 34 <i>Información Oficial de EXPOCH 2025</i>	151

Lista de Apéndices

Apéndice A <i>Acta de Reunión N.1</i>	173
Apéndice B <i>Acta de Reunión N.1</i>	176
Apéndice C <i>Respuestas de Entrevistas Semiestructuradas</i>	179
Apéndice D <i>Certificado de Participación como Conferencistas</i>	196

Glosario

Aseguramiento de la Calidad del Software (SQA - Software Quality Assurance): Conjunto de procesos y metodologías diseñadas para garantizar la calidad del software a lo largo de su ciclo de vida, asegurando que cumpla con los estándares, requisitos y expectativas de los usuarios (IEEE, 2021) (IEEE Std 730-2022).

Automatización de Pruebas: Implementación de herramientas y frameworks para ejecutar pruebas de manera automatizada, reduciendo errores humanos y optimizando tiempos de validación (ISTQB, 2023; Glossary of Testing Terms. International Software Testing Qualifications Board).

Ciclo de Vida del Desarrollo de Software (SDLC - Software Development Life Cycle): Modelo estructurado que define las fases desde la planificación hasta la implementación y mantenimiento del software (Sommerville, 2016). Software Engineering (10th ed.).

CMMI (Capability Maturity Model Integration): Modelo que evalúa la madurez de procesos organizacionales (CMMI Institute, 2018). CMMI for Development, Version 2.0.

DevOps: Metodología que integra desarrollo (Dev) y operaciones (Ops) para mejorar la entrega continua y la eficiencia en la implementación del software (Project Management Institute, 2021). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition.

Estructura de Desglose del Trabajo (EDT - Work Breakdown Structure): Documento utilizado en gestión de proyectos para descomponer tareas en unidades manejables y facilitar su ejecución organizada (Project Management Institute, 2021). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition.

Frameworks de Pruebas: Herramientas que facilitan la ejecución automatizada de pruebas, tales como: a) Selenium: Framework para automatización de pruebas web; b) JMeter: Herramienta para pruebas de carga y rendimiento; c) JUnit: Framework para pruebas unitarias en Java (Garousi et al., 2016).

Gestión del Conocimiento en Software: Uso de repositorios, documentación estructurada y metodologías de transferencia de conocimientos dentro de equipos de desarrollo y pruebas (Project Management Institute, 2021). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition.

Gestión del Riesgo en Desarrollo de Software: Repositorio para la identificación y mitigación de factores que pueden afectar la calidad del software, tales como fallos en la integración, inestabilidad del sistema y vulnerabilidades de seguridad (Project Management Institute, 2021). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition; Pressman & Maxim (2014).

Integración y Despliegue Continuo (CI/CD - Continuous Integration/Continuous Deployment): Práctica de desarrollo que permite la integración y entrega continua de software mediante automatización en la validación y despliegue de código (Humble & Farley, 2010).

ISO/IEC 25010: Estándar que define características de calidad en software (ISO/IEC, 2011). ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.

Metodologías Ágiles: Enfoques iterativos e incrementales aplicados al desarrollo de software, tales como: Scrum: Framework basado en sprints, roles específicos y revisiones constantes; Kanban: Método visual para la gestión del flujo de trabajo y optimización de procesos (Schwaber & Sutherland, 2020). The Scrum Guide.

Modelos de Madurez en Calidad del Software: Estructuras diseñadas para medir y mejorar la calidad en procesos de desarrollo (CMMI Institute. (2018). CMMI for Development, Version 2.0.).

MTTF (Mean Time to Failure): Tiempo promedio entre fallos en un sistema estable (Pham, H. (2012). System Software Reliability. Springer).

MTTR (Mean Time to Repair): Promedio de tiempo requerido para corregir un defecto (O'Connor & Kleyner, 2012). Practical Reliability Engineering. Wiley.).

Optimización de Costos en el Aseguramiento de Calidad: Estrategias para reducir gastos operativos sin comprometer la calidad, incluyendo automatización de pruebas, gestión eficiente de defectos y reducción de reprocesos (Project Management Institute, 2021). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition.

Pruebas de Carga y Estrés (Testing): Evalúan la capacidad del software ante altos volúmenes de usuarios (Myers et al., 2011); (ISTQB, 2023).

Pruebas de Integración (Testing): Evaluación del correcto funcionamiento entre módulos (ISTQB, 2023).

Pruebas de Regresión (Testing): Aseguran que nuevas modificaciones no afecten funcionalidades previas (ISTQB, 2023).

Pruebas de Seguridad (Testing): Identifican vulnerabilidades y riesgos de acceso no autorizado (ISTQB, 2023).

Pruebas Unitarias (Testing): Verifican el correcto funcionamiento de componentes individuales (ISTQB, 2023).

Six Sigma: Metodología basada en datos para reducir defectos y mejorar la eficiencia.

Tasa de defectos: Relación entre fallos detectados y código entregado (Pyzdek & Keller, 2014).

Testing de Software: Proceso de validación y verificación que permite identificar defectos en un sistema antes de su despliegue en producción (ISTQB, 2023).

Trazabilidad de Requisitos: Capacidad para rastrear la relación entre requisitos definidos y las pruebas ejecutadas para su validación, asegurando alineación con las necesidades del usuario (Gotel & Finkelstein, 1994).

Introducción

¿Es posible alcanzar la excelencia operativa en el desarrollo de software sin estructurar una estrategia rigurosa para la gestión de calidad durante todo su ciclo de vida? La presente investigación, titulada “*Mejoramiento del plan estratégico de los procesos de calidad en el ciclo de vida del desarrollo de software*”, abordó de forma exhaustiva esta interrogante a partir de un estudio aplicado al área de pruebas y certificación en una empresa colombiana especializada en tecnología financiera, denominada en este trabajo académico bajo el seudónimo PDT, asegurando el cumplimiento de la Ley de Protección de Datos Personales (Ley 1581 de 2012)¹. En este contexto, se analizaron sistemáticamente factores críticos tales como deficiencias estructurales en los procesos de testing, alta incidencia de defectos en entornos productivos, insuficiencia en la descentralización del conocimiento técnico, así como los elevados sobrecostos operativos generados por pruebas manuales redundantes.

En términos estratégicos, la investigación contempló la adopción integral de herramientas avanzadas para la gestión de la calidad según estándares definidos por ISTQB e IEEE, enfatizando especialmente en metodologías como la generación automatizada de casos de prueba desde la etapa de requisitos, pruebas continuas de regresión, carga y estrés mediante frameworks robustos, complementadas con técnicas manuales fortalecidas; además, se propuso una transformación organizacional en PDT mediante el establecimiento de planes formativos especializados, orientados a consolidar la capacidad técnica del talento humano en las fábricas de software, incentivando así la creación de planes de carrera que mitigaran la rotación y promovieran la cohesión interna. Esto permitió consolidar una cultura organizacional orientada

¹ **Congreso de Colombia.** (2012, 17 de octubre). *Ley 1581 de 2012: Por la cual se dictan disposiciones generales para la protección de datos personales.*

hacia la excelencia, alineada con la misión institucional de PDT, basada en proveer soluciones innovadoras y seguras en tecnología financiera, impulsadas por estándares éticos elevados, excelencia operativa, integridad y compromiso hacia la mejora continua.

La justificación de este estudio se sustentó en evidencias concretas sobre la importancia crítica de gestionar estratégicamente el conocimiento técnico y operacional dentro del proceso de calidad, especialmente mediante la simplificación y estandarización de la documentación asociada a pruebas; con ello se buscó prevenir la repetición recurrente de errores críticos, optimizar la productividad técnica del equipo humano y generar altos índices de confianza sobre las versiones entregadas en producción. Además, se analizó cómo la integración efectiva entre técnicas manuales y automatizadas—mediante herramientas tecnológicas especializadas—contribuyó a disminuir sustancialmente los tiempos de entrega, incrementar la eficiencia global del ciclo de vida del software, y finalmente, potenciar la rentabilidad y sostenibilidad financiera de PDT.

Además, este documento académico se estructuró en capítulos: inicialmente, el capítulo uno incluyó la descripción detallada del problema investigado, planteando preguntas críticas que guiaron el análisis hacia la identificación clara y precisa de los factores que afectaron la calidad operativa en PDT; posteriormente, se presentó la justificación del proyecto, sustentada mediante argumentos sólidos acerca del impacto financiero y estratégico que tuvo la ausencia de metodologías avanzadas en pruebas y certificación, resaltando especialmente la importancia de la prevención temprana de errores. De esta manera, se definieron bases sólidas para plantear tanto el objetivo general como los específicos que orientaron la investigación, estableciendo una ruta estratégica hacia el fortalecimiento del aseguramiento de calidad.

Posteriormente, en el capítulo del Marco Teórico, se desarrollaron conceptos especializados para profundizar en los fundamentos teóricos y metodológicos sobre calidad del software, su evolución histórica y su estrecha relación con la competitividad organizacional en la industria tecnológica. Además, se analizaron en detalle los principales modelos y estándares internacionales en gestión de calidad, destacando especialmente la integración efectiva de metodologías ágiles y modelos tradicionales como CMMI, IEEE e ISTQB, así como el marco legal y normativo vigente que reguló la calidad y seguridad en el desarrollo tecnológico en Colombia.

Finalmente, esta investigación reconoció que, en un contexto de constante innovación tecnológica, asegurar la calidad del software trascendió la simple corrección de errores para convertirse en una ventaja competitiva diferenciadora. PDT, alineado con sus principios organizacionales de innovación, confianza, excelencia, integridad y responsabilidad, estableció esta estrategia integral de gestión de calidad como un camino imprescindible para posicionarse como líder tecnológico en el mercado colombiano. A continuación, se presenta la definición de la problemática junto con los principales interrogantes que guiarán el desglose del Proyecto.

Definición del Problema

En el ámbito del desarrollo de software, el aseguramiento de la calidad constituyó un componente esencial orientado a salvaguardar la estabilidad, seguridad y eficiencia de las soluciones desplegadas en entornos productivos (Alarcón et al., 2011); no obstante, en diversas organizaciones, los procesos y las metodologías vinculados a la calidad del software se aplicaron de forma incompleta o desestructurada, provocando inconsistencias recurrentes durante las fases de pruebas, deficiencias sustanciales en la documentación técnica y desaprovechamiento de las lecciones aprendidas (Callejas et al., 2017). Esta problemática tuvo como consecuencia un impacto significativo en la calidad de los productos finales entregados al cliente, desencadenando fallos críticos en producción, cuestionamientos severos sobre la idoneidad metodológica adoptada y sobrecostos considerables relacionados con la corrección tardía de defectos; así lo indican estudios, tales como los expuestos por Sommerville (20156), Spillner y Linz (2021).

Asimismo, se evidenció que una falencia recurrente en la industria fue la ausencia de planeación estratégica en la etapa específica de pruebas y certificación de productos informáticos, situación que generó inconsistencias marcadas en la aplicación de metodologías establecidas formalmente, falta notoria de trazabilidad en la ejecución de pruebas, así como insuficiencias sustanciales en la documentación técnica almacenada en los repositorios empresariales (Rasheed et al., 2021). Esta ausencia estructural impidió la detección temprana de errores durante el ciclo de desarrollo, ocasionando que defectos críticos alcanzaran entornos productivos, afectando significativamente la continuidad operativa de los sistemas involucrados (Gupta & Sahu, 2024); además, la escasez de documentación actualizada y fácilmente accesible restringió la reutilización eficiente del conocimiento adquirido en experiencias previas,

provocando la repetición sistemática de fallos similares (Comisión Económica para América Latina y el Caribe (CEPAL), 2021), lo que derivó en un incremento considerable del esfuerzo humano y económico destinado a su corrección (Institute of Electrical and Electronics Engineers (IEEE), 2021).

En particular, en la empresa PDT se identificó que aproximadamente un 75% de los requerimientos técnicos de software carecieron de una adecuada evaluación mediante pruebas funcionales, no funcionales, técnicas y de rendimiento, antes de ser liberados a producción; este déficit sustancial propició numerosas incidencias en los productos entregados, debilitando significativamente la confianza de los usuarios finales y elevando considerablemente los costos de mantenimiento correctivo. Adicionalmente, durante los últimos doce meses, se constató que el 66% de los mantenimientos realizados estuvo asociado a un incremento promedio significativo en los tiempos de ejecución, con la consecuente elevación del gasto operacional; de hecho, se cuantificó un aumento aproximado del 38% en la asignación presupuestal destinada a pruebas y corrección de errores manuales, afectando sustancialmente la eficiencia económica del área de aseguramiento de calidad (International Software Testing Qualifications Board (ISTQB), 2024).

Otro aspecto relevante identificado en este diagnóstico fue la dependencia excesiva respecto a la ejecución manual de pruebas; dicha práctica no solamente contribuyó al incremento exponencial en los costos operacionales, sino que también introdujo riesgos adicionales derivados de errores humanos inherentes al proceso manual de validación (Campos, 2022). La escasa incorporación de herramientas tecnológicas destinadas a la automatización de pruebas y a la implementación de técnicas avanzadas, tales como las pruebas de regresión automatizadas y análisis continuo de rendimiento, resultó determinante para que se introdujeran defectos inadvertidos en el ambiente productivo (Serna et al., 2019); la automatización representa una

práctica fundamental en términos de agilidad, precisión y reducción de errores humanos, según lo indicado por Fewster y Graham (2019) y Abad y Ko (2023), quienes recomiendan su aplicación rigurosa dentro de las prácticas estándar del ciclo de desarrollo.

Conviene precisar que la problemática en PDT no se derivó de una falta absoluta de modelos de referencia o estándares metodológicos internacionalmente reconocidos, sino que tuvo origen en la deficiente ejecución de estrategias concretas y efectivas para implementar satisfactoriamente las mejores prácticas disponibles en la industria; aunque existen múltiples estándares robustos en la literatura especializada (Meso et al., 2021), tales como las guías sugeridas en la gestión integral del proyecto por el PMI (2021), y los lineamientos técnicos presentados por Sommerville (2016), sin la existencia de un marco metodológico claramente definido y adaptado a la realidad específica organizacional, la calidad del software seguirá siendo una vulnerabilidad persistente y costosa en múltiples organizaciones contemporáneas (Ybañez et al., 2023).

Ante este panorama, se tornó indispensable diseñar e implementar estrategias integrales de optimización orientadas específicamente al fortalecimiento y mejora continua en las áreas de pruebas y certificación de software; este diseño de estrategias debía fundamentarse sólidamente en herramientas avanzadas de gestión del conocimiento organizacional, metodologías robustas de mejora continua y técnicas innovadoras de automatización de procesos. Al adoptar estas iniciativas de mejora sustancial, se pretendía minimizar la probabilidad de incidencias críticas en producción, optimizar sustancialmente la eficiencia operativa del aseguramiento de calidad, disminuir considerablemente el costo asociado al reproceso (Pérez, 2011), y sobre todo garantizar una experiencia más confiable para los usuarios finales del producto, cumpliendo así estándares internacionales de calidad reconocidos en el ámbito tecnológico (IEEE, 2021).

Finalmente, la problemática diagnosticada en la empresa PDT puso en evidencia una realidad crítica ampliamente extendida en el sector del desarrollo de software, caracterizada por deficiencias reiteradas en la implementación efectiva de estrategias y procedimientos rigurosos de aseguramiento de la calidad; esta situación generó efectos negativos evidentes tanto en la continuidad operativa como en la sostenibilidad financiera de los proyectos tecnológicos. En consecuencia, la elaboración e implementación estructurada de un plan estratégico que contemple la adopción efectiva de estándares internacionales, como la aplicación integral del modelo ISTQB y de prácticas ágiles sugeridas por el PMI, permitirá alcanzar niveles superiores de calidad, confiabilidad y competitividad; de esta forma, se reducirá la incidencia de fallos críticos, mejorando sustancialmente la estabilidad operativa y aumentando la satisfacción y confianza del usuario final ante los productos entregados por la organización.

Preguntas Problema

A partir de la problemática identificada en los procesos de calidad del ciclo de vida del desarrollo de software en el área de pruebas y certificación de la empresa PDT, se plantea la necesidad de mejorar el plan estratégico mediante la implementación de herramientas establecidas en la Guía del PMBOK® (2021). Para ello, se define la siguiente pregunta problema general, que orienta el desarrollo de la investigación:

¿Cómo optimizar el plan estratégico de los procesos de calidad en el ciclo de vida del desarrollo de software en el área de pruebas y certificación de la empresa PDT, mediante la aplicación de herramientas y mejores prácticas establecidas en la Guía del PMBOK®?

En función de este cuestionamiento central, se plantean las siguientes preguntas secundarias, que permiten abordar de manera estructurada los aspectos clave del estudio:

¿Cuáles son las principales deficiencias operacionales y malas prácticas en el área de testing de software que afectan la calidad del producto final en la empresa PDT? ¿Qué herramientas y metodologías de análisis permiten evaluar las causas raíz de las deficiencias en los procesos de calidad dentro del ciclo de vida del desarrollo de software? ¿Cómo diseñar estrategias efectivas para optimizar los procesos de pruebas y certificación del software, alineadas con la Guía del PMBOK® y las mejores prácticas del sector?

Justificación

La optimización de los procesos de calidad en el desarrollo de software representó un aspecto crucial para asegurar la eficacia operativa y la confiabilidad funcional de los productos implementados en ambientes productivos; en ese contexto, la aplicación rigurosa de metodologías especializadas para el aseguramiento de calidad se consolidó como una práctica indispensable para minimizar la incidencia de errores en las soluciones informáticas, incrementar significativamente la satisfacción del usuario final y reducir sustancialmente los costos derivados de la corrección tardía de fallas críticas en etapas avanzadas del ciclo de vida del desarrollo. Al respecto, Sommerville (2016) señaló que las organizaciones capaces de implementar estrategias efectivas de gestión de calidad lograron reducir hasta en un 30% los defectos generados en producción, mientras que Jaakkola y Thalheim (2005) destacaron cómo la automatización adecuada de pruebas impulsó la eficiencia operacional en más del 40%; dichos hallazgos justificaron la necesidad imperiosa de diseñar e implementar un plan estratégico capaz de optimizar de forma integral los procesos de pruebas y certificación, garantizando así estabilidad y alto rendimiento en los sistemas tecnológicos desplegados.

En particular, dentro del entorno operativo de la empresa PDT, la implementación de un plan estructurado de mejora en el área específica de pruebas y certificación respondió directamente a la urgencia de mitigar la alta dependencia de prácticas manuales y fortalecer la trazabilidad efectiva de los defectos detectados durante la validación del software; asimismo, la descentralización organizada del conocimiento especializado y el uso sistemático de herramientas basadas en la Guía del PMBOK® (Project Management Institute [PMI], 2021) facilitaron notablemente la estandarización de procedimientos internos y la optimización racional del aprovechamiento de recursos técnicos y humanos. Benavides (2025) indicó que mejorar los

procesos de calidad mediante la incorporación estructurada de metodologías robustas de gestión de proyectos permitió a diversas empresas disminuir hasta en un 20% los costos operacionales, al tiempo que aceleraron en aproximadamente un 25% los tiempos promedio de respuesta en la detección temprana y corrección efectiva de errores en entornos productivos; en consecuencia, la adopción estratégica de estos enfoques metodológicos en PDT se tradujo en una significativa mejora organizacional en términos de eficiencia, calidad operativa y optimización presupuestal.

Por otro lado, la dimensión económica constituyó otro factor determinante en la justificación del estudio realizado; según lo expuesto por Villa (2024), los sobrecostos derivados de defectos no identificados oportunamente en etapas tempranas del desarrollo llegaron a representar hasta un 45% del presupuesto total asignado al proyecto, impactando negativamente la rentabilidad global de las organizaciones del sector tecnológico. En línea con esta evidencia, la incorporación formalizada de metodologías más rigurosas y automatizadas permitió reducir en un 35% el tiempo requerido para la ejecución de pruebas, así como disminuir en un 50% los incidentes registrados en producción (Álvarez, 2025); tales resultados fortalecieron sustancialmente la necesidad de estructurar e implementar estrategias concretas y efectivas orientadas hacia la optimización integral del proceso de aseguramiento de calidad en PDT, facilitando así una gestión eficiente, estratégica y responsable de los recursos disponibles en la organización.

Asimismo, más allá del ámbito estrictamente financiero, la optimización de los procesos de calidad impactó positivamente en la satisfacción del usuario final y en la reputación organizacional; se comprobó que la ausencia o deficiencia en el control riguroso durante los procesos de certificación facilitó la aparición de defectos críticos en producción, lo cual perjudicó considerablemente la percepción de confiabilidad por parte de los clientes, generando

además incrementos sustanciales en la carga operativa destinada a la corrección reactiva de errores detectados. De acuerdo con Irurita y Villanueva (2012), la implementación efectiva de estrategias de calidad redujo hasta en un 60% la probabilidad de defectos en producción y fortaleció significativamente la percepción externa de solidez operativa de los sistemas tecnológicos; en consecuencia, la adopción en PDT de un marco estratégico robusto, fundamentado en herramientas avanzadas de gestión, aseguró la estabilidad y rendimiento de sus productos, minimizando de esta forma los riesgos operacionales inherentes a la falta de rigurosidad en los procesos técnicos y metodológicos adoptados.

Desde la perspectiva del fortalecimiento del capital humano, la implementación de programas continuos de formación y capacitación especializada en el área específica de pruebas y certificación emergió como una estrategia esencial para mitigar la rotación de personal y garantizar la preservación efectiva del conocimiento organizacional adquirido a través del tiempo; en este sentido, un estudio realizado por PMI (2021) señaló que aquellas empresas que invirtieron activamente en formación continua y certificaciones técnicas específicas lograron reducir la rotación del personal en un 28%, incrementando paralelamente la retención y consolidación del conocimiento en aproximadamente un 35%. Dentro del contexto organizacional específico de PDT, fortalecer sistemáticamente las competencias técnicas del equipo dedicado a pruebas permitió generar un entorno laboral más eficiente, estructurado y motivador, asegurando así la continuidad operativa y facilitando mejoras sustanciales en los procesos internos vinculados al aseguramiento riguroso de calidad.

Adicionalmente, la ausencia histórica de documentación técnica estructurada y la inexistencia de un repositorio accesible de lecciones aprendidas afectaron considerablemente la continuidad y eficacia de los procesos de calidad en PDT; según Álvarez (2025), alrededor del

80% de los proyectos de software que carecieron de documentación adecuada presentaron serias dificultades en la detección temprana de errores, generando pérdidas económicas sustanciales asociadas a retrasos operacionales y reprocesos técnicos recurrentes. La implementación formalizada de herramientas tecnológicas avanzadas para la gestión efectiva del conocimiento permitió en consecuencia optimizar significativamente la trazabilidad de los procesos internos, facilitando la toma de decisiones estratégicas basadas en datos concretos, reduciendo sustancialmente la repetición de errores, y fortaleciendo así una cultura organizacional orientada hacia la mejora continua.

Finalmente, en términos del impacto sectorial, la optimización efectiva de los procesos de pruebas y certificación representó no solo beneficios internos directos para PDT, sino que además sentó un precedente metodológico relevante aplicable por otras organizaciones que enfrentaron desafíos similares en la región. Según el informe emitido por ISTQB (2023), más del 60% de las empresas tecnológicas en Latinoamérica reportaron deficiencias significativas en sus procesos de calidad, abriendo oportunidades valiosas para implementar modelos robustos de mejora alineados con estándares internacionales reconocidos; la implementación en PDT de un marco estratégico sólido basado en la Guía del PMBOK® permitió posicionar a la empresa como referencia metodológica destacada en el ámbito regional del aseguramiento de calidad del software, fortaleciendo notablemente su competitividad en el mercado tecnológico global.

Objetivos

Objetivo General

Optimizar el plan estratégico de los procesos de calidad en el ciclo de vida del desarrollo de software en el área de pruebas y certificación para la empresa PDT, mediante la implementación de herramientas y mejores prácticas establecidas en la Guía del PMBOK® (2021), con el fin de fortalecer la eficiencia, confiabilidad y cumplimiento de estándares en la validación y certificación del software.

Objetivos Específicos

Diagnosticar las deficiencias operacionales del área de pruebas de software, identificando las malas prácticas que afectan la calidad del desarrollo y reconociendo aquellas prácticas existentes que pueden ser fortalecidas o escaladas, a través del análisis de encuestas, registros históricos, lecciones aprendidas y la metodología DOFA, con el propósito de establecer un marco de referencia para su optimización.

Diagnosticar las prácticas operacionales del área de pruebas de software, identificando tanto aquellas que afectan negativamente la calidad del desarrollo como las que aportan valor al proceso, a través del análisis de encuestas, registros históricos, lecciones aprendidas y la metodología DOFA, con el propósito de establecer un marco de referencia integral para su optimización.

Analizar los procesos de calidad dentro del ciclo de vida del desarrollo de software, empleando herramientas como el análisis de causa raíz, análisis de procesos, diagrama de Ishikawa (causa y efecto), diagrama de Pareto y diagrama de afinidad, para determinar las oportunidades de mejora en la gestión del área de testing.

Diseñar estrategias de optimización para los procesos de calidad en el área de pruebas y certificación, fundamentadas en herramientas de dirección de proyectos y metodologías de mejora continua, asegurando su viabilidad y alineación con los estándares de la Guía del PMBOK® y las mejores prácticas del sector.

Socializar los resultados con los involucrados, agradecimiento de los beneficios del proyecto de la empresa para la UNAD y el programa de la maestría MGP.

Marco Teórico

El Marco Teórico constituye un elemento fundamental e indispensable en la estructuración de cualquier investigación académica, dado que proporciona el sustento conceptual y metodológico sobre el cual se edifica el análisis riguroso del problema abordado (Trigo, 2021); particularmente, en el contexto del aseguramiento de la calidad dentro del ciclo de vida del desarrollo de software, dicho marco permitió definir con precisión los principios esenciales, metodologías vigentes y enfoques estratégicos aplicables para optimizar los procesos asociados al área de pruebas y certificación. Asimismo, la adecuada formulación de este Marco Teórico facilitó sustancialmente la comprensión de los modelos existentes, propiciando la identificación clara y sistemática de brechas en la gestión de calidad, así como la validación efectiva de herramientas destinadas a mejorar significativamente la eficiencia operativa en el área específica de testing dentro de la empresa PDT.

La optimización efectiva del plan estratégico para pruebas y certificación de software exigió un análisis minucioso y estructurado de los diversos modelos de calidad, así como de las metodologías especializadas en gestión de proyectos; esto con el propósito fundamental de garantizar una reducción considerable de defectos en producción, optimizar estratégicamente los costos operativos implicados en el desarrollo de soluciones tecnológicas, y mejorar sustancialmente la confiabilidad de los sistemas implementados. En este orden de ideas, la Guía del PMBOK® (2021) proporcionó un conjunto robusto de herramientas y prácticas idóneas para estructurar estrategias efectivas en materia de aseguramiento de calidad; por otra parte, las metodologías y estándares internacionales definidos por el International Software Testing Qualifications Board (ISTQB, 2023) desempeñaron un rol determinante en la estandarización adecuada de los procesos técnicos asociados a pruebas, permitiendo garantizar la trazabilidad

completa de defectos y el cumplimiento riguroso de requisitos establecidos durante el desarrollo y validación de software (PMI, 2021; ISTQB, 2023).

El presente Marco Teórico se articuló específicamente alrededor de cuatro ejes centrales claramente definidos: las bases conceptuales, los antecedentes históricos, las metodologías aplicadas, y finalmente, las normativas vigentes. Respecto a las bases conceptuales, éstas facilitaron la definición exhaustiva y precisa de la calidad del software, la identificación de sus múltiples dimensiones operativas y su estrecha relación con los principios fundamentales de la gestión de proyectos tecnológicos. En lo concerniente a los antecedentes históricos, éstos contextualizaron de manera crítica la evolución experimentada por los diversos modelos de calidad en la industria, permitiendo identificar tendencias destacadas en la gestión especializada de pruebas y certificación; además, en cuanto a las metodologías aplicadas, se analizaron en profundidad modelos ampliamente reconocidos como CMMI, Six Sigma, Agile y DevOps, con el fin de evaluar su aplicabilidad específica y práctica en la optimización integral de los procesos de testing.

Fundamentos de la Calidad en el Desarrollo de Software

Concepto de Calidad en Software

El concepto de calidad ha sido históricamente objeto de múltiples definiciones, evolucionando constantemente conforme aumentaron las exigencias del mercado globalizado y los estándares internacionales vigentes; particularmente, en el ámbito del desarrollo de software, la calidad representó un factor crítico determinante que influyó significativamente sobre la confiabilidad, eficiencia operativa y sostenibilidad funcional de los sistemas implementados. Al respecto, Sommerville (2016) definió la calidad del software como “el grado en que un sistema, componente o proceso cumplió efectivamente con los requisitos especificados y las expectativas

manifiestas de los usuarios” (p. 154); esta concepción enfatizó claramente la necesidad de evaluar la calidad desde dos perspectivas fundamentales, tanto objetiva como subjetiva, al considerar la conformidad con especificaciones técnicas predefinidas y, simultáneamente, la satisfacción experimentada por el usuario final durante su interacción directa con el producto desarrollado.

Complementariamente, Pressman (2019) señaló que “la calidad en un producto de software debió medirse a través de atributos fundamentales tales como funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad” (p. 67); estableciendo, en consecuencia, que la calidad no constituyó un estado estático, sino un proceso dinámico y continuo que debió asegurarse durante todas las fases del ciclo de vida del desarrollo.

Desde una perspectiva centrada en la gestión estratégica de proyectos, la Guía del PMBOK® (Project Management Institute [PMI], 2021) aportó una definición integral sobre calidad en proyectos tecnológicos, indicando explícitamente que “la calidad implicó satisfacer efectivamente las expectativas de los interesados, así como cumplir de manera rigurosa con los requisitos establecidos para el proyecto y el producto entregable” (p. 47); desde esta visión amplia, la calidad no quedó limitada únicamente a la ausencia de defectos técnicos en el software, también contempló su capacidad adaptativa frente a las necesidades cambiantes del cliente y su operatividad sostenible en el tiempo.

Asimismo, el PMI (2021) destacó que los equipos responsables del proyecto utilizaron métricas específicas y criterios claros de aceptación basados en requisitos definidos, que pudieron provenir directamente de los interesados, contratos establecidos, políticas organizacionales vigentes o estándares regulatorios aplicables; esta observación subrayó la importancia crucial de una planificación estructurada en el aseguramiento efectivo de calidad,

garantizando que los entregables respondieran adecuadamente a las exigencias del entorno empresarial.

El concepto de calidad en software experimentó una notable transformación durante las últimas décadas, proceso que se produjo en función de los avances tecnológicos constantes, las exigencias del mercado globalizado, así como la incorporación progresiva de estándares internacionales de gestión especializados (López et al., 2008); este desarrollo continuo llevó a cuestionar cómo ha variado la percepción sobre la calidad en el desarrollo del software desde sus inicios, y cuáles han sido los aspectos cruciales que han incidido en la aparición de nuevos enfoques para su gestión efectiva (Sunkel & Trucco, 2012).

Del mismo modo, emergieron interrogantes relevantes relacionados con los enfoques metodológicos que permitieron garantizar, de manera sistemática y sostenible, tanto la confiabilidad técnica como la eficiencia funcional de los sistemas implementados; estas inquietudes se abordaron mediante el análisis detallado de definiciones y metodologías propuestas por diferentes referentes académicos, cuyos aportes resultaron determinantes para comprender la evolución conceptual y práctica de la calidad del software en distintos contextos organizacionales.

Tabla 1*Evolución del Concepto de Calidad en el Desarrollo de Software*

Autor	Definición de Calidad
Ishikawa (1986)	La calidad no quedó limitada exclusivamente al producto final entregado, sino que abarcó también la calidad del trabajo realizado, los servicios brindados, los procesos internos, la información generada y la estructura organizacional en su conjunto.
Crosby (1988)	La calidad es el cumplimiento riguroso de requisitos previamente establecidos, los cuales debieron ser medibles y verificables para asegurar objetivamente la conformidad.
Juran (1990)	Existieron dos enfoques principales: el primero relacionado con la capacidad del producto para cumplir eficientemente con las expectativas del cliente, y el segundo orientado hacia la reducción sistemática de deficiencias y errores durante el proceso productivo.
Senlle (2004)	La estabilidad operativa, robustez técnica y facilidad efectiva para actualizar un software constituyeron factores decisivos en la determinación de su nivel global de calidad, permitiendo optimizar sustancialmente su ciclo de vida y reducir considerablemente los costos asociados al mantenimiento reactivo.
Cuatrecasas (2012)	La calidad de un sistema informático no solo debió medirse por su capacidad inicial para satisfacer los requisitos planteados por los clientes, sino además por su facilidad intrínseca de mantenimiento continuo y evolución sostenida.
Guía del PMBOK® (2021)	La metodología del costo de la calidad (COQ) fue utilizada para encontrar el equilibrio óptimo entre la inversión en actividades preventivas y de evaluación de calidad, buscando reducir al máximo la aparición de defectos o fallas en los productos entregados.
IEEE (2022)	Un software de calidad cumpliera adecuadamente criterios exigentes de fiabilidad, mantenibilidad, eficiencia operativa y seguridad técnica, los cuales debieron ser evaluados minuciosamente mediante pruebas automatizadas y auditorías rigurosas de código fuente.

Autor	Definición de Calidad
ISTQB (2023)	Los estándares para asegurar calidad en software debieron centrarse prioritariamente en la prevención efectiva de defectos, en la implementación rigurosa de pruebas sistemáticas y en la certificación formalizada de los procesos de desarrollo utilizados.

Nota. La Tabla resume diversas conceptualizaciones sobre el significado de calidad, formuladas por autores y organismos reconocidos a lo largo del tiempo.

Inicialmente, Ishikawa (1986) propuso que la calidad no debía limitarse únicamente al producto final entregado, sino que también debía abarcar integralmente los procesos internos, los servicios relacionados, la calidad de la información generada y, de manera complementaria, la estructura organizacional misma; esta visión integral sentó bases sólidas para la adopción sistemática del aseguramiento de calidad, tanto en industrias manufactureras como en sectores orientados a servicios especializados. Más adelante, autores influyentes como Juran (1990) y Crosby (1988) ampliaron sustancialmente esta perspectiva inicial, enfatizando particularmente la relevancia estratégica de satisfacer adecuadamente las expectativas del cliente y reducir al máximo posible la incidencia de errores en los procesos productivos involucrados; entonces surgió otra pregunta clave: ¿por qué se hizo relevante medir la calidad de manera objetiva? Crosby introdujo así el principio esencial de “conformidad con requisitos”, estableciendo explícitamente que la calidad debía ser un atributo claramente medible, verificable y documentado para asegurar objetivamente la conformidad esperada.

Con el crecimiento exponencial del software como industria predominante, emergió claramente la necesidad específica de evaluar la calidad en términos de su capacidad para el mantenimiento y evolución sostenida; en esta línea, autores como Cuatrecasas (2012) y Senlle (2004) subrayaron que un software de calidad no solo debía cumplir inicialmente con los

requisitos establecidos, sino que también debía exhibir robustez técnica, capacidad adaptativa efectiva frente a cambios futuros y minimizar sustancialmente los costos derivados de actividades continuas de mantenimiento preventivo y correctivo durante su ciclo de vida operativo. En años más recientes, los enfoques estratégicos hacia la calidad en software evolucionaron decididamente hacia modelos basados en la prevención sistemática de defectos y la optimización integral del denominado costo de la calidad; específicamente, la Guía del PMBOK® (2021) destacó que mantener un equilibrio adecuado entre actividades preventivas y correctivas fue clave para lograr una eficiencia operativa sostenible, mientras que los estándares internacionales establecidos por ISTQB (2023) e IEEE (2022) consolidaron formalmente prácticas rigurosas de auditoría de código y pruebas automatizadas para garantizar plenamente la fiabilidad y desempeño de los productos tecnológicos antes de su despliegue final en producción.

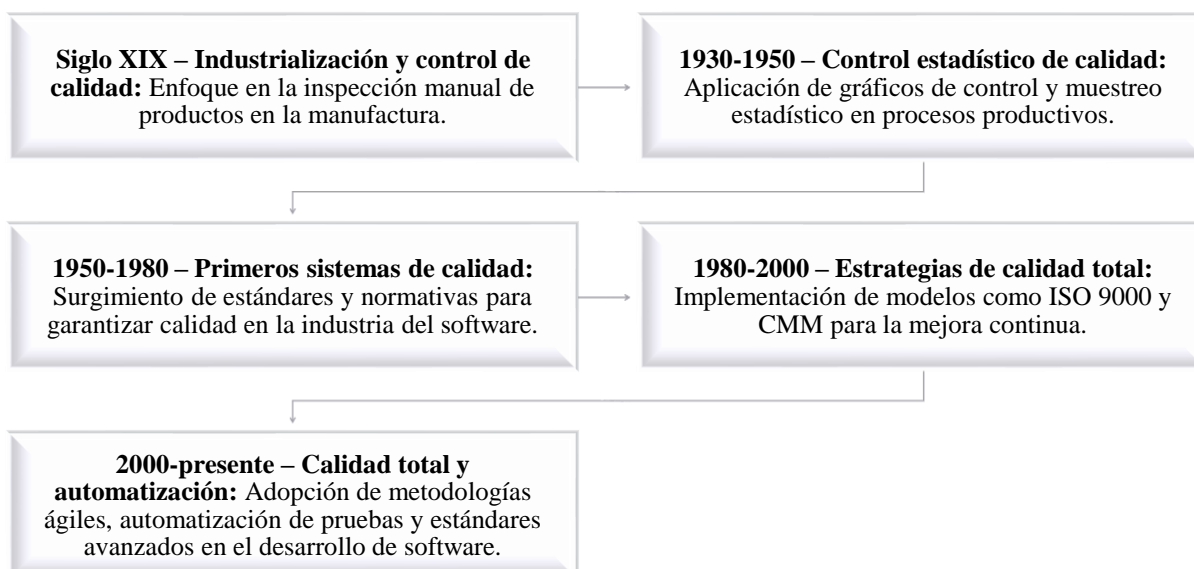
Evolución Histórica de la Calidad en Software

Según lo indicado por Irurita & Villanueva (2012), durante el siglo XIX, la calidad se fundamentó principalmente en la inspección sistemática de productos finales, buscando cumplir rigurosamente con estándares predefinidos de manufactura industrial. Posteriormente, en el transcurso de los años treinta y cuarenta del siglo XX, emergieron enfoques centrados específicamente en el control estadístico de calidad, destacando el uso riguroso de gráficos de control y técnicas estadísticas para optimizar procesos productivos, los cuales evolucionaron significativamente hasta consolidarse en los primeros sistemas formales de aseguramiento de calidad entre 1950 y 1980; estos sistemas establecieron estándares claros y metodologías estructuradas que posteriormente serían adoptadas por diversas industrias, incluida la naciente industria tecnológica del software (Irurita & Villanueva, 2012).

A partir de la década de 1980, se consolidaron estrategias avanzadas enfocadas en calidad total, enfatizando especialmente la prevención efectiva de defectos y colocando la satisfacción del cliente como objetivo primordial dentro de cualquier estrategia de desarrollo tecnológico; en esta etapa se desarrollaron estándares internacionales robustos, tales como IEEE, ISO y CMMI, orientados específicamente hacia la mejora continua y la reducción sistemática de errores en todas las fases del desarrollo de software. En los años recientes, estas estrategias evolucionaron hacia metodologías más sofisticadas y automatizadas, integrando conceptos como las metodologías ágiles, DevOps, la generación automática de casos de prueba y auditorías técnicas avanzadas, las cuales garantizaron mayor fiabilidad, eficiencia operativa y optimización en los costos del aseguramiento de calidad en productos de software (Irurita-Alzuetza & Villanueva-Roldán, 2012)

Figura 1

Evolución de la Calidad en Software



Nota. Adaptado de. Sistemas de gestión de la calidad, Irurita, J. & Villanueva, P. (2012)

<https://core.ac.uk/download/pdf/10851013.pdf>

Relación entre Calidad de Software y Competitividad en la Industria Tecnológica

La calidad del software constituyó un elemento decisivo para alcanzar altos niveles de competitividad en el sector tecnológico, dado que impactó directamente en la satisfacción del cliente, la eficiencia operativa y la rentabilidad general de las organizaciones (Nabot & Al-Qerem, 2024); en este sentido, las empresas que adoptaron rigurosamente estándares internacionales como ISO, IEEE y CMMI lograron reducir sustancialmente defectos en producción, lo cual generó importantes ahorros en costos operacionales y fortaleció su ventaja competitiva en el mercado (Đorđević, 2017). Por otro lado, Vaníček (2006) destacó que cumplir con requisitos estrictos de calidad en software aumentó significativamente la confianza percibida por los clientes, facilitando así la diferenciación estratégica en contextos altamente competitivos. Adicionalmente, Momaya y Ambastha (2005) señalaron que las empresas tecnológicas que invirtieron activamente en gestión avanzada de la calidad optimizaron su eficiencia interna y mejoraron sustancialmente su desempeño en el largo plazo.

Por otra parte, Madhavaram et al. (2023) destacaron que la falta de una gestión eficiente de calidad en empresas tecnológicas provocó graves afectaciones económicas y daños reputacionales, especialmente en sectores sensibles como salud, finanzas y telecomunicaciones; en complemento, Syahchari et al. (2020) afirmaron que el uso estratégico de tecnologías avanzadas en la gestión de calidad del software fortaleció significativamente la posición competitiva de las empresas en mercados exigentes. Desde una perspectiva similar, Chaudhry y Chaudhry (2023) confirmaron que las organizaciones capaces de innovar constantemente en sus procesos internos de aseguramiento de calidad generaron ventajas competitivas sostenibles mediante la optimización operativa. Finalmente, Madhavaram et al. (2023) y St. Quintin (1993) concluyeron que el capital intelectual, las relaciones intraorganizacionales y el enfoque

preventivo en calidad fueron factores determinantes para consolidar la competitividad tecnológica, garantizando así productos más robustos, eficientes y confiables.

Principales Modelos y Estándares para la Gestión de Calidad en Software

La industria del software desarrolló diversas metodologías y estándares internacionales para asegurar la calidad de sus productos, buscando evaluar de forma sistemática el cumplimiento de requisitos técnicos y funcionales exigidos por los clientes (Đorđević, 2017); entre estos enfoques destacó particularmente el modelo Capability Maturity Model Integration (CMMI), que estableció una metodología estructurada para evaluar la madurez organizacional y mejorar continuamente los procesos internos relacionados con la calidad (Jones, 1997).

Paralelamente, Martínez et al. (2022) resaltaron la importancia de adaptar estos modelos al contexto de sistemas basados en inteligencia artificial, estableciendo lineamientos específicos para asegurar su calidad y desempeño funcional. Asimismo, Castillo et al. (2010) desarrollaron el modelo REASQ, orientado a identificar aspectos críticos y requisitos clave para evaluar integralmente la calidad del software, enfatizando así la necesidad de una evaluación objetiva y metodológicamente robusta durante el desarrollo.

Adicionalmente, estándares internacionales como IEEE definieron claramente la necesidad de implementar auditorías técnicas y pruebas automatizadas para garantizar la fiabilidad, seguridad y eficiencia operativa del software entregado al mercado (Eisty & Carver, 2022); en consonancia con esta perspectiva, Mustafa et al. (2021) destacaron la importancia de la generación automatizada de casos de prueba desde los requisitos iniciales para detectar tempranamente posibles defectos en el producto. Liu et al. (2022) propusieron, mediante el modelo iTrustEval, una metodología innovadora basada en procesos analíticos jerárquicos

(AHP) para evaluar integralmente la confiabilidad técnica del software desarrollado.

Adicionalmente, AlOmar et al. (2023) demostraron cómo la aplicación del “linting” durante la educación en ingeniería de software permitió identificar tempranamente patrones negativos en el código, fortaleciendo la calidad desde las etapas iniciales; finalmente, Tyagi et al. (2019) resaltaron la importancia de los modelos de confiabilidad enfocados especialmente en seguridad para software de código abierto, mostrando la necesidad de asegurar la calidad de manera holística y adaptativa según las características del entorno tecnológico específico (Đorđević, 2017).

Figura 2

Relación entre Calidad del Software y Competitividad en la Industria Tecnológica



Nota. La Figura integra los principales componentes del concepto de calidad del software frente al impacto en la competitividad, así como sus dimensiones y factores estratégicos.

Pruebas y Certificación de Calidad en Software

No existe software de alta calidad sin un proceso riguroso de pruebas y una certificación técnica bien estructurada. En la actualidad, garantizar la confiabilidad, eficiencia y seguridad de los sistemas informáticos no es una opción, sino una necesidad estratégica. Las pruebas han dejado de ser actividades correctivas al final del ciclo de desarrollo para convertirse en procesos continuos, automatizados y predictivos, capaces de detectar errores críticos desde las fases más tempranas. Por su parte, la certificación aporta evidencia objetiva de que un producto tecnológico cumple con los estándares funcionales, técnicos y regulatorios exigidos por la industria. Esta dupla —pruebas y certificación— forma el núcleo del aseguramiento de calidad moderno, optimizando tiempos de entrega, reduciendo costos de mantenimiento y consolidando la confianza del cliente. Bajo marcos metodológicos como ISTQB, IEEE y el enfoque propuesto en la Guía del PMBOK® (Project Management Institute, 2021), las organizaciones pueden estructurar procesos más ágiles, robustos y alineados con las exigencias dinámicas del mercado digital.

Siguiendo este contexto, la evolución acelerada de los sistemas interconectados, los despliegues continuos y las arquitecturas distribuidas incrementó de forma exponencial la complejidad del entorno tecnológico; en consecuencia, la gestión técnica del aseguramiento de calidad del software se posicionó como una función crítica para preservar la integridad operacional, garantizar la resiliencia sistémica y consolidar la confianza en los productos entregados al mercado. Bajo este nuevo enfoque especializado, la calidad dejó de interpretarse como un proceso reactivo centrado en la detección puntual de fallos; en su lugar, se adoptó un modelo proactivo que integró técnicas predictivas, análisis estáticos y dinámicos del código

fuentes, automatización sistemática de procesos y ciclos iterativos de validación orientados a la prevención temprana de defectos críticos; en este contexto, el reto principal para las organizaciones consistió en evaluar las implicaciones operativas derivadas de la ausencia de metodologías estructuradas en pruebas y certificación, así como en cuantificar el impacto técnico y económico asociado a la falta de estrategias efectivas para mitigar los riesgos inherentes al desarrollo acelerado del software (Sader et al., 2021; Sassi et al., 2023).

La incorporación estratégica de modelos técnicos robustos permitió optimizar el rendimiento de las soluciones tecnológicas, reducir de manera significativa los costos asociados al mantenimiento correctivo y fortalecer la percepción de confiabilidad de los usuarios finales; aquellas organizaciones que adoptaron estándares internacionales rigurosos, como los definidos por ISTQB e IEEE, complementados con marcos ágiles como DevOps, Scrum o Kanban, lograron implementar soluciones informáticas con mayor robustez técnica, escalabilidad operativa y alineación precisa frente a los requisitos exigidos por entornos globales altamente competitivos.

Un ejemplo destacado de esta integración metodológica fue la adopción de pipelines automatizados para pruebas de regresión continua, soportados en frameworks especializados, los cuales permitieron validar el código fuente de manera permanente antes de su despliegue final en ambientes productivos; bajo este enfoque end-to-end, el aseguramiento de calidad trascendió la validación puntual, consolidándose como un proceso transversal, iterativo e integral que garantizó la entrega de software confiable, seguro y sostenible desde su concepción hasta su puesta en operación (Sebestyén et al., 2023; Urbano & Kamran, 2023; Okpala & Korzeniowska, 2021); es por esto que, resulta indispensable detallar las principales metodologías especializadas que soportan el proceso de pruebas y certificación del software.

Metodologías Especializadas en Pruebas y Certificación de Software

El aseguramiento de calidad requirió la implementación articulada de múltiples niveles de validación técnica, ejecutados mediante procesos sistemáticos que garantizaron la integridad funcional, estructural y operativa del producto final ¿Cuáles fueron estas metodologías y cómo contribuyeron a la mejora sustancial del producto?

Inicialmente, se llevaron a cabo pruebas unitarias para verificar exhaustivamente la cohesión y precisión algorítmica de los componentes individuales, detectando anomalías a nivel de módulos discretos desde las fases tempranas del ciclo de vida (Homès, 2024). Posteriormente, se desarrollaron pruebas de integración orientadas a la validación funcional de las interfaces internas entre módulos, asegurando la interoperabilidad técnica y mitigando tempranamente potenciales defectos por dependencias estructurales y comportamientos inesperados en las interfaces (Homès, 2024; Piattini et al., 2021).

Seguidamente, la estrategia contempló pruebas avanzadas de regresión y estrés, orientadas específicamente a mantener la estabilidad funcional y el desempeño óptimo bajo escenarios críticos, tales como actualizaciones del código base o condiciones extremas de carga transaccional y uso concurrente (Lim et al., 2022). Finalmente, se ejecutaron pruebas de aceptación que garantizaron, mediante criterios estrictamente definidos por stakeholders técnicos y usuarios expertos, la conformidad explícita del producto con los requisitos funcionales, técnicos y normativos establecidos para la entrega final (Mustafa et al., 2021; Lim et al., 2022).

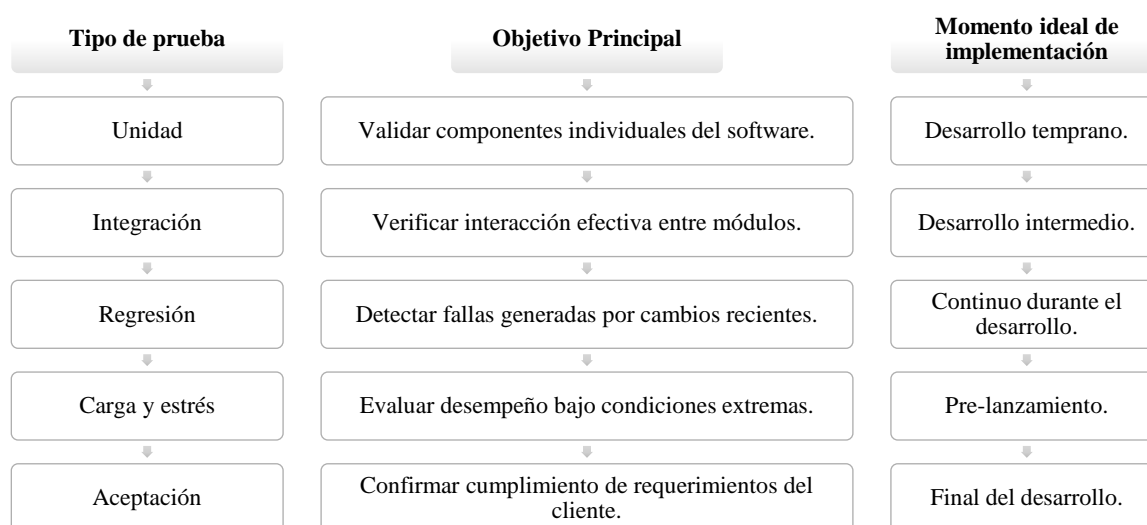
Respecto a la certificación técnica del software, su ejecución exigió adherirse rigurosamente a marcos metodológicos internacionalmente reconocidos, implementados mediante auditorías especializadas, análisis estático y dinámico del código fuente, así como

generación automatizada de casos de prueba basados estrictamente en la trazabilidad detallada de los requisitos iniciales del sistema (Xue & Zou, 2023; Antony et al., 2021).

Como consecuencia, en sectores estratégicos y críticos—particularmente aquellos con altos estándares regulatorios y de seguridad operativa—la certificación proporcionó evidencia técnica verificable sobre la idoneidad del producto para ambientes altamente exigentes, facilitando así su despliegue e integración eficaz en ecosistemas empresariales complejos; por ende, esta práctica especializada no únicamente elevó sustancialmente el nivel de calidad técnica del software producido, sino que además consolidó significativamente la credibilidad técnica y reputacional de la organización en el ámbito tecnológico (Ababneh, 2020; Gatto et al., 2023).

Figura 3

Metodologías de Pruebas de Software



Nota. La figura presenta los tipos de prueba más relevantes en el aseguramiento de calidad del software, su propósito principal y el momento óptimo para su ejecución dentro del ciclo de desarrollo. Adaptado de. *Fundamentals of software testing*, Homès, B. (2024)

<https://download.e-bookshelf.de/download/0000/7533/44/L-G-0000753344-0002285971.pdf>

De acuerdo con Berényi (2023), la inclusión de estrategias de calidad desde las primeras etapas del ciclo de vida del software permite: A) Identificar riesgos y fallos desde el inicio del proyecto; B) Optimizar la asignación de recursos en pruebas y validaciones; C) Garantizar la entrega de un producto confiable y libre de defectos. La Guía del PMBOK® (2021) enfatiza que la calidad en proyectos tecnológicos depende tanto del cumplimiento de requisitos funcionales, como de la capacidad de gestionar riesgos asociados a defectos de software en todas sus etapas.

Marcos de trabajo Ágiles en la Gestión de Calidad de Software

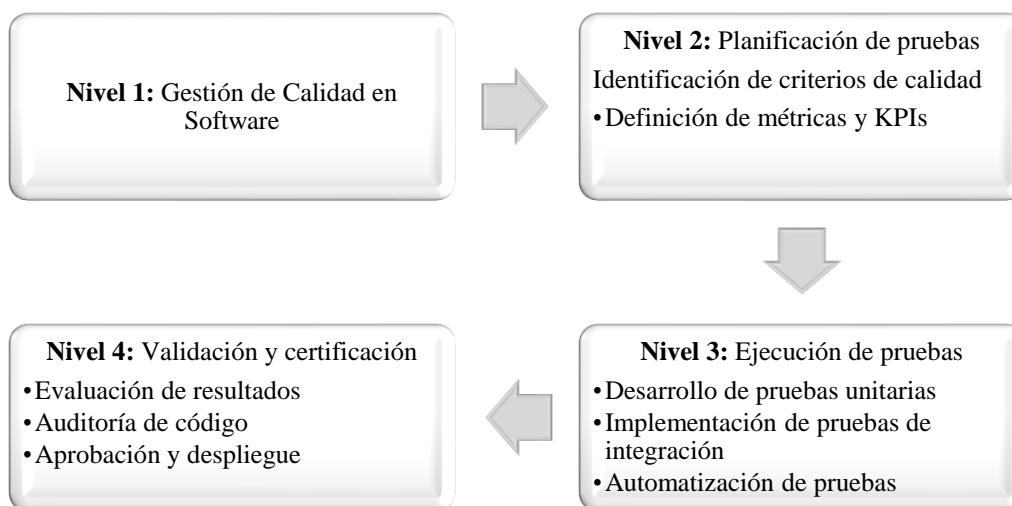
La irrupción de los marcos de trabajo ágiles transformó radicalmente el paradigma tradicional de la gestión de calidad en el ciclo de vida del software, desplazando modelos secuenciales como el enfoque en cascada, y promoviendo esquemas iterativos, adaptativos e incrementales (Piattini et al., 2021); en este contexto, la integración entre metodologías como Scrum, DevOps y Kanban con estándares internacionales de calidad técnica permitió optimizar la eficiencia operativa, reducir considerablemente el tiempo de entrega (lead time), y anticipar la detección de anomalías funcionales y no funcionales mediante validaciones continuas en entornos colaborativos (Ozkaya, 2023).

De acuerdo con Homès (2024), la sinergia entre estos marcos de trabajo y las prácticas normativas consolidadas facilitó evaluaciones más flexibles y adaptativas, habilitando la implementación de frameworks híbridos que fusionaron métodos estructurados de aseguramiento de calidad con dinámicas ágiles de desarrollo; esta combinación mejoró sustancialmente la capacidad de respuesta ante requerimientos cambiantes, redujo errores de integración y fortaleció la alineación del producto con las expectativas reales del entorno empresarial (Sader et al., 2021).

En este escenario, la Estructura de Desglose del Trabajo (EDT), conocida como Work Breakdown Structure (WBS), se consolidó como una herramienta técnica clave para la planificación detallada y jerarquizada de actividades dentro de los proyectos de aseguramiento de calidad; mediante la identificación precisa de paquetes de trabajo, se logró una asignación eficiente de responsabilidades, estimación objetiva del esfuerzo requerido, control efectivo de riesgos y optimización de los recursos técnicos y humanos disponibles (PMI, 2021). Asimismo, la EDT permitió establecer entregables específicos vinculados a las fases de pruebas y certificación, facilitando la ejecución de tareas especializadas como auditorías de código, pruebas automatizadas, validaciones funcionales iterativas y documentación exhaustiva de resultados (Berényi, 2023; Gatto et al., 2023). Un modelo representativo de EDT aplicada al aseguramiento de calidad se presenta a continuación:

Figura 4

Modelo de EDT



Nota. Adaptado de. Six Sigma methodology based on the PMBOK guide in the release management process in software developer enterprise. Gatto, D., Evangelista, J., da Silva, R., & Sassi, R. (2023). <https://doi.org/10.55905/cuadv15n10-035>

Gestión de Proyectos en el Desarrollo de Software

La gestión efectiva de proyectos en el desarrollo de software exige seleccionar estratégicamente el enfoque más adecuado para la planificación, ejecución y control integral del ciclo de vida del producto, considerando factores determinantes como el nivel de incertidumbre, la complejidad tecnológica inherente, el grado de madurez organizacional y las expectativas específicas de los interesados clave. Desde esta perspectiva, la Guía del PMBOK® (Project Management Institute, 2021) establece tres enfoques principales ampliamente adoptados en el ámbito del desarrollo tecnológico: el enfoque predictivo (tradicional o cascada), el enfoque adaptativo (ágil) y el enfoque híbrido (integrador). Cada uno posee características diferenciadoras que ofrecen ventajas significativas según diversos aspectos críticos, como la claridad en la definición de los requisitos iniciales, el grado de riesgo involucrado en el desarrollo y la naturaleza dinámica del entorno operativo del proyecto.

Gestión de Proyectos Tradicional (Ciclo en Cascada)

Según la Guía del PMBOK®, “un enfoque predictivo es útil cuando los requisitos del proyecto y del producto pueden definirse, recopilarse y analizarse al comienzo del proyecto” (PMI, 2021, p. 35). En este modelo, también denominado —en cascada—, la planificación se realiza en las fases iniciales, estableciendo con precisión el alcance, el cronograma, el presupuesto y los recursos necesarios para la ejecución; es particularmente apropiado en contextos donde la incertidumbre es baja, los entregables son bien conocidos desde el inicio y los cambios son mínimos durante el desarrollo.

Este enfoque tradicional se caracteriza por una ejecución secuencial de etapas, desde el análisis de requisitos hasta la entrega final, lo que permite controlar rigurosamente los avances y

minimizar la incertidumbre técnica; sin embargo, su naturaleza rígida limita la capacidad de adaptación ante cambios imprevistos o retroalimentación temprana de los usuarios, aunque es eficaz en proyectos altamente estructurados, puede ser ineficiente cuando los requisitos evolucionan rápidamente, como suele ocurrir en entornos tecnológicos dinámicos (PMI, 2021).

Metodologías Ágiles (Scrum, Kanban, DevOps)

En contraste, los enfoques adaptativos —denominados en muchos casos como metodologías ágiles— son “útiles cuando los requisitos están sujetos a un alto nivel de incertidumbre y volatilidad y es probable que cambien a lo largo del proyecto” (PMI, 2021, p. 38), estas favorecen la entrega continua de valor mediante ciclos cortos de desarrollo y validación, conocidos como iteraciones o sprints, permitiendo incorporar retroalimentación constante por parte de los interesados. Scrum, Kanban y DevOps son ejemplos de marcos de trabajo que se alinean con esta filosofía adaptativa:

Scrum. Como marco ágil de trabajo, tiene relevancia en contextos donde la adaptabilidad, la entrega continua y la colaboración entre equipos resultan fundamentales para responder a las exigencias cambiantes del mercado; según la Guía de Scrum 2020, este no debe entenderse como una metodología rígida, más bien debe entenderse como una estructura ligera que permite abordar problemas complejos mediante soluciones adaptativas, promoviendo la generación de valor de forma continua (Schwaber & Sutherland, 2020); este marco se basa en tres pilares clave: la transparencia, la inspección y la adaptación, principios que guían tanto la interacción entre roles como la gestión empírica del proyecto, facilitando ajustes en tiempo real y reduciendo la incertidumbre técnica y organizacional.

A lo largo del proceso de trabajo con Scrum, se distinguen roles específicos como el Scrum Master, el Product Owner y el equipo de desarrollo, quienes trabajan de manera

coordinada para garantizar que el producto evolucione conforme a las necesidades del cliente y a los objetivos estratégicos del negocio; además, la Guía de 2020 incorpora una perspectiva renovada sobre el “*Equipo Scrum*” como una unidad cohesiva con responsabilidad compartida, lo que elimina la separación tradicional de funciones y promueve una cultura de compromiso colectivo con la entrega del producto.

En ese sentido, las ceremonias de Scrum —como la planificación del Sprint, las reuniones diarias, la revisión del Sprint y la retrospectiva— se convierten en espacios esenciales para fomentar la inspección constante del trabajo y la adaptación de estrategias, asegurando así una mejora continua que impacta positivamente en la calidad de los entregables; así, el equipo trabaja con una lista de elementos priorizados —backlog— seleccionando aquellos que pueden completarse dentro de un sprint (1-4 semanas), al final del cual se presenta un incremento funcional del producto (Schwaber & Sutherland, 2020).

Kanban. Se fundamenta en un enfoque visual para la gestión del trabajo en curso, mediante el uso de tableros y tarjetas que permiten visualizar cada etapa del flujo de tareas; esta herramienta no impone iteraciones fijas ni estructuras rígidas, lo que la convierte en una alternativa flexible y adaptable para equipos que requieren gestionar múltiples actividades de manera simultánea; su principio central consiste en limitar el trabajo en curso (WIP), de modo que se promueva un ritmo de trabajo sostenible, se reduzcan los cuellos de botella y se facilite la entrega continua de valor; en este modelo, la transparencia del flujo de trabajo permite identificar con claridad los bloqueos, optimizar los tiempos de ciclo y fomentar la mejora continua mediante decisiones informadas; así, Kanban se ha convertido en una práctica valiosa tanto para equipos ágiles como para entornos donde la demanda es variable y el trabajo requiere alta capacidad de respuesta (PMI, 2021).

DevOps. DevOps representa una evolución del enfoque ágil, al extender sus principios más allá del desarrollo hacia las operaciones, integrando prácticas de integración continua, entrega continua y automatización; esta filosofía promueve una colaboración estrecha entre desarrolladores, testers y personal de infraestructura, eliminando las barreras tradicionales entre estas funciones para lograr una entrega más rápida, confiable y sostenible del software; DevOps enfatiza la cultura de responsabilidad compartida, la automatización de tareas repetitivas y la monitorización constante, lo que contribuye significativamente a reducir errores humanos, mejorar los tiempos de despliegue y asegurar la calidad del software desde etapas tempranas del ciclo de vida (PMI, 2021).

Herramientas Ágiles (Según Guía del PMBOK® 2021)

Según la Guía del PMBOK® (2021), la selección de herramientas debe adaptarse al entorno, los objetivos del proyecto y las restricciones organizacionales. Estas herramientas no solo permiten visualizar el flujo de trabajo y facilitar la colaboración entre los miembros del equipo, sino que también apoyan la automatización, la integración continua y la entrega rápida de valor. Su uso oportuno potencia la mejora continua, la eficiencia operativa y la trazabilidad del desempeño, siendo fundamentales para aplicar enfoques ágiles de manera efectiva; las herramientas ágiles según la Guía del PMBOK son:

Tabla 2*Herramientas Ágiles en Gestión de Calidad de Software*

Herramienta Ágil	Funcionalidad Principal	Aplicación en Calidad de Software
Jira Software	Gestión de tareas, tableros ágiles, seguimiento de incidencias.	Seguimiento de defectos, gestión de sprints, trazabilidad de pruebas.
Trello	Tableros visuales tipo Kanban.	Organización de tareas de pruebas, priorización de bugs y revisión de ciclos.
Azure DevOps	Integración continua, gestión de código, pipelines de despliegue.	Automatización de pruebas, validación continua, gestión de versiones.
GitLab CI/CD	Repositorio de código, integración y entrega continua, gestión de tareas.	Automatización de pruebas unitarias y funcionales, monitoreo de builds.
TestRail	Gestión de casos de prueba y resultados de pruebas manuales o automáticas.	Documentación de pruebas, control de calidad, análisis de cobertura.
Postman	Automatización y pruebas de APIs.	Validación de servicios web, control de errores en backend.
Selenium	Automatización de pruebas en aplicaciones web.	Ejecución de pruebas funcionales automatizadas, validación de interfaz.
Jenkins	Orquestación de integración continua y automatización de procesos.	Despliegue automático, ejecución de pruebas, control de calidad en pipelines.
Asana	Gestión de tareas colaborativas.	Coordinación de equipos de pruebas, seguimiento de actividades.
Zephyr	Extensión para gestión de pruebas integrada con Jira.	Planificación, ejecución y reporte de pruebas manuales y automáticas.

Nota. La selección de estas herramientas ágiles se fundamenta en su aplicabilidad dentro de entornos adaptativos y su compatibilidad con enfoques de mejora continua, automatización y

trazabilidad de la calidad en el desarrollo de software. Su implementación debe adaptarse a las características del proyecto, el nivel de madurez del equipo y las restricciones organizacionales.

Adaptado de. El estándar para la dirección de proyectos y Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®) (7ª ed.). Project Management Institute (PMI). (2021) <https://www.pmi.org/>

Enfoque Híbrido en Proyectos Tecnológicos

Entre ambos extremos —el predictivo y el adaptativo— surge el enfoque híbrido, descrito por la Guía del PMBOK® como “una combinación de enfoques adaptativos y predictivos. Esto significa que se usan algunos elementos de un enfoque predictivo y otros de un enfoque adaptativo” (PMI, 2021, p. 36); el cual resulta especialmente útil cuando algunos componentes del proyecto pueden definirse y planificarse desde el inicio, mientras que otros requieren mayor exploración o están sujetos a cambios.

El enfoque híbrido se apoya frecuentemente en modelos iterativos e incrementales, “Un enfoque iterativo es útil para aclarar los requisitos e investigar diversas opciones [...] se utiliza un enfoque incremental para producir un entregable a lo largo de una serie de iteraciones. Cada iteración añade funcionalidad dentro de un marco de tiempo predeterminado” (PMI, 2021, p. 36), permitiendo que los entregables evolucionen progresivamente, integrando mejoras continuas, sin sacrificar la planificación anticipada de elementos críticos o de alto riesgo.

En el contexto del desarrollo de software, el enfoque híbrido ofrece una solución estratégica para proyectos que requieren fases de exploración técnica (por ejemplo, en la definición de funcionalidades innovadoras), pero también demandan entregas con alta confiabilidad técnica y cumplimiento regulatorio; así, se puede aplicar un enfoque adaptativo en

la construcción del producto, mientras que el despliegue y la estabilización en producción pueden gestionarse de forma predictiva. Se puede decir por tanto que es cada vez más común en organizaciones que buscan combinar la agilidad operativa con el control tradicional, especialmente en sectores financieros, de salud o gubernamentales, donde el cumplimiento normativo y la trazabilidad son tan importantes como la innovación y la velocidad.

La Guía del PMBOK® y su Aplicación al Aseguramiento de Calidad en Software

La Guía de los Fundamentos para la Dirección de Proyectos (PMBOK® Guide), en su Séptima Edición, elaborada por el Project Management Institute (PMI) (2021), ofreció un marco operativo altamente flexible, concebido para adaptarse a diversos contextos sectoriales y niveles de complejidad. Sus principios, dominios de desempeño, modelos, métodos y artefactos brindaron lineamientos versátiles que resultaron idóneos para proyectos de carácter tecnológicamente exigente, como aquellos enfocados en el aseguramiento de la calidad del software.

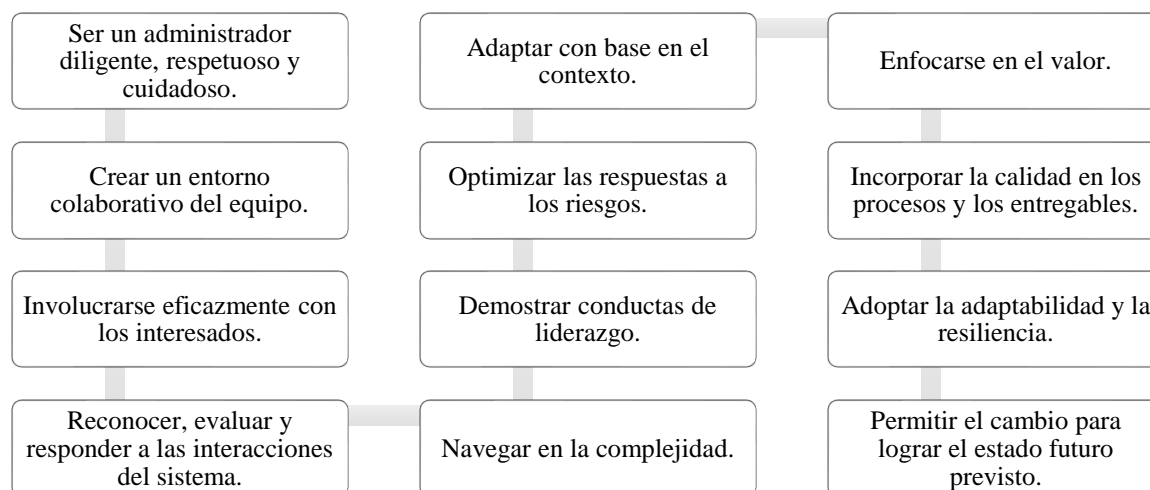
Bajo esta perspectiva, el aseguramiento de la calidad no se concibió como una mera serie de actividades reactivas de inspección o supervisión, sino como un conjunto de prácticas transversales plenamente integradas a lo largo de todo el ciclo de vida del proyecto. La Guía enfatizó concretamente que “la calidad es incorporada en el producto a través de la planificación, el diseño y la ejecución” (PMI, 2021, p. 153), reforzando la idea de adoptar enfoques sistemáticos y proactivos para responder de forma consistente a los requisitos y expectativas del cliente. Este posicionamiento estratégico supuso una transformación en la manera de entender la calidad, abordándola desde la perspectiva de un valor intrínseco que configura e influye de manera continua en cada fase del desarrollo de soluciones tecnológicas.

Estructura Conceptual de la Guía del PMBOK®

La estructura conceptual de la Guía se basa en doce principios fundamentales, entre los cuales destacan aquellos directamente relacionados con la calidad, como “incorporar la calidad en los procesos y los entregables”, “enfocarse en el valor” y “optimizar las respuestas a los riesgos” como se detalla a continuación.

Figura 5

Los 12 Principios de la Guía del PMBOK® Séptima Edición



Nota. La figura presenta los doce principios rectores para la gestión de proyectos según la Guía del PMBOK® Séptima Edición, los cuales orientan la conducta profesional, la toma de decisiones y la adaptación de prácticas según el entorno del proyecto. Adaptado de. El estándar para la dirección de proyectos y Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®) (7ª ed.). Project Management Institute (PMI). (2021) <https://www.pmi.org/>

Estos principios que contribuyen a lograr la estructura de la Guía del PMBOK® también proporcionan una orientación ética y operativa sobre cómo debe conducirse el trabajo del proyecto, más allá del cumplimiento de tareas específicas, promoviendo una cultura organizacional que valore la mejora continua y la excelencia operativa (PMI, 2021).

Dominios de Desempeño de la Guía del PMBOK®

Según la Guía del PMBOK® (2021), los dominios de desempeño son áreas de enfoque interrelacionadas que constituyen una base sólida para la ejecución eficaz de un proyecto. Son ocho:

Tabla 3

Dominios de Desempeño

N.º	Dominio de Desempeño	Enfoque principal
1	Equipo	Cultura, colaboración y desempeño del equipo.
2	Interesados	Compromiso y gestión de interesados clave.
3	Ciclo de Vida y Enfoque de Desarrollo	Elección del ciclo de vida y enfoque de entrega adecuado.
4	Planificación	Integración y adaptación de los planes del proyecto.
5	Trabajo del Proyecto	Entrega efectiva del trabajo y control del proyecto.
6	Entrega	Cumplimiento de los requisitos y satisfacción del cliente.
7	Medición	Evaluación del rendimiento, métricas, análisis de datos.
8	Incertidumbre	Identificación y gestión de riesgos y oportunidades.

Nota. La Tabla presenta los ocho dominios de desempeño definidos en la Guía del PMBOK® (2021), los cuales establecen las áreas críticas que deben gestionarse en todo proyecto. Adaptado de. El estándar para la dirección de proyectos y Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®) (7ª ed.). Project Management Institute (PMI). (2021)

<https://www.pmi.org/>

La Guía incorpora un modelo para analizar e intervenir las distintas dimensiones del proyecto; particularmente, el dominio de Calidad se relaciona con “el cumplimiento de los

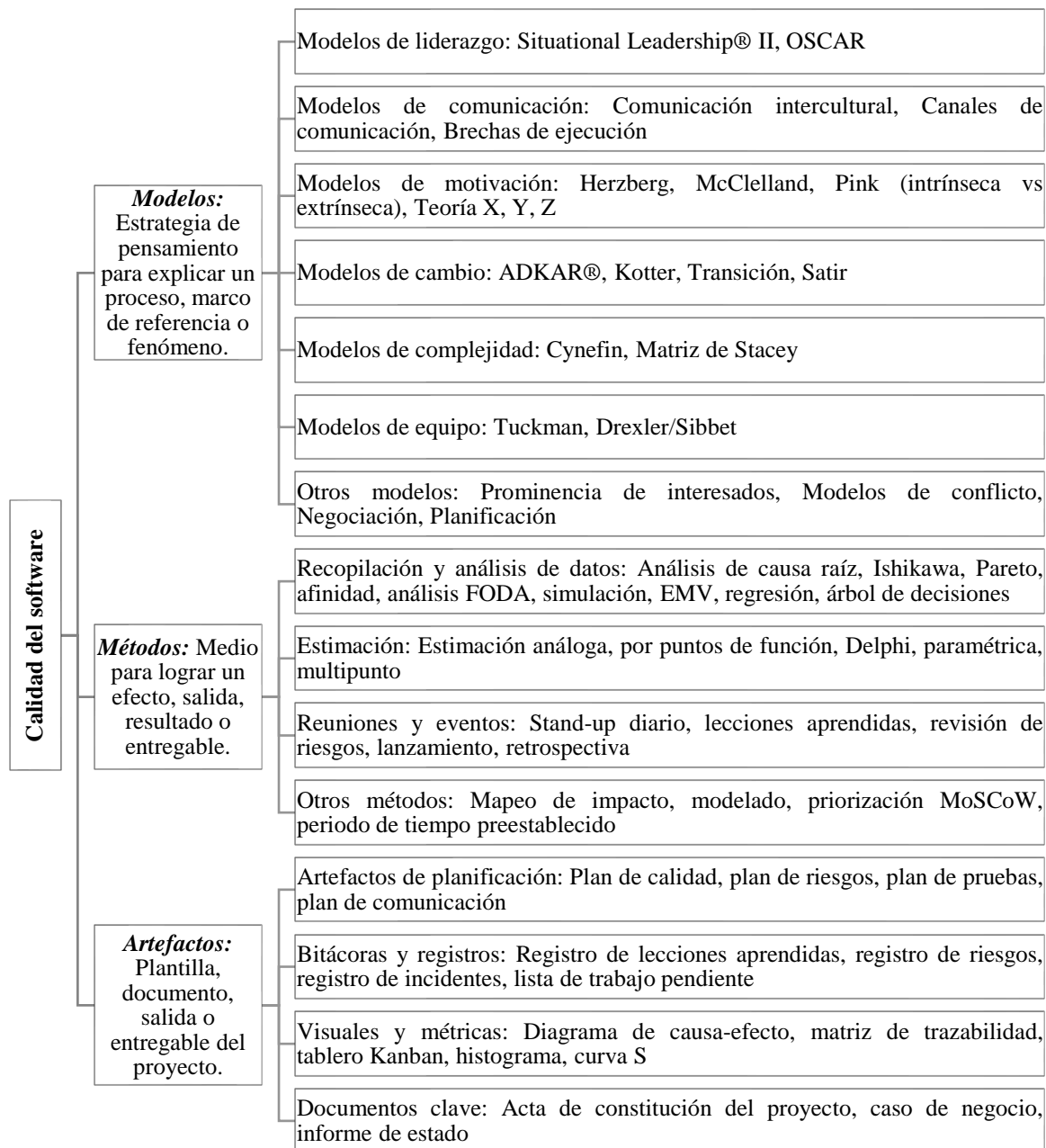
requisitos” y con “la satisfacción de las necesidades y expectativas de los interesados” (PMI, 2021, p. 29), considerando tanto los procesos que generan los entregables como los entregables mismos, y propone que la calidad se planifique, se mida y se gestione de forma proactiva, en lugar de limitarse a inspecciones reactivas.

Herramientas en la Toma de Decisiones

Complementariamente, la Guía del PMBOK® promueve el uso de modelos, métodos y artefactos como herramientas estructuradas que apoyan la toma de decisiones, el análisis de procesos y la mejora del desempeño. Un modelo, según el PMI, es “una estrategia de pensamiento para explicar un proceso, marco de referencia o fenómeno”; un método es “el medio para lograr un efecto, salida, resultado o entregable del proyecto”; y un artefacto es “una plantilla, documento, salida o entregable del proyecto” (PMI, 2021, p. 153).

Figura 6

Modelos, Métodos, Artefactos de la Guía del PMBOK®



Nota. La figura organiza los elementos conceptuales propuestos por la Guía del PMBOK® (2021) para la gestión de calidad en proyectos tecnológicos, clasificándolos en modelos, métodos y artefactos. Adaptado de. El estándar para la dirección de proyectos y Guía de los fundamentos

para la dirección de proyectos (Guía del PMBOK®) (7ª ed.). Project Management Institute (PMI). (2021) <https://www.pmi.org/>

En el campo del aseguramiento de calidad en software, la Guía contempla la aplicación de diversas técnicas analíticas, tales como el análisis de procesos, el análisis de causa raíz, el diagrama de causa y efecto (Ishikawa), el diagrama de Pareto y el diagrama de afinidad, que permiten comprender los orígenes de los defectos, visualizar relaciones causales, clasificar problemas y priorizar acciones correctivas. Dichas herramientas, clasificadas como métodos de “recopilación y análisis de datos”, son ampliamente recomendadas en proyectos donde la calidad está estrechamente ligada al rendimiento de procesos iterativos como ocurre en el desarrollo de software (PMI, 2021, pp. 174–176).

Por otro lado, la Guía también destaca la importancia de los métodos de evaluación del desempeño, tales como el análisis de valor ganado, el análisis de variación y la simulación, entre otros, que permiten medir de forma cuantitativa el progreso del proyecto frente a sus metas, identificar desviaciones y evaluar escenarios futuros. En el ámbito del software, estas herramientas contribuyen a asegurar que los entregables no solo cumplan con los estándares técnicos, sino que también estén alineados con los objetivos estratégicos del negocio.

Finalmente, la Guía del PMBOK® reconoce la necesidad de adaptar todos estos elementos al entorno particular del proyecto. “Los equipos de proyecto deben tener en cuenta las consecuencias al decidir qué elementos usar”, evitando prácticas que “dupliquen esfuerzos, no resulten útiles, produzcan información incorrecta o atiendan necesidades individuales sobre las del equipo” (PMI, 2021, p. 154). Este principio de adaptación y racionalización resulta crucial en

entornos de desarrollo ágil o híbrido, donde las prácticas tradicionales deben combinarse con enfoques flexibles y centrados en el cliente.

Estrategias de Optimización en Procesos de Calidad de Software

En el contexto del aseguramiento de calidad en proyectos tecnológicos, la optimización representa un conjunto de decisiones estratégicas orientadas a mejorar continuamente la eficiencia, la productividad y el cumplimiento de objetivos en entornos complejos como el desarrollo de software; este proceso implica no solo reducir fallas o reprocesos, también incluye anticipar amenazas, aprovechar oportunidades y maximizar la entrega de valor de forma sostenida; en esta línea, la Guía del PMBOK® (2021) plantea un estrategias diferenciadas según se trate de amenazas o de oportunidades:

Tabla 4*Estrategias de Optimización en Procesos de Calidad de Software*

Categoría	Estrategia	Descripción
Gestión de Amenazas	Evitar	Se modifica el plan para eliminar la amenaza o impedir que ocurra; útil cuando el riesgo puede prevenirse por completo desde la planificación.
	Escalar	Se transfiere la responsabilidad de gestión del riesgo a niveles superiores, cuando excede la autoridad del equipo del proyecto.
	Transferir	Se asigna el riesgo a un tercero (proveedor o contratista), quien asume sus posibles consecuencias.
	Mitigar	Se implementan acciones para reducir la probabilidad o el impacto de la amenaza, de forma proactiva.
	Aceptar	Se reconoce el riesgo y no se toman acciones inmediatas, aunque puede desarrollarse un plan de contingencia si es necesario.
Gestión de Oportunidades	Explotar	Se asegura que la oportunidad ocurra para maximizar su impacto positivo en el proyecto.
	Escalar	Se remite la oportunidad al patrocinador u otro nivel estratégico si excede el alcance del equipo.
	Compartir	Se asigna la oportunidad a otra parte interesada que puede aprovecharla mejor, como un socio o colaborador.
	Mejorar	Se actúa para incrementar la probabilidad de que la oportunidad se materialice o amplificar su efecto beneficioso.
	Aceptar	Se reconoce la oportunidad sin acciones inmediatas, quedando disponible si ocurre espontáneamente.
Estrategias Organizacionales	Equipos	Se mantiene el mismo equipo durante el ciclo de vida del producto, lo que fortalece el conocimiento y mejora la calidad del software.
	Estables	

Categoría	Estrategia	Descripción
para Optimización	Financiamiento Incremental	Se revisa el rendimiento en períodos cortos (ej. trimestral) para redirigir inversiones hacia proyectos de mayor valor.
	Dirección de Programas	Se coordina una cartera de proyectos relacionados para entregar beneficios sostenidos a largo plazo y adaptarse a cambios del mercado.

Nota. La Tabla presenta las principales estrategias de optimización en proyectos tecnológicos, clasificadas según su aplicación en la gestión de amenazas, oportunidades y enfoques organizacionales. Adaptado de. El estándar para la dirección de proyectos y Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®) (7ª ed.). Project Management Institute (PMI). (2021) <https://www.pmi.org/>

Estado del Arte: Diagnóstico de la Empresa PDT

La Empresa PDT, a propósito de su rol en el sector tecnológico colombiano, se configura como un actor de alto impacto en la provisión de soluciones de vanguardia para el ámbito financiero nacional. Por lo que respecta a la visión corporativa, la organización se concentra en ser líder en servicios financieros digitales, sustentándose en principios de innovación continua y orientándose a la excelencia en tecnología financiera (fintech) y banca digital. En lo que concierne a su valor estratégico, la Empresa PDT enfatiza la entrega de productos altamente confiables y escalables, respondiendo de manera ágil a los requerimientos de plataformas transaccionales y servicios bancarios críticos.

Con respecto a la estructura interna, PDT opera mediante fábricas de software distribuidas, configuradas por especialidad funcional y técnica. En lo relativo a la dinámica de trabajo, estas fábricas gestionan proyectos adaptativos, combinando marcos tradicionales y

metodologías ágiles. En ese sentido, la Guía del PMBOK® (Séptima Edición) menciona que “el desempeño exitoso del proyecto depende de la capacidad del equipo para seleccionar, adaptar y aplicar las herramientas adecuadas a su entorno” (PMI, 2021, p. 152), lo que motiva a PDT a adoptar un enfoque híbrido que optimiza la calidad y minimiza riesgos. Aún más, la empresa desarrolla soluciones críticas como pasarelas de pago y sistemas de autenticación biométrica, en las cuales la trazabilidad y la seguridad se consideran elementos imprescindibles.

En cuanto a la gestión de proyectos, PDT incorpora eventos iterativos —tales como retrospectivas y revisiones de iteración—, a fin de alinear las exigencias del cliente con la evolución de cada entrega. A este respecto, la Guía del PMBOK® destaca la utilidad de “herramientas estructuradas que apoyan la toma de decisiones, el análisis de procesos y la mejora del desempeño” (PMI, 2021, p. 153). De igual forma, la organización emplea artefactos de seguimiento y control, por añadidura tableros de información y diagramas de desempeño, que facilitan la interpretación de datos relevantes para la evaluación constante de la calidad (PMI, 2021, p. 188).

En lo concerniente a la adaptación metodológica, PDT combina métodos predictivos con prácticas ágiles, por lo que se refiere a la rapidez de respuesta ante cambios y la necesidad de mantener altos estándares de confiabilidad. De igual forma, las fábricas de software realizan análisis de causa raíz y diagrama de Ishikawa para identificar brechas de calidad, logrando una mejora continua acorde con la visión de excelencia empresarial. En realidad, esta combinación de rigor metodológico y flexibilidad operativa refleja la cultura de innovación de la compañía, la cual integra el aprendizaje organizacional como pilar fundamental de su crecimiento.

Por lo que atañe al impacto en la industria, la capacidad de PDT para ofrecer productos digitales confiables, seguros e innovadores se traduce en un fortalecimiento de la banca nacional,

impulsando procesos clave en la operatividad de entidades financieras. Por añadidura, la adopción de enfoques híbridos descritos en la Guía del PMBOK® impulsa la eficiencia de los equipos y fortalece la cohesión entre los distintos roles participantes. De este modo, la cultura de calidad y adaptación metodológica se revela como la variante esencial que consolida la posición competitiva de la Empresa PDT dentro del ecosistema colombiano de servicios tecnológicos.

Marco Legal y Normativo en Calidad de Software

En el desarrollo y aseguramiento de la calidad del software, el cumplimiento normativo es un pilar fundamental para garantizar la seguridad, confiabilidad y eficiencia de los sistemas digitales (Román, 2024). La legislación colombiana ha evolucionado para responder a los desafíos que plantea la transformación digital, estableciendo un marco regulatorio que abarca aspectos clave como la protección de datos personales, la ciberseguridad, la propiedad intelectual del software y la transparencia en la gestión de la información (Naranjo, 2024).

En este contexto, la siguiente tabla presenta un compendio de las normativas más relevantes en Colombia, destacando sus disposiciones principales, las entidades encargadas de su cumplimiento y su impacto en el ciclo de vida del software. La aplicación de estos marcos regulatorios fomenta la adopción de estándares internacionales en la industria tecnológica, promoviendo buenas prácticas en la ingeniería del software y asegurando la calidad de los productos desarrollados.

Tabla 5*Marco Normativo en Calidad de Software*

Norma	Año	Entidad	Contenido	Fuente
Ley 1581 de Protección de Datos Personales	2012	Superintendencia de Industria y Comercio (SIC)	Regula el tratamiento de datos personales en software, garantizando seguridad, confidencialidad y derechos de acceso, actualización y rectificación.	(Congreso de Colombia, 2012).
Ley 1273 de Delitos Informáticos	2009	Fiscalía General de la Nación	Penaliza delitos informáticos como acceso no autorizado, interceptación de datos, daño informático y uso de malware.	(Congreso de Colombia, 2009).
Ley 33 de Protección de Software como Derecho de Autor	1987	Dirección Nacional de Derechos de Autor	Reconoce el software como propiedad intelectual, otorgando protección a su código fuente y funcionalidades.	(Congreso de Colombia, 1987).
Ley 23 de Derechos de Autor	1982	Dirección Nacional de Derechos de Autor	Protege los derechos morales y patrimoniales de los creadores de software.	(Congreso de Colombia, 1982).
Decreto 1360 de Protección de Propiedad Intelectual del Software	1989	Ministerio del Interior	Reglamenta la inscripción de software en el Registro Nacional de Derechos de Autor.	(Presidencia de la República de Colombia, 1989).
Convenio de Budapest contra la Ciberdelincuencia	2020	Ministerio TIC y Fiscalía General de la Nación	Establece normas internacionales para prevenir y sancionar la ciberdelincuencia,	(Consejo de Europa, 2001).

Norma	Año	Entidad	Contenido	Fuente
			promoviendo cooperación internacional.	
Ley 44 de Derechos de Autor	1993	Dirección Nacional de Derechos de Autor	Penaliza el uso ilegal de software sin licencia y la falsificación de programas.	(Congreso de Colombia, 1993).
Ley 1266 de Habeas Data Financiera	2008	Superintendencia Financiera de Colombia	Regula el manejo de información financiera y crediticia, asegurando la seguridad de los datos personales en software bancario.	(Congreso de Colombia, 2008).
Ley 1712 de Transparencia en el Acceso a la Información Pública	2014	Presidencia de la República	Regula el acceso a información pública, promoviendo estándares de datos abiertos en software gubernamental.	(Congreso de Colombia, 2014).
Decreto 1078 de Regulación del Sector TIC	2015	Ministerio TIC	Regula las telecomunicaciones, el espectro radioeléctrico y la infraestructura digital para garantizar conectividad segura.	(Presidencia de la República de Colombia, 2015).
Resolución 460 de Plan Nacional de Infraestructura de Datos	2022	MinTIC	Define estándares de calidad para la gestión de datos en sistemas de información estatales.	(Ministerio de Tecnologías de la Información y las Comunicaciones, 2022).
CONPES 3854 de Política de Ciberseguridad y Ciberdefensa	2016	MinTIC y Presidencia de la República	Establece estrategias para proteger sistemas de información y fortalecer la ciberseguridad nacional.	(Consejo Nacional de Política Económica y Social [CONPES], 2016).

Norma	Año	Entidad	Contenido	Fuente
SIC - Circular Externa 002 sobre IA	2024	Superintendencia de Industria y Comercio	Regula el tratamiento de datos personales en sistemas de inteligencia artificial.	(Superintendencia de Industria y Comercio [SIC], 2024).

Nota. La tabla consolida las principales normas legales nacionales e internacionales que impactan el desarrollo, uso y protección del software en Colombia.

La legislación y regulaciones que enmarcan el desarrollo y el aseguramiento de calidad del software en Colombia configuran un entorno normativo integral que armoniza la innovación tecnológica con la seguridad jurídica y la protección efectiva de los derechos del usuario. En este contexto, normativas específicas como la Ley Estatutaria 1581 de 2012, orientada a garantizar la privacidad, confidencialidad y adecuada gestión de los datos personales en entornos digitales, y marcos estratégicos como el documento CONPES 3854 de 2016, enfocado en fortalecer la ciberseguridad nacional, constituyen pilares fundamentales que guían el establecimiento de estándares rigurosos de calidad, seguridad y control dentro del ciclo de vida del software.

Al adoptar estas disposiciones, las empresas tecnológicas colombianas no solo se alinean con las mejores prácticas internacionales en materia de calidad y protección de datos, sino que además fortalecen significativamente sus procesos internos, asegurando así la integridad, disponibilidad y confidencialidad de la información manejada en sus plataformas tecnológicas. En consecuencia, la implementación efectiva de estos marcos regulatorios trasciende la obligatoriedad jurídica, consolidándose como una ventaja competitiva estratégica que fortalece la confianza de los usuarios, mitiga riesgos operacionales y contribuye decisivamente a la sostenibilidad y desarrollo del sector tecnológico en Colombia.

Metodología

Enfoque de Trabajo

Este estudio adopta un enfoque de investigación mixto integrando técnicas cuantitativas y cualitativas, lo cual permite una comprensión holística, objetiva e interpretativa de los procesos de calidad en el ciclo de vida del desarrollo de software, con especial énfasis en el área de pruebas y certificación. Según Folgueiras (2022), la metodología mixta permite superar las limitaciones de las preguntas cual y cuan y potenciar la capacidad explicativa de los estudios aplicados, al integrar datos de naturaleza diversa bajo un mismo diseño analítico. Esta elección metodológica está alineada con lo planteado por la Guía del PMBOK® (PMI, 2021), el cual resalta que “no existe una única forma correcta de gestionar proyectos” y que la adaptación metodológica debe responder al contexto del proyecto, su complejidad, el entorno organizacional y la naturaleza de los entregables (p. 153).

La orientación mixta permite abordar el fenómeno desde múltiples dimensiones: A) Por un lado, el enfoque cuantitativo posibilita la medición objetiva y sistemática de variables relacionadas con el rendimiento, el costo, el tiempo y la calidad, recurriendo a métodos como el análisis de datos, hojas de verificación, matrices, y análisis de tendencias, entre otros (PMI, 2021, pp. 174-176). B) Por otro lado, el enfoque cualitativo permite capturar el conocimiento tácito (PMI, 2021, p. 153), aquel que no puede codificarse fácilmente y que reside en la experiencia y las percepciones de los actores involucrados. Este tipo de conocimiento es recolectado mediante entrevistas, talleres, foros de discusión y observación directa, los cuales son métodos reconocidos por la Guía del PMBOK® como válidos para vincular a quienes poseen el conocimiento con quienes lo necesitan (PMI, 2021, p. 153).

Además, la Guía del PMBOK® sugiere la combinación de modelos, métodos y artefactos en función de la naturaleza del proyecto; en esta investigación, dicha combinación se aplica mediante el uso de artefactos como los informes, actas y matrices de evaluación; métodos como el análisis de causa raíz, análisis de alternativas, análisis costo-beneficio, análisis FODA, y entrevistas estructuradas; y modelos como la lógica del ciclo de mejora continua y el modelo de procesos de calidad; tal como menciona Folgueiras-Bertomeu (2022) este conjunto de técnicas y paradigma buscan explorar con el fin de lograr expandir resultados.

El análisis de los datos recolectados se estructura según lo dispuesto por la Sección 4.6 de la Guía del PMBOK®, en la cual se especifica que los métodos de recopilación y análisis de datos “se utilizan para recopilar, valorar y evaluar datos e información con el fin de obtener una comprensión más profunda de una situación” (PMI, 2021, p. 174). De este modo, los resultados obtenidos en esta fase no sólo informan las decisiones, sino que también contribuyen al diseño de estrategias de optimización viables y alineadas con las mejores prácticas internacionales.

Finalmente, la elección del enfoque mixto también responde a la necesidad de integrar análisis justificativos del negocio, como lo señala la Guía del PMBOK® al mencionar que los proyectos deben estar sustentados en evaluaciones de valor, retorno de inversión (ROI), análisis de viabilidad y estudios comparativos (PMI, 2021, p. 174), elementos fundamentales cuando se busca mejorar la eficiencia operativa y la calidad en entornos altamente regulados y tecnológicos como el de la empresa PDT.

Diseño de Investigación

El diseño metodológico del presente proyecto responde a un enfoque no experimental, de tipo descriptivo y exploratorio, lo que implica que no se manipulan intencionadamente variables independientes, sino que se analizan fenómenos en su contexto natural, permitiendo comprender

en profundidad las dinámicas actuales del área de pruebas y certificación de software en la empresa objeto de estudio. Este tipo de diseño resulta especialmente apropiado cuando se busca examinar procesos reales sin intervención directa del investigador, y en situaciones donde las variables no pueden ser controladas de forma ética o práctica. Tal como lo explica Landero (2021), la investigación no experimental “se realiza cuando, durante el estudio, el investigador no puede controlar, manipular o alterar a los sujetos, sino que se basa en la interpretación o las observaciones para llegar a una conclusión”, observando los fenómenos “exactamente cómo ocurrieron” (p. 3).

Desde esta perspectiva, el estudio pretende identificar patrones operativos, deficiencias estructurales y oportunidades de mejora en los procesos actuales del aseguramiento de calidad en PDT, además, se abordan eventos ocurridos en el pasado reciente que inciden directamente en la eficiencia y trazabilidad del proceso de pruebas, lo cual es coherente con el carácter retrospectivo y contextual que define a los diseños no experimentales, según los principios metodológicos descritos por Landero (2021).

Este diseño se justifica en función de los objetivos del estudio, que requieren tanto la recolección y análisis sistemático de datos sobre las deficiencias en los procesos de calidad, como la evaluación de alternativas y la propuesta de estrategias de mejora. Por tanto, se integra una estructura metodológica basada en los enfoques sugeridos por la Guía del PMBOK®, particularmente en las prácticas vinculadas con el análisis de datos, la gestión de calidad y la evaluación del desempeño del proyecto.

La investigación se fundamenta en el Modelo de Valor de Entrega de la Guía del PMBOK® (PMI, 2021), que promueve una gestión adaptativa, orientada a la generación de beneficios sostenibles, considerando las necesidades del entorno organizacional y los cambios en

el entorno operativo. Este enfoque permite implementar técnicas analíticas que faciliten el diagnóstico de desviaciones y el diseño de estrategias de mejora continua. En este contexto, el análisis de datos es una práctica central para “valorar y evaluar datos e información con el fin de obtener una comprensión más profunda de una situación” (PMI, 2021, p. 174).

Tipo de Estudio

El presente estudio se enmarca dentro de un enfoque aplicado, orientado a la solución práctica de un problema específico dentro del contexto organizacional: la optimización de los procesos de calidad en el área de pruebas y certificación de software en una empresa tecnológica. Este tipo de investigación se caracteriza por buscar resultados que puedan ser implementados directamente en entornos reales para mejorar procesos, resolver deficiencias estructurales o fortalecer capacidades institucionales. Según Hernández et al. (2006), la investigación aplicada tiene como finalidad resolver problemas particulares, con la intención de incidir en la mejora de procesos, servicios o productos dentro de un ámbito determinado.

Este tipo de estudio no busca únicamente generar conocimiento teórico, sino proponer soluciones viables, medibles y sostenibles, apoyadas en buenas prácticas de la gestión de proyectos, y orientadas al fortalecimiento de los procesos de aseguramiento de calidad de software. La naturaleza aplicada del estudio se alinea con el principio del PMBOK® denominado “Adaptabilidad y resiliencia”, el cual sugiere que la dirección de proyectos debe responder dinámicamente a las necesidades organizacionales mediante la integración estratégica del conocimiento generado (PMI, 2021, p. 27).

En este sentido, el estudio incorpora métodos cualitativos y cuantitativos de recolección y análisis de información, alineados con el Modelo de Valor de Entrega del PMBOK®, el cual promueve la generación de resultados que aporten valor tangible a los interesados y fortalezcan

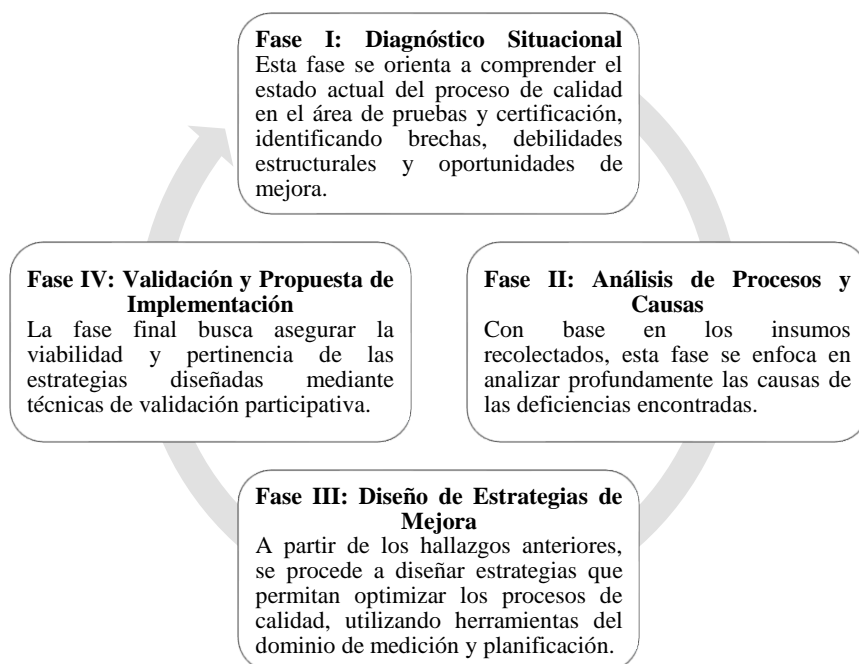
las capacidades organizacionales (PMI, 2021, pp. 7–8). Así, la investigación trasciende el diagnóstico para construir propuestas de mejora estructuradas, apoyadas en herramientas como el análisis causa-raíz, el análisis de procesos, la matriz DOFA, el análisis de interesados y la evaluación de alternativas, todos reconocidos dentro del marco de la Guía del PMBOK® como técnicas robustas de gestión (PMI, 2021, pp. 174–177).

Fases del Marco de Trabajo

En alineación con el enfoque adaptativo y basado en principios que propone la Guía del PMBOK® Séptima Edición, este proyecto adopta un marco de trabajo cíclico y secuencial compuesto por cuatro fases interrelacionadas, cada una sustentada en modelos, métodos y artefactos recomendados por el Project Management Institute (PMI, 2021, p. 154), permitiendo una integración flexible del conocimiento, considerando el contexto del proyecto, sus restricciones y las necesidades cambiantes de los interesados.

Figura 7

Fases del Marco de Trabajo



Nota. La figura presenta la secuencia metodológica utilizada para estructurar el proceso de mejora en la gestión de calidad del software, compuesta por cuatro fases: diagnóstico situacional, análisis de causas, diseño de estrategias y validación. Adaptado de. El estándar para la dirección de proyectos y Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®) (7ª ed.). Project Management Institute (PMI). (2021) <https://www.pmi.org/>

La Fase I es crucial para sentar las bases del diseño estratégico, fundamentando las decisiones posteriores en datos concretos y en un conocimiento contextualizado de la organización; mientras que la Fase II de carácter analítico, permite generar hallazgos relevantes que orientan la toma de decisiones estratégicas, al entender el origen sistémico de los problemas; además, la Fase III está orientada a la generación de valor, como lo plantea el modelo de entrega de valor de la Guía del PMBOK® al articular soluciones alineadas con los objetivos organizacionales y las expectativas de los interesados (PMI, 2021, p. 8); finalmente, la Fase IV consolida el ciclo metodológico, permitiendo que los resultados del estudio no se limiten al diagnóstico, sino que se conviertan en insumos para la toma de decisiones estratégicas y la implementación progresiva de mejoras sostenibles.

Técnicas, Herramientas y Artefactos de la Guía del PMBOK®

En el marco de la Guía del PMBOK® cada fase contempla técnicas específicas de análisis y herramientas reconocidas en la gestión de proyectos, adaptadas al contexto organizacional y alineadas con los dominios de desempeño definidos por el PMI. La siguiente tabla presenta un desglose detallado de las técnicas aplicadas en cada fase, junto con su fundamentación conceptual y referencia correspondiente:

Tabla 6*Aplicación de la Guía del PMBOK® por Fase del Proyecto*

Fase	Método	Detalle
Fase I: Diagnóstico Situacional	Encuestas estructuradas	Aplicación de cuestionarios con preguntas cerradas y escala Likert, dirigidos a los actores clave del área de calidad, con el fin de recopilar datos cuantificables sobre las prácticas actuales (PMI, 2021, p. 103).
	Matriz DOFA (SWOT Analysis)	Utilizada para evaluar fortalezas, oportunidades, debilidades y amenazas del entorno organizacional (PMI, 2021, p. 177).
	Registro de lecciones aprendidas	Reconocido como artefacto clave del aprendizaje organizacional, que contribuye a identificar patrones de errores o éxitos pasados (PMI, 2021, p. 179).
	Análisis de interesados	Definido como una técnica para “recopilar y analizar de manera sistemática información cuantitativa y cualitativa acerca de los interesados, a fin de determinar qué intereses deben tenerse en cuenta a lo largo del proyecto” (PMI, 2021, p. 177).
Fase II: Análisis de Procesos y Causas	Análisis de causa raíz (Root Cause Analysis)	Como técnica para “determinar la causa subyacente básica para una variación, un defecto o un riesgo” (PMI, 2021, p. 177).
	Diagrama de Ishikawa (espina de pescado)	Como representación gráfica para organizar causas potenciales de problemas en categorías (PMI, 2021, p. 174).
	Diagrama de Pareto	Permite priorizar las causas de mayor impacto, siguiendo el principio 80/20, y facilitando el enfoque en lo verdaderamente relevante (PMI, 2021, p. 174).
	Diagrama de afinidad	Útil para agrupar ideas, opiniones y causas de manera estructurada, facilitando la síntesis del análisis (PMI, 2021, p. 174).

Fase	Método	Detalle
	Análisis de procesos	Definido como “una revisión sistemática de los pasos y procedimientos para realizar una actividad” (PMI, 2021, p. 176), que contribuye a identificar actividades que no generan valor o generan cuellos de botella.
Fase III: Diseño de Estrategias de Mejora	Análisis de alternativas	Como método para “evaluar las opciones identificadas a fin de seleccionar las más apropiadas” (PMI, 2021, p. 174).
	Análisis costo-beneficio	Herramienta que “determina los beneficios proporcionados por un proyecto respecto a sus costos”, permitiendo tomar decisiones basadas en el valor esperado (PMI, 2021, p. 175).
	Mapeo de flujo de valor (Value Stream Mapping)	Puede ser usado para visualizar y optimizar el flujo de información y materiales en los procesos (PMI, 2021, p. 177).
Fase IV: Validación y Propuesta de Implementación	Juicio de expertos	Descrito por la Guía del PMBOK® como una técnica en la que “se aplica el conocimiento y experiencia en áreas especializadas para validar decisiones y soluciones” (PMI, 2021, p. 174).
	Reuniones de revisión y retrospectiva	Donde se analizan los entregables con stakeholders clave, generando consensos y observaciones constructivas (PMI, 2021, p. 179).
	Lecciones aprendidas	Nuevamente utilizadas como herramienta de cierre para capitalizar el conocimiento generado y promover una cultura de mejora continua.
	Análisis de escenarios tipo “¿Qué pasa si...?”	Como técnica para simular los efectos de diferentes cursos de acción (PMI, 2021, p. 177).

Nota. La tabla describe la metodología estructurada utilizada en las distintas fases del proyecto, integrando técnicas y herramientas propuestas por la Guía del PMBOK® (2021). Adaptado de. El estándar para la dirección de proyectos y Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®) (7ª ed.). Project Management Institute (PMI). (2021)

<https://www.pmi.org/>

Para garantizar la trazabilidad metodológica entre los objetivos específicos del estudio y las actividades desarrolladas durante el proyecto, se establece una relación estructurada que permite vincular cada propósito con su correspondiente fase de ejecución, las acciones concretas a realizar, las herramientas metodológicas adoptadas desde la Guía del PMBOK® y los productos o entregables resultantes.

Tabla 7*Relación entre Objetivos Específicos, Fases del Proyecto*

Objetivo Específico	Fase del Proyecto	Actividades Por Desarrollar	Modelo o Método (Guía del PMBOK®)	Producto o Entregable
Diagnosticar las deficiencias operacionales del área de pruebas de software.	Fase 1: Diagnóstico situacional	Contextualización del equipo sobre el proyecto Aplicación de encuestas y entrevistas Revisión de registros históricos Análisis de stakeholders Aplicación de matriz DOFA Revisión de lecciones aprendidas	Análisis FODA (p.177) Juicio de expertos (p.174) Hoja de verificación (p.175) Análisis de interesados (p.177) Revisión documental	Actas de reunión Resultados de encuestas y entrevistas Matriz DOFA Documento de diagnóstico
Analizar los procesos de calidad dentro del ciclo de vida del desarrollo de software.	Fase 2: Análisis de procesos y causas	Aplicación de análisis de causa raíz Elaboración de diagrama de Ishikawa Análisis de procesos Elaboración de diagrama de Pareto Aplicación de técnica de tormenta de ideas Clasificación con diagrama de afinidad	Análisis causa raíz (p.177) Análisis de procesos (p.176) Diagrama de causa-efecto (p.177) Diagrama de afinidad (p.176) Tormenta de ideas (p.174)	Diagramas de Ishikawa, Pareto y afinidad Informe de hallazgos Base de datos con información categorizada

Objetivo Específico	Fase del Proyecto	Actividades Por Desarrollar	Modelo o Método (Guía del PMBOK®)	Producto o Entregable
Diseñar estrategias de optimización para los procesos de calidad.	Fase 3: Diseño de estrategias de mejora	Realización de sesiones de ideación Evaluación de alternativas Estimación del ROI y NPV Evaluación de impacto	Análisis de alternativas (p.174) ROI, NPV (p.175) Evaluación de impacto (métodos del negocio)	Informe de estrategias Mapa de impacto de alternativas Roadmap de implementación
Socializar los resultados con los involucrados.	Fase 4: Validación y propuesta de implementación	Presentación de resultados a stakeholders Validación con juicio de expertos Retrospectivas Documentación de lecciones aprendidas	Juicio de expertos (p.174) Reuniones de lecciones aprendidas (p.179) Comunicación con interesados	Presentaciones Actas de reunión Documento de validación Informe final del proyecto

Nota. La tabla presenta la relación estructurada entre los objetivos específicos del proyecto, las fases de ejecución, las actividades principales a desarrollar, los modelos y métodos propuestos en la Guía del PMBOK® (2021), y los productos o entregables correspondientes. Adaptado de. El estándar para la dirección de proyectos y Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®) (7ª ed.). Project Management Institute (PMI). (2021) <https://www.pmi.org/>

Población y Muestra

La población objeto de estudio corresponde a los colaboradores del área de pruebas y certificación de la empresa PDT, organización del sector tecnológico que desarrolla soluciones de software para entidades del sistema financiero colombiano. Esta población está conformada

por perfiles técnicos y estratégicos que participan activamente en el ciclo de vida del desarrollo de software, particularmente en las actividades vinculadas al aseguramiento de calidad y testing.

Para esta investigación, se adoptó una muestra no probabilística de tipo intencional, compuesta por 54 empleados seleccionados con base en su nivel de conocimiento, experiencia y participación directa en los procesos analizados, la cual incluye analistas de pruebas, testers manuales y automatizadores, líderes de pruebas, desarrolladores, líderes de desarrollo, Scrum Masters, analistas de requerimientos y otros perfiles clave del área técnica. Este tipo de muestra se justifica en función de los objetivos del estudio, los cuales requieren acceso directo a la experiencia especializada de los actores involucrados en la problemática; tal como señalan Hernández-Sampieri et al. (2006), en las muestras no probabilísticas “la elección de los elementos no depende de la probabilidad, sino de causas relacionadas con las características de la investigación o de quien hace la muestra” (p. 241); en este caso, la selección se basó en criterios estratégicos que aseguran la relevancia y profundidad del análisis de los procesos de calidad observados.

Este procedimiento se encuentra en línea con las orientaciones metodológicas presentes en el Dominio de Desempeño de los Interesados (Guía del PMBOK®, 2021, Sección 2.1), donde se reconoce la importancia de identificar y comprometer a las partes interesadas con mayor relevancia e influencia dentro del contexto del proyecto. Asimismo, se vincula con el Dominio de Desempeño del Trabajo del Proyecto (Sección 2.5), donde se destaca la necesidad de incorporar recursos humanos adecuados que contribuyan al cumplimiento de los objetivos, así como con el Dominio de Medición (Sección 2.7), donde se aborda la importancia de contar con fuentes válidas y significativas para el análisis.

Técnicas de Recolección de Datos

Las técnicas de recolección de datos permiten obtener información confiable, válida y contextualizada para responder a los objetivos del estudio; ahora bien, en el marco de este proyecto, se integran técnicas cualitativas y cuantitativas conforme al enfoque mixto adoptado, lo cual posibilita una visión más completa del fenómeno analizado. Según Hernández et al. (2006), la selección de técnicas debe guardar coherencia con el tipo de estudio y el diseño metodológico, considerando tanto el acceso a la población como el tipo de datos requeridos para el análisis. Por su parte, Folgueiras (2022) destaca que el uso combinado de enfoques permite articular distintas fuentes de evidencia empírica, favoreciendo la triangulación y el contraste entre perspectivas objetivas y subjetivas.

Adicionalmente, la Guía del PMBOK® (Project Management Institute, 2021) señala que las técnicas de recopilación de datos forman parte de los métodos comúnmente utilizados en la gestión de proyectos para generar información robusta que apoye decisiones estratégicas informadas, enfatizando su papel en procesos como la planificación, el control de calidad y la gestión del conocimiento (Sección 4.4.1). En este estudio, las técnicas seleccionadas se estructuran bajo las siguientes modalidades: encuestas estructuradas, entrevistas semiestructuradas, revisión documental y análisis de registros históricos del área de calidad, garantizando así un abordaje integral de la problemática:

Encuestas Estructuradas

Las encuestas estructuradas permiten recolectar información cuantificable de forma estandarizada, lo que facilita la comparación de datos, la medición de variables clave y la identificación de tendencias en las prácticas de aseguramiento de calidad de software. Este instrumento consiste en un conjunto de preguntas cerradas diseñadas para evaluar dimensiones

como los tiempos de ejecución de pruebas, el uso de herramientas tecnológicas, la frecuencia de defectos y los niveles de satisfacción. Según Hernández et al. (2006), “el cuestionario es uno de los instrumentos más utilizados para recolectar datos, pues permite medir variables específicas de forma sistemática, ya sea de manera presencial, telefónica o digital” (p. 309). Además, la Guía del PMBOK® (2021) reconoce los cuestionarios como métodos formales de recolección de datos dentro del ámbito de la gestión de proyectos, especialmente en contextos donde se requiere información estructurada y procesable para la toma de decisiones (Sección 4.4.1).

Entrevistas Semiestructuradas

Las entrevistas semiestructuradas permiten acceder a datos cualitativos profundos, orientados a comprender percepciones, experiencias y juicios críticos de los actores involucrados en los procesos de calidad. Este formato combina una guía temática con flexibilidad discursiva, facilitando la espontaneidad en las respuestas sin perder el enfoque analítico. Según Hernández et al. (2006), “la entrevista personal es un contexto ideal para aplicar cuestionarios o guías de preguntas abiertas, permitiendo captar matices y significados que difícilmente se obtienen mediante instrumentos cerrados” (p. 335). Por su parte, la Guía del PMBOK® (2021) destaca que las entrevistas son técnicas esenciales para explorar temas complejos y captar el conocimiento tácito de los expertos, especialmente en entornos de alta incertidumbre o proyectos que requieren adaptación metodológica (Sección 4.4.1).

Revisión de Documentación Interna, Procesos y Lecciones Aprendidas

La revisión documental constituye una técnica indispensable para el análisis de información secundaria contenida en registros históricos, manuales de procedimiento, informes de pruebas, actas de retrospectiva y lecciones aprendidas. Este proceso permite reconstruir el comportamiento pasado del sistema de calidad y evaluar patrones de desempeño. Como afirman

Hernández et al. (2006), esta técnica “consiste en el análisis sistemático de documentos que contienen datos relevantes para el problema de investigación, incluyendo informes, archivos institucionales y publicaciones técnicas” (p. 333). En línea con esta visión, la Guía del PMBOK® (2021) identifica los artefactos organizacionales, como los registros de riesgos, informes de desempeño y matrices de seguimiento, como elementos clave para el análisis retrospectivo y la mejora continua en proyectos (Secciones 4.6.2 y 4.6.7).

Sesiones Colaborativas y Talleres con los Equipos de QA y Desarrollo

Las sesiones grupales —como retrospectivas, talleres técnicos o reuniones de análisis de causa— son espacios participativos orientados a la construcción colectiva de conocimiento y la identificación colaborativa de oportunidades de mejora. Estas dinámicas permiten aplicar herramientas como la tormenta de ideas, los diagramas de afinidad, el análisis de causa raíz y el mapeo de flujo de valor, fortaleciendo la integración entre actores y promoviendo la inteligencia colectiva. De acuerdo con Hernández et al. (2006), “la combinación de técnicas cualitativas en espacios colaborativos permite profundizar en el análisis de procesos grupales, identificar relaciones causales y generar propuestas de mejora de manera participativa” (p. 336). En la misma línea, la Guía del PMBOK® (2021) recomienda estas técnicas como métodos efectivos de recopilación y análisis de información, alineados con el Dominio de Desempeño del Equipo y el Dominio de los Interesados, los cuales promueven la cocreación de valor y el aprendizaje organizacional (Secciones 2.1, 2.2 y 4.6.6).

Validación de la Encuesta

Previo a la aplicación general de la encuesta estructurada, se llevó a cabo un proceso de validación mediante juicio de expertos, con el fin de asegurar la pertinencia, claridad y relevancia de cada ítem. Para ello, se implementó el siguiente proceso:

Tabla 8*Flujo de Validación de la Encuesta*

Pasos	Detalle
Diseño inicial del instrumento	Redacción de preguntas cerradas con escalas Likert alineadas a los objetivos del estudio.
Revisión por juicio de expertos	Selección de 3 expertos en calidad de software y metodología de evaluación. Revisión de coherencia, redacción y pertinencia de ítems. Ajustes menores en redacción de enunciados y escalas.
Aplicación de prueba piloto	Selección de 7 colaboradores con distintos roles del área de pruebas de PDT. Aplicación individual del cuestionario bajo condiciones controladas.
Análisis de consistencia interna	Consolidación de respuestas en matriz de datos. Cálculo del Alfa de Cronbach para evaluar fiabilidad del instrumento.
Ajustes finales	Eliminación o reescritura de ítems con baja correlación. Confirmación de estructura definitiva del cuestionario.
Implementación de la encuesta validada	Aplicación a la muestra total (54 participantes).

Nota. La tabla representa las etapas metodológicas seguidas para validar el instrumento de recolección de datos aplicado en la investigación. Incluye desde el diseño inicial hasta la implementación definitiva, incorporando revisión por expertos, aplicación piloto y análisis estadístico mediante el coeficiente Alfa de Cronbach.

Cálculo del Alfa de Cronbach

El Alfa de Cronbach (α) es un coeficiente que evalúa la consistencia interna de un conjunto de ítems o preguntas que miden una misma dimensión o constructo. Se interpreta como una medida de fiabilidad o coherencia interna: mientras más cercano a 1, mayor fiabilidad tiene el instrumento (Tavakol & Dennick, 2011).

Formula general:

$$\alpha = \frac{N \cdot \bar{c}}{\bar{v} + (N - 1) \cdot \bar{c}}$$

Donde:

N = número de ítems del cuestionario

\bar{c} = covarianza promedio entre los ítems

\bar{v} = varianza promedio de los ítems

Número de ítems: 12 preguntas en escala Likert (1-5)

Prueba piloto: aplicada a 7 personas

Las respuestas fueron codificadas en una matriz 7x12

Se calculó con software estadístico (como SPSS o Excel con complemento RealStats)

Tabla 9

Nivel de Fiabilidad de Alfa de Cronbach

Valor de α	Nivel de fiabilidad
> 0.9	Excelente
0.8 - 0.9	Buena
0.7 - 0.8	Aceptable
0.6 - 0.7	Cuestionable
< 0.6	Deficiente o inaceptable

Nota. La tabla presenta la interpretación de los valores del coeficiente alfa de Cronbach, utilizado para evaluar la fiabilidad interna de instrumentos de medición. Adaptado de. Making sense of Cronbach's alpha. Tavakol, M., & Dennick, R. (2011). <https://doi.org/10.5116/ijme.4dfb.8dfd>

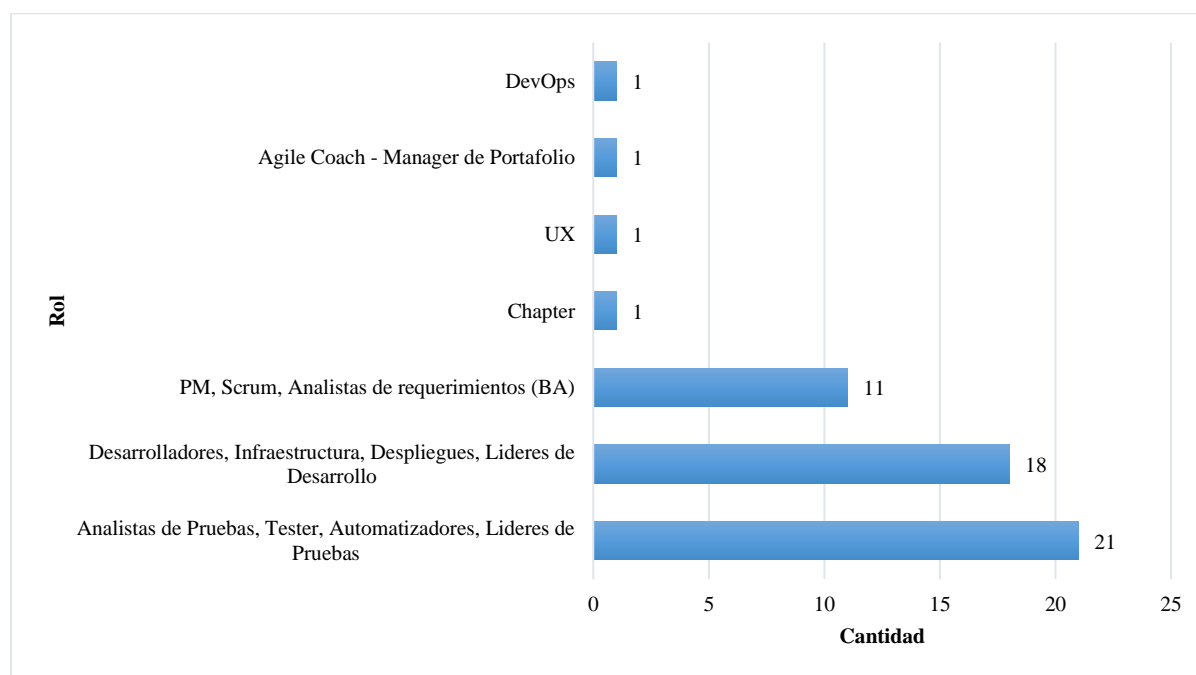
Análisis de Resultados

Diagnóstico de las Deficiencias Operacionales del Área de Pruebas de Software

Como punto de partida del proyecto, la fase de diagnóstico situacional se centró en identificar las practicas existentes en el área de pruebas de software en la empresa PDT; para ello, las técnicas aplicadas incluyeron encuestas, entrevistas, revisión documental de registros históricos y consolidación de lecciones aprendidas; estos participantes según la segmentación de equipo ocupan los siguientes roles:

Figura 8

Segmentación de Equipos por Roles



Nota. La figura muestra la distribución de los roles participantes en el proceso de evaluación del aseguramiento de calidad en software dentro de la Empresa PDT.

Tabla 10*Roles de los Colaboradores Encuestados*

¿Cuál es su rol en un proyecto?	Q	%
Analistas de Pruebas, Tester, Automatizadores, Lideres de Pruebas	21	39%
Desarrolladores, Infraestructura, Despliegues, Lideres de Desarrollo	18	33%
PM, Scrum, Analistas de requerimientos (BA)	11	20%
Chapter	1	2%
UX	1	2%
Agile Coach - Manager de Portafolio	1	2%
DevOps	1	2%
Total, general	54	100.00%

Nota. Distribución de roles de los encuestados en los proyectos de desarrollo y aseguramiento de calidad.

Con el fin de conocer cómo se conectan estos roles con las practicas generales en el área de pruebas de software en PDT se realiza la siguiente matriz:

Tabla 11

Matriz de Prácticas Generales en el Área de Pruebas de Software de PDT

Rol	Prácticas Generales	Herramientas	Impacto	Conexión con la Calidad de los Datos de Prueba
Tester funcional / Analista QA	Diseño de casos de prueba con base en criterios de aceptación definidos. Ejecución manual de pruebas en ambientes controlados. Reporte de defectos con trazabilidad. Validación de flujos funcionales completos. Participación en demos de producto para revisión de funcionalidades.	Jira Zephyr Xray Excel TestRail	Garantiza la cobertura de funcionalidades. Reduce errores en producción. Mejora la comunicación con desarrollo.	Asegura consistencia entre escenarios funcionales y datos usados. Detecta datos erróneos o incompletos en etapas tempranas.
Tester automatizador	Creación de scripts de regresión automatizada. Mantenimiento de suites automatizadas en pipelines. Validación continua de funcionalidades críticas. Implementación de pruebas en paralelo por microservicio. Uso de buenas prácticas en programación para pruebas.	Selenium Cypress Jenkins GitLab CI Postman	Aumenta la eficiencia del ciclo de pruebas. Reduce tiempos de validación y errores repetitivos	Permite usar grandes volúmenes de datos simulados con precisión. Fortalece la cobertura en diferentes combinaciones de datos
Líder de QA	Planeación de pruebas por release o sprint. Coordinación de entregables de pruebas con otros roles. Definición de estrategias de cobertura y riesgos. Acompañamiento en el PI Planning y retros.	Confluence Jira Power BI Excel Trello	Fortalece la alineación del equipo. Eleva la trazabilidad de los resultados.	Supervisa que los datos estén correctamente documentados.

Rol	Prácticas Generales	Herramientas	Impacto	Conexión con la Calidad de los Datos de Prueba
	Consolidación de métricas de calidad por proyecto.		Mejora la calidad entregada al cliente.	Promueve su estandarización entre equipos.
Desarrollador	Programación con enfoque en calidad y pruebas unitarias. Integración con herramientas de análisis de código. Generación de datos de prueba en ambiente de desarrollo. Validación cruzada con analistas QA. Corrección de defectos detectados en etapas tempranas.	Junit SonarQube GitHub IntelliJ Docker	Reduce defectos técnicos y de lógica. Fortalece la calidad estructural del software.	Crea datos de entrada coherentes con requisitos. Apoya la generación de casos complejos para QA.
Scrum Master / PM	Seguimiento del avance de pruebas y bloqueos. Coordinación de sesiones de retrospectiva con foco en calidad. Apoyo en solución de impedimentos técnicos. Comunicación de prioridades con enfoque en calidad. Verificación de cumplimiento de criterios de entrega.	Jira Miro Confluence MS Teams	Asegura el cumplimiento del proceso ágil. Mejora la coordinación entre áreas.	Requiere reportes con datos confiables. Valida entregables basados en evidencias claras.
Analista de Negocio (BA)	Identificación de criterios funcionales clave. Participación en el refinamiento de historias. Revisión de entregables con enfoque en valor al usuario. Clarificación de condiciones de aceptación con QA. Validación de resultados desde la perspectiva del negocio.	Confluence Lucidchart Excel Reuniones técnicas	Minimiza la ambigüedad en requisitos. Permite pruebas mejor enfocadas. Mejora satisfacción del cliente.	Define los datos que deben reflejar los casos reales del negocio. Contribuye a crear pruebas más representativas.
DevOps	Configuración de ambientes de pruebas estables. Automatización del proceso de despliegue. Monitoreo de pipelines de CI/CD. Soporte en pruebas de rendimiento y carga.	Jenkins Docker Kubernetes GitLab CI/CD Grafana	Estabiliza el ciclo de desarrollo. Asegura que las pruebas se ejecuten en condiciones óptimas.	Replica ambientes de producción para datos precisos. Facilita el versionamiento de datos usados por QA.

Rol	Prácticas Generales	Herramientas	Impacto	Conexión con la Calidad de los Datos de Prueba
	Generación de snapshots para validaciones repetitivas.			
UX / Diseño	Revisión de interfaces para validar consistencia visual.	Figma	Mejora la experiencia	Asegura que los datos sean claros.
	Testeo de usabilidad con usuarios reales.	Adobe XD	percibida por el usuario	claros.
	Apoyo en historias centradas en experiencia de usuario.	InVision	final.	Accesibles y
	Validación de flujos de navegación; Prototipado interactivo para pruebas.	Maze Miro	Previene errores de diseño.	comprensibles en las interfaces.
Manager de portafolio / Agile Coach	Evaluación de madurez de prácticas en pruebas.	Power BI	Mejora la alineación entre	Solicita datos consistentes
	Revisión de indicadores clave de calidad por célula.	Dashboards	áreas.	y verificables.
	Impulso a la cultura de mejora continua.	internos	Permite medir avances en	Exige trazabilidad de las
	Mentoría en prácticas ágiles de QA.	PM Canvas	calidad organizacional.	métricas para auditorías y
	Definición de OKRs con enfoque en calidad.	KPIs		mejora.

Nota. La presente matriz resume las prácticas generales más representativas observadas en el área de pruebas de software de PDT, clasificadas según los roles clave que intervienen en el ciclo de vida del desarrollo.

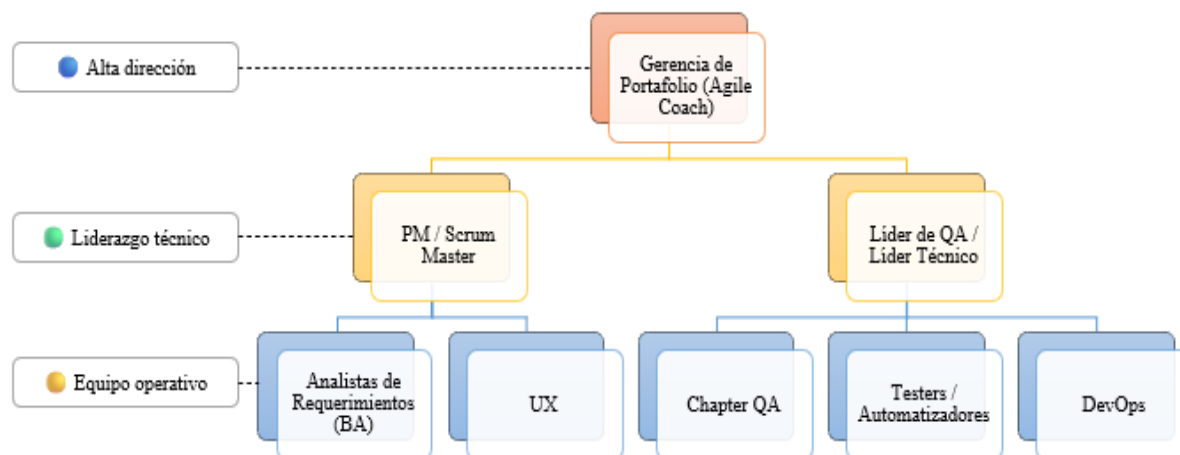
En la primera reunión de apertura del proyecto (ver Anexo A), se contextualizó al equipo sobre los objetivos de mejora, se socializaron los instrumentos de recolección de datos y se construyó de manera colaborativa un mapa de interesados, lo cual permitió identificar los actores clave involucrados en el aseguramiento de calidad y asignar responsabilidades técnicas para la fase diagnóstica. En esta sesión, se destacó el liderazgo de los Scrum Masters, líderes de pruebas y desarrolladores, así como el apoyo transversal de roles estratégicos como el Agile Coach.

La segunda sesión de trabajo (ver Anexo B) se orientó a la validación de hallazgos preliminares obtenidos mediante encuestas y revisión de registros. En este espacio, el equipo técnico participó activamente en la identificación y categorización de problemas críticos del proceso de pruebas, empleando herramientas como el análisis de afinidad y la priorización por impacto. Se discutieron causas recurrentes de ineficiencia operacional, lo que permitió construir una visión compartida sobre el estado actual del área de QA y su relación con los entregables estratégicos del negocio.

El Stakeholder Map correspondiente a esta reunión ubicó en la zona de alta influencia y participación a los Scrum Masters, Project Managers y Líderes de QA, quienes asumieron un papel de liderazgo tanto metodológico como técnico; seguidamente, en un próximo nivel, se identificaron actores operativos como testers, automatizadores, desarrolladores y analistas de requerimientos, fundamentales para aportar insumos empíricos a través de encuestas y entrevistas. Finalmente, en un nivel de influencia estratégica más general, se ubicó al Agile Coach, el Chapter y el rol de UX, quienes contribuyeron desde la visión de experiencia del usuario, lineamientos de portafolio y gobernanza de calidad. Este mapeo se presenta como la Figura 9.

Figura 9

Mapa de Interesados - Reunión de Apertura del Proyecto



Nota. La figura representa la estructura jerárquica del equipo vinculado al proceso de aseguramiento de calidad en software dentro de la Empresa PDT, diferenciando claramente los niveles de alta dirección, liderazgo técnico y operación.

El instrumento de encuesta fue aplicado a 54 colaboradores de la empresa PDT, representando roles técnicos y estratégicos involucrados directamente en proyectos de desarrollo y aseguramiento de calidad. La encuesta fue diseñada en cinco secciones que abordaron temas clave: percepción del proyecto, problemas generales, limitaciones en proveedores, debilidades en la planeación y deficiencias en la ejecución de pruebas.

Los resultados se consolidaron en cinco bloques críticos, cada uno con cinco problemas representativos (total: 25 ítems), cuyas frecuencias fueron homogéneas, reflejando la existencia de patrones persistentes más que tendencias aisladas, los bloques analizados en cada pregunta fueron los siguientes:

Tabla 12*Preguntas Asociadas a la Encuesta*

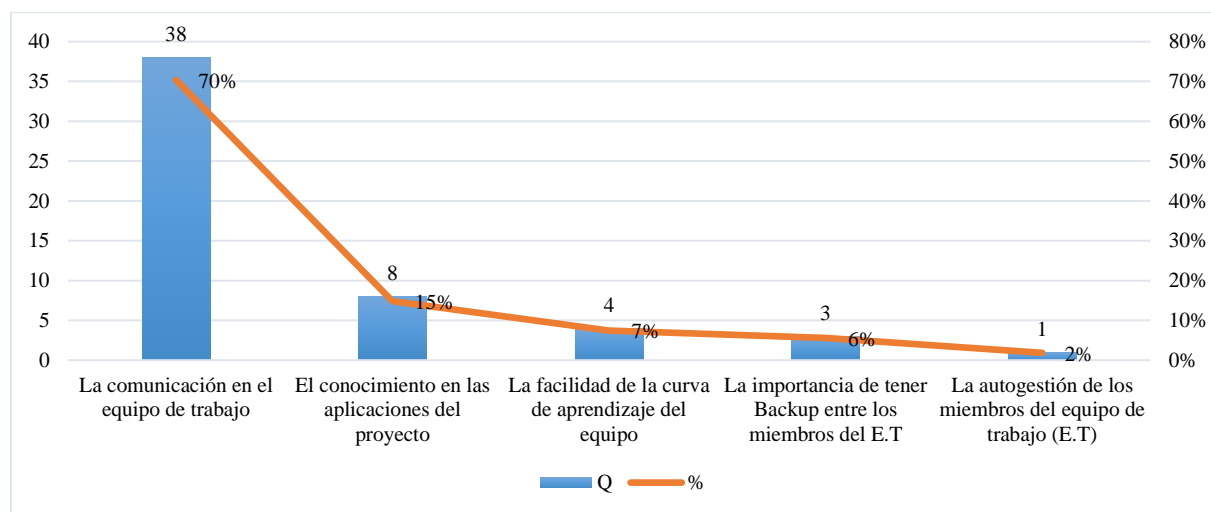
General	Detalle	Total
¿Qué tan importante es un proyecto?	[El conocimiento en las aplicaciones del proyecto]	1
	[La autogestión de los miembros del E.T]	1
	[La comunicación en el equipo de trabajo]	1
	[La facilidad de la curva de aprendizaje del equipo]	1
	[La importancia de tener Backup entre los miembros del E.T]	1
¿Cuáles son los principales problemas de un proyecto?	[El alcance no está bien definido]	1
	[Motivación del Equipo de Trabajo]	1
	[Planeación reducida (todo es para ya)]	1
	[Problemas de presupuesto]	1
¿Cuáles son los problemas de las fábricas de Software (Proveedores)?	[Recurso Humano con Experiencia]	1
	[Alta rotación de los colaboradores]	1
	[Desconocimiento del negocio (falta de capacitación)]	1
	[Falta de auditoria para el proceso de conciliación]	1
¿Cuáles son los problemas del Proceso de calidad de Software - Planeación?	[Falta de innovación en herramientas tecnológicas]	1
	[Presupuesto insuficiente por parte del cliente]	1
	[Aplicación parcial de técnicas de pruebas]	1
	[Ausencia de aplicación de metodologías]	1
¿Cuáles son los problemas del Proceso de calidad de Software - Ejecuciones?	[Ausencia de planes de pruebas]	1
	[Desactualización de las tendencias tecnológicas]	1
	[Falta de métricas e indicadores]	1
	[Alto proceso de documentación (roba tiempo)]	1
¿Cuáles son los problemas del Proceso de calidad de Software - Ejecuciones?	[Ausencia de planes de mejora]	1
	[Bug no reportados]	1
	[Falta de automatización de Flujos Principales]	1
	[Re-ejecuciones sin valor agregado]	1
Total, general		25

Nota. La tabla consolida las cinco categorías temáticas principales evaluadas mediante el instrumento de encuesta aplicado a los colaboradores del área de calidad y desarrollo.

El siguiente gráfico muestra los factores que los colaboradores consideran más relevantes para el éxito de un proyecto:

Figura 10

¿Qué Tan Importante es un Proyecto?



Nota. La figura muestra los factores internos que influyen en el desempeño del equipo de trabajo, destacando la comunicación como el aspecto más crítico (70 %), seguido por el conocimiento en aplicaciones del proyecto y la curva de aprendizaje.

Tabla 13

Factores Clave para el Éxito de los Proyectos de Software

Factor	Q	%
La comunicación en el equipo de trabajo	38	70%
El conocimiento en las aplicaciones del proyecto	8	15%
La facilidad de la curva de aprendizaje del equipo	4	7%
La importancia de tener Backup entre los miembros del (E.T)	3	6%
La autogestión de los miembros del equipo de trabajo (E.T)	1	2%

Nota. La tabla presenta la priorización de factores considerados críticos para el éxito de los proyectos de software, basada en la percepción de los colaboradores encuestados.

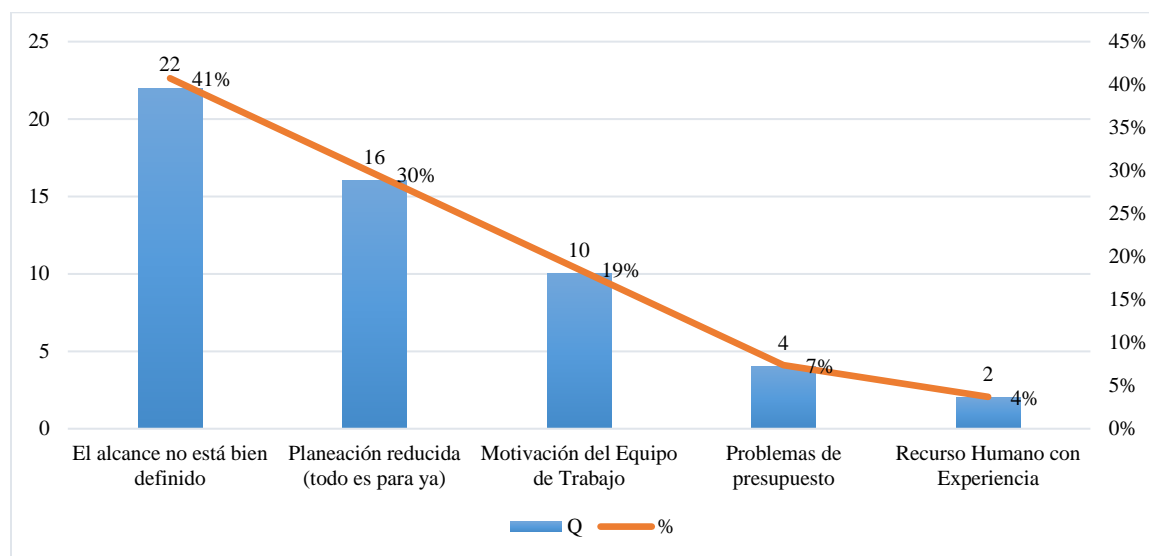
La variable correspondiente a la comunicación dentro del equipo de trabajo evidencia una superioridad significativa en relación con las demás, concentrando aproximadamente el 70% de las respuestas agregadas; este porcentaje no solo la posiciona como el factor predominante en el conjunto de indicadores evaluados, sino que permite inferir, con base en la distribución de frecuencias, que la fluidez comunicativa constituye el principal facilitador transversal de la eficiencia operativa, la sincronización técnica de tareas y el cumplimiento oportuno de los objetivos funcionales, particularmente en entornos de alta complejidad como el área de pruebas y certificación de software.

Ahora bien, si se examinan las demás variables —entre ellas, la autogestión de los miembros del equipo técnico, la facilidad de la curva de aprendizaje y la disponibilidad de mecanismos de respaldo funcional (backups)—, se advierte que sus porcentajes individuales son considerablemente más bajos, situándose en rangos menores al 15%, lo cual evidencia una percepción subordinada respecto a su nivel de criticidad.

Este comportamiento estadístico sugiere una priorización colectiva del trabajo colaborativo como eje estructural de la dinámica organizacional, aunque, al mismo tiempo, podría estar revelando una subvaloración de factores estratégicos asociados a la autonomía profesional, la gestión del conocimiento distribuido y la redundancia operativa, todos ellos esenciales para mitigar riesgos asociados a la dependencia técnica y la discontinuidad en el flujo de procesos.

Figura 11

¿Cuáles son los Principales Problemas de un Proyecto?



Nota. La figura presenta los principales factores que afectan la eficiencia del proyecto desde una perspectiva de gestión, destacando la falta de definición del alcance (41 %) y la planeación reducida (30 %) como los obstáculos más frecuentes.

Tabla 14

Principales Problemas Identificados en los Proyectos de Software

Problema	Q	%
El alcance no está bien definido	22	34%
Planeación reducida (todo es para ya)	18	62%
Motivación del Equipo de Trabajo	10	77%
Problemas de presupuesto	4	83%
Recurso Humano con Experiencia	4	100%

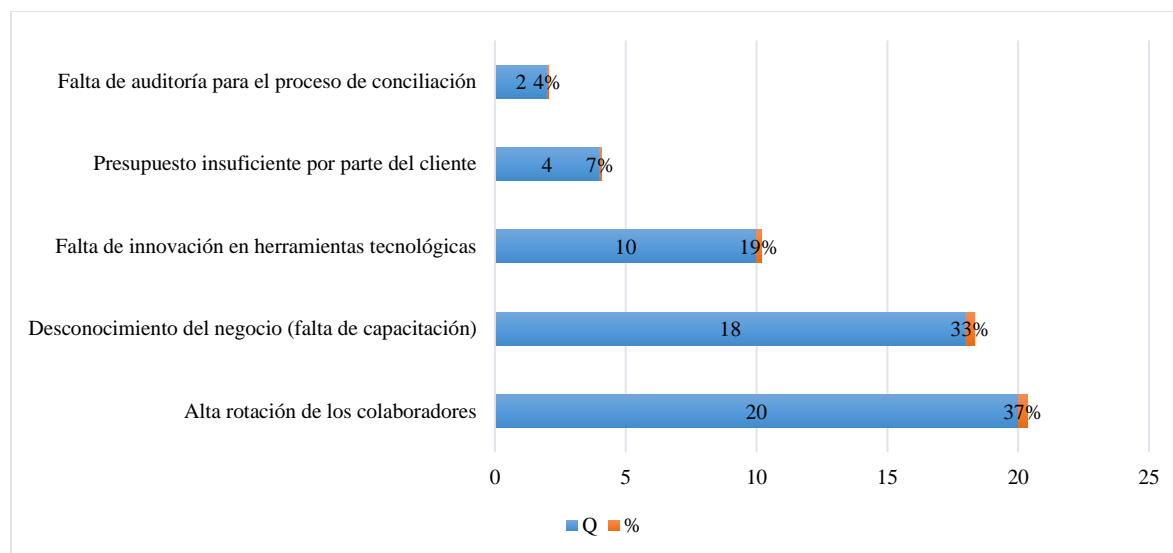
Nota. La tabla resume los problemas más críticos detectados en la gestión de proyectos de software de la empresa PDT, de acuerdo con los resultados de la encuesta aplicada a los colaboradores.

El principal obstáculo percibido corresponde a la falta de definición clara del alcance del proyecto, aspecto que acumula el 41% de las menciones y que, en consecuencia, debe considerarse prioritario dentro de la gestión operativa. Este hallazgo sugiere que, en ausencia de una delimitación precisa de los objetivos, las metas y los entregables, se produce un efecto en cadena que impacta negativamente la planificación, el uso racional de recursos y la alineación de expectativas entre los diferentes actores; además, incrementa el margen de ambigüedad y dificulta el control de los cambios que puedan surgir a lo largo del ciclo de vida del proyecto, el cual comprende desde la iniciación hasta el cierre, según los lineamientos del Project Management Institute (PMI).

Adicionalmente, con una incidencia del 30%, se encuentra la problemática asociada a la planeación reducida o improvisada, la cual refleja una cultura organizacional centrada en la ejecución acelerada de tareas, sin una evaluación previa de viabilidad técnica ni un análisis de riesgos; por tanto, esta condición compromete tanto la calidad de los resultados como la capacidad de respuesta ante eventualidades operativas, ya que impide anticipar escenarios y establecer planes de contingencia robustos que permitan mitigar impactos adversos.

Figura 12

¿Cuáles son los Problemas de las Fábricas de Software (Proveedores)?



Nota. La figura expone los principales factores que afectan la estabilidad y eficiencia operativa en los equipos de trabajo, destacando la alta rotación de colaboradores (37 %) y el desconocimiento del negocio (33 %) como los más críticos.

Tabla 15

Principales Problemas Asociados a las Fábricas de Software (Proveedores)

Problema	Q	%
Alta rotación de los colaboradores	20	37%
Desconocimiento del negocio (falta de capacitación)	18	33%
Falta de innovación en herramientas tecnológicas	10	19%
Presupuesto insuficiente por parte del cliente	4	7%
Falta de auditoría para el proceso de conciliación	2	4%

Nota. La tabla expone los problemas más relevantes percibidos en los proveedores de servicios de software para la empresa PDT, según la encuesta aplicada. Construcción propia, 2025.

El principal problema identificado en las fábricas de software corresponde a la alta rotación del personal, la cual presenta una frecuencia significativa del 37% dentro del total de menciones recogidas, lo que permite inferir que esta variable constituye un obstáculo estructural para la eficiencia operativa en este tipo de entornos; dicha rotación genera múltiples consecuencias que afectan negativamente el desempeño global del equipo, entre las cuales se destacan la fuga de conocimiento tácito y especializado, el aumento de reprocesos, la mayor incidencia de errores humanos y la pérdida de tiempo productivo derivado del reentrenamiento constante de nuevos colaboradores, lo que a su vez repercute en el deterioro del clima organizacional y en la fragmentación del entendimiento colectivo respecto al alcance del proyecto, entendido este como el conjunto de objetivos, entregables y restricciones previamente definidos.

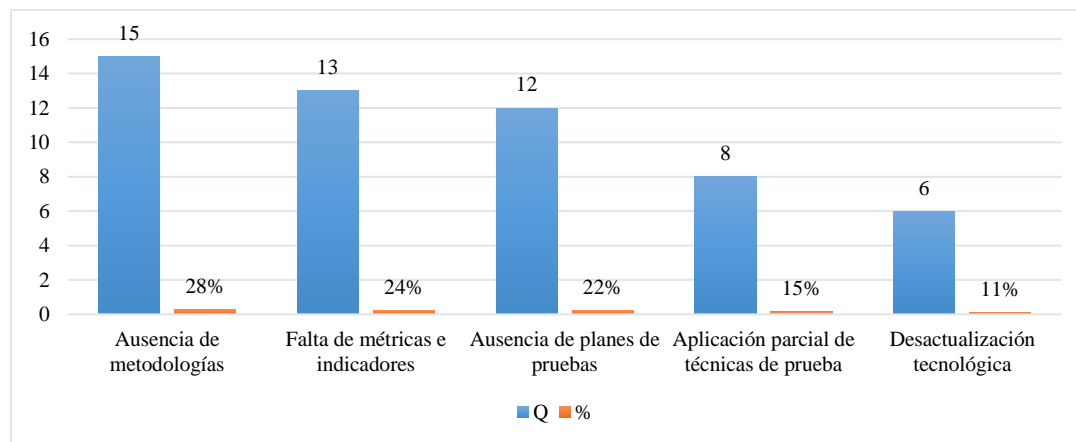
De manera complementaria, el desconocimiento del negocio, que ocupa el segundo lugar en frecuencia, se relaciona con deficiencias en los procesos de inducción funcional y formación técnica, pues al no proporcionar al equipo una capacitación adecuada, se limita su capacidad de interpretar las necesidades reales del cliente, alinearse con los objetivos estratégicos establecidos y tomar decisiones informadas durante la ejecución del proyecto; en este sentido, la carencia de dominio sobre el contexto empresarial impide establecer una visión holística del producto o servicio y, por tanto, restringe el aporte de valor desde la etapa de desarrollo.

Asimismo, la ausencia de innovación tecnológica en el entorno de trabajo, especialmente en lo que respecta a la actualización del stack técnico —conjunto de herramientas, lenguajes, marcos y plataformas utilizados en la construcción de software— representa un factor crítico que, aunque menos mencionado de forma explícita, incide de manera directa sobre la calidad de los entregables, la velocidad en los ciclos de desarrollo y el grado de automatización alcanzado

en los procesos clave; en efecto, la falta de evolución en las herramientas conlleva un rezago competitivo frente a estándares de la industria y limita la capacidad de adaptación a las exigencias del mercado.

Figura 13

¿Cuáles son los Problemas del Proceso de Calidad de Software - Planeación?



Nota. La figura muestra las principales deficiencias técnicas identificadas en los procesos de calidad del software, siendo la ausencia de metodologías (28 %) y la falta de métricas e indicadores (24 %) las más frecuentes, seguidas por la ausencia de planes de prueba, aplicación parcial de técnicas y desactualización tecnológica.

Tabla 16

Problemas Identificados en la Fase de Planeación del Proceso de Calidad

Problema identificado	Q	%
Ausencia de metodologías	15	28%
Falta de métricas e indicadores	13	24%
Ausencia de planes de pruebas	12	22%
Aplicación parcial de técnicas de prueba	8	15%
Desactualización tecnológica	6	11%

Nota. La tabla refleja las principales deficiencias detectadas durante la etapa de planeación en los procesos de aseguramiento de calidad de software en la empresa PDT.

Se observa que la ausencia de aplicación de metodologías constituye el problema más reportado durante la fase de planeación del proceso de calidad del software, concentrando el 28% del total de menciones; este hallazgo refleja una debilidad estructural en cuanto a la estandarización de los procedimientos de aseguramiento de calidad, entendida esta como la implementación sistemática de buenas prácticas que garanticen la consistencia en las actividades de validación, verificación y prueba. Como consecuencia, dicha omisión genera inconsistencias tanto en la planificación como en la ejecución de las pruebas, lo que dificulta la replicabilidad de los resultados, debilita la trazabilidad de los errores y restringe la capacidad de mejora continua en contextos altamente dinámicos.

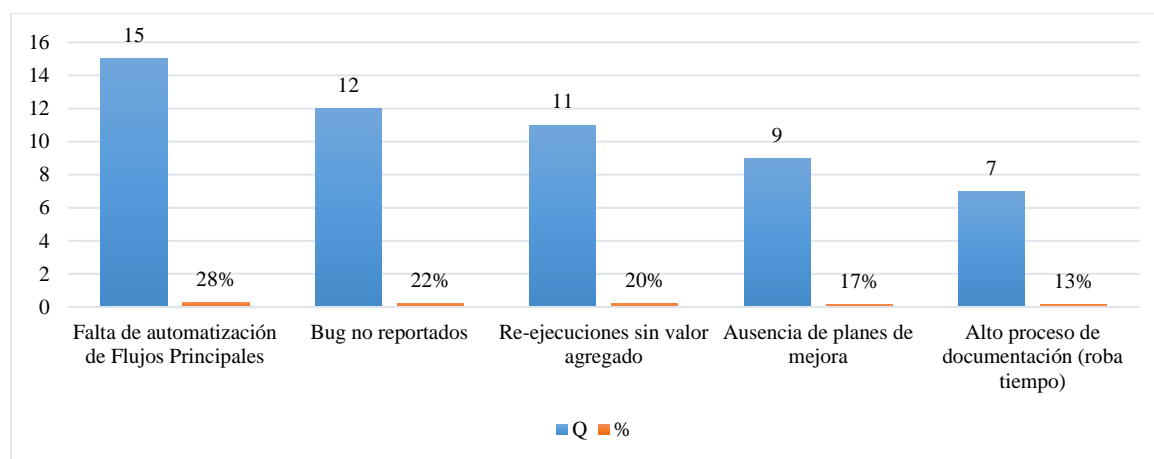
Por otra parte, la falta de métricas e indicadores representa el 24% de las menciones, mientras que, en segundo lugar, la ausencia de planes de prueba concentra el 22%; ambas situaciones evidencian carencias significativas en los mecanismos de monitoreo, control y estructuración formal del testing, entendido este último como el conjunto de técnicas destinadas a verificar el correcto funcionamiento de un sistema. En este sentido, la falta de visibilidad sobre los resultados esperados, así como la ausencia de parámetros cuantificables para evaluar el rendimiento del área, impiden un análisis objetivo de desempeño y limitan la calidad de la toma de decisiones estratégicas.

De igual forma, aunque con menor frecuencia relativa, emergen dos problemáticas adicionales: por una parte, la aplicación parcial de técnicas de prueba y, por otra, la desactualización tecnológica, las cuales, si bien no encabezan la lista de prioridades, no pueden

ser ignoradas en la evaluación global del proceso. En efecto, la cobertura funcional del sistema —entendida como la capacidad del proceso de pruebas para validar todos los escenarios críticos de uso— puede verse comprometida cuando se aplican únicamente técnicas fragmentadas o cuando las herramientas utilizadas no están alineadas con los estándares vigentes en la industria del software.

Figura 14

¿Cuáles son los Problemas del Proceso de Calidad de Software - Ejecuciones?



Nota. La figura ilustra los principales factores que afectan la eficiencia en los procesos de prueba de software, destacándose la falta de automatización de flujos principales (28 %), la presencia de bugs no reportados (22 %) y las re-ejecuciones sin valor agregado (20 %) como los más representativos.

Tabla 17*Problemas Identificados en la Fase de Ejecución del Proceso de Calidad*

Problema	Q	%
Falta de automatización de Flujos Principales	15	28%
Bug no reportados	12	22%
Re-ejecuciones sin valor agregado	11	20%
Ausencia de planes de mejora	9	17%
Alto proceso de documentación (roba tiempo)	7	13%

Nota. La tabla evidencia las principales deficiencias en la fase de ejecución del aseguramiento de calidad en la empresa PDT, destacándose la falta de automatización en flujos principales y la presencia de bugs no reportados como los problemas más recurrentes que afectan la eficiencia operativa y la calidad de los entregables.

El principal problema identificado en el proceso de aseguramiento de calidad del software corresponde a la falta de automatización en los flujos principales, situación que representa una barrera crítica para la eficiencia operativa, especialmente cuando se trata de procesos repetitivos o de alta frecuencia que podrían optimizarse mediante la implementación de frameworks automáticos, entendidos estos como estructuras predefinidas que permiten ejecutar tareas de prueba de forma sistemática y reproducible; dicha carencia, además de ralentizar el ciclo de desarrollo, incrementa la probabilidad de errores humanos, reduce la escalabilidad de las soluciones y limita la capacidad del equipo de QA para focalizarse en actividades de mayor valor agregado.

Seguidamente, se identifican otras problemáticas relevantes, tales como la presencia de bugs no reportados y la ejecución recurrente de reprocesos sin valor agregado, los cuales reflejan deficiencias en los sistemas de trazabilidad, es decir, en la capacidad de registrar, rastrear y

analizar los eventos ocurridos durante el ciclo de pruebas; esta situación afecta directamente el control de versiones, complica la identificación oportuna de regresiones y limita el aprendizaje organizacional, ya que impide construir una base de conocimiento reutilizable en futuros desarrollos.

Por otra parte, también se reporta con frecuencia la ausencia de planes de mejora y un alto nivel de documentación manual, lo que genera impactos negativos tanto en la productividad del equipo como en los tiempos de entrega del producto final; la excesiva dependencia de registros manuales, además de consumir recursos operativos, introduce una variabilidad indeseada en la calidad de la información registrada, dificultando la estandarización y el seguimiento de los avances.

Los resultados obtenidos a través del instrumento de encuesta permitieron establecer un panorama cuantitativo claro sobre las principales deficiencias operacionales del área de pruebas de software en la empresa PDT. Cada uno de los bloques temáticos evidenciaron patrones recurrentes que limitan la calidad, eficiencia y capacidad de respuesta del equipo.

No obstante, aunque los datos estadísticos resultaron esenciales para la identificación y jerarquización de problemas, se diseñó una entrevista semiestructurada orientada a ampliar y profundizar el diagnóstico situacional desde el lente estratégico, técnico y organizacional. Las respuestas recolectadas a través de esta técnica cualitativa permitieron interpretar la dinámica real del área de testing, los vacíos estructurales en la gestión de calidad, y las posibles rutas de mejora sostenible (Anexo C).

La información obtenida a partir de las entrevistas a actores clave del proceso fue sometida a un análisis cualitativo el cual se realizó utilizando el software especializado Atlas.Ti25. Para ello, se definieron siete categorías temáticas, cada una asociada a una de las

preguntas formuladas durante la entrevista, permitiendo así una segmentación analítica precisa del discurso de los participantes. Las categorías fueron las siguientes:

Barreras para la calidad, asociada a la percepción de los obstáculos estructurales en el área de pruebas.

Alineación entre roles, centrada en la sinergia —o falta de ella— entre los equipos de desarrollo, QA y producto.

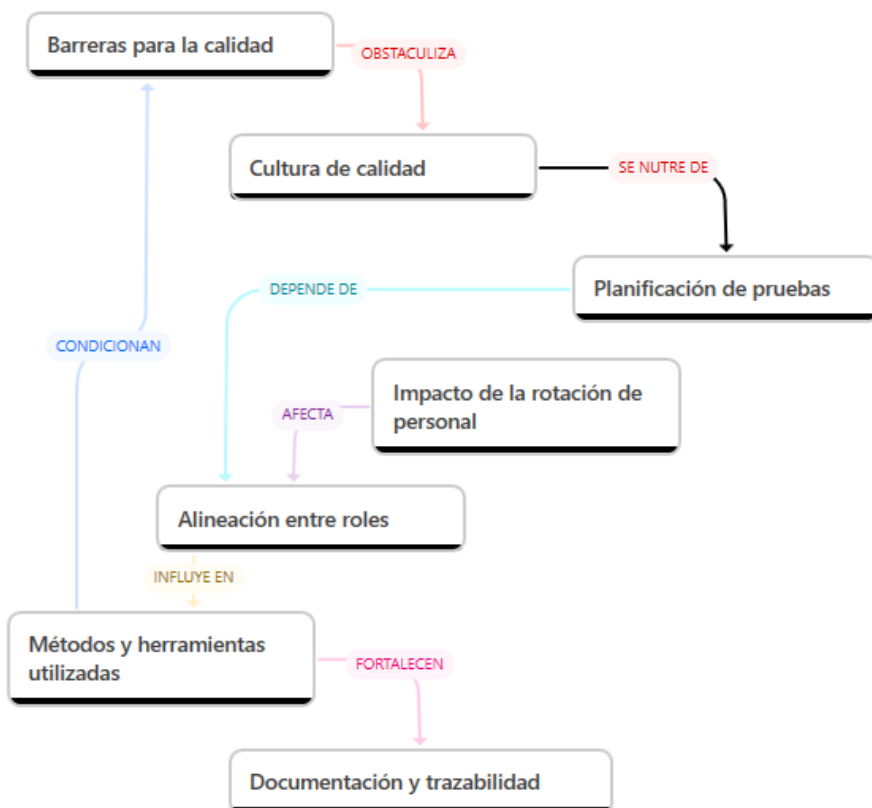
Impacto de la rotación de personal, categoría que agrupó los efectos operativos derivados de la desestabilización de los equipos.

Métodos y herramientas utilizadas, vinculada al grado de adecuación entre prácticas implementadas y necesidades reales del proyecto.

Planificación de pruebas, relacionada con la anticipación de actividades de aseguramiento y sus implicaciones en la calidad.

Documentación y trazabilidad, referida al nivel de soporte técnico e histórico disponible para el análisis de fallos y la mejora continua.

Cultura de calidad, categoría transversal orientada a identificar estrategias sostenibles que fortalezcan la orientación a la excelencia dentro de entornos de alta demanda.

Figura 15*Mapa Relacional de Códigos*

Nota. La figura representa un mapa conceptual de los factores que inciden en la calidad del software, evidenciando cómo las barreras para la calidad condicionan la cultura organizacional, la cual se ve afectada por la planificación de pruebas, la alineación de roles y el impacto de la rotación de personal. Atlas.Ti.25

Las barreras para la calidad actúan como un factor estructurante que condiciona el resto de los componentes del proceso; la cultura de calidad, a su vez, se ve obstaculizada por dichas barreras, pero también se alimenta directamente de la planificación de pruebas, lo que demuestra su carácter dinámico. Esta planificación depende del impacto de la rotación de personal, el cual

mientras que criterio refleja la importancia de contar con parámetros comunes —como los criterios de aceptación— que orienten la validación del producto. Finalmente, entendimiento evidencia que las divergencias en la interpretación de las tareas o requisitos son una fuente constante de reprocesos, lo cual refuerza la necesidad de fortalecer el diálogo funcional entre los roles técnicos, operativos y estratégicos.

Aunque existen marcos ágiles y ceremonias establecidas, los actores destacan la persistencia de desajustes conceptuales entre los equipos:

“En la práctica, esa alineación es parcial. Aunque trabajamos bajo un marco ágil, con ceremonias y artefactos compartidos, muchas veces el entendimiento del alcance no está completamente sincronizado entre los roles. Esto se debe, en gran medida, a una definición inicial ambigua del alcance por parte de los stakeholders del negocio” (Comentario del entrevistado, Chapter Lead de Calidad, 2025).

Es decir, los elementos formales de la agilidad no garantizan por sí solos una comprensión unificada; la ambigüedad en los entregables iniciales genera interpretaciones dispares, que luego se traducen en reprocesos o malentendidos funcionales:

“Hay avances importantes, pero aún enfrentamos desafíos en términos de entendimiento compartido. Muchas veces, el equipo de desarrollo empieza a construir funcionalidades antes de tener claros los criterios de aceptación, lo que implica ajustes posteriores y, en algunos casos, retrabajo” (Comentario del entrevistado, Scrum Master y Facilitador de DevOps, 2025).

La sincronización tardía de criterios se convierte en una fuente de ineficiencia operativa. Si los desarrolladores avanzan sin una guía validada por QA y producto, el riesgo de rehacer entregables se incrementa significativamente, afectando los tiempos de entrega:

“Existe un esfuerzo continuo por alinear las prioridades, pero aún se presentan inconsistencias, sobre todo cuando las definiciones de alcance no se actualizan de manera iterativa. En ocasiones, se asume que todos comprenden lo mismo por ‘alcance’, pero no siempre hay un documento fuente, como una historia bien refinada o un criterio de aceptación claro” (Comentario del entrevistado, Gerente de Producto, 2025).

Aquí se revela la raíz del problema: la falta de documentación clara y viva. A pesar de los esfuerzos por mantener la alineación, sin insumos tangibles y actualizados se favorece la interpretación subjetiva del alcance, debilitando la trazabilidad de lo que realmente debe cumplirse en el sprint:

“La alineación mejora cuando hay ceremonias como refinamiento bien estructuradas y Definition of Done claras. Sin embargo, todavía hay muchas historias que llegan a QA con criterios ambiguos o incompletos, lo cual genera interpretaciones distintas entre los equipos. Esto impacta directamente en la definición de los casos de prueba y en la cobertura” (Comentario del entrevistado, Analista Senior de QA Automatizado, 2025).

Aunque se reconoce que prácticas como el refinamiento conjunto y la Definition of Done son útiles, también se evidencia que no están siendo aplicadas de forma sistemática, la consecuencia directa es una mala cobertura en las pruebas y una ejecución ineficiente del plan de QA.

relevantes, como la experiencia o la eficiencia. Por otro lado, la nube de palabras refleja la prominencia de términos como “cuello de botella, integración, criterio, prueba y valor”, lo que permite inferir que las barreras también se materializan en la estructura técnica y en los procesos automatizados. En ese sentido, se indica que: “[...] hay una falta de integración efectiva entre los pipelines de entrega continua y los ciclos de validación, lo que genera cuellos de botella cuando las pruebas no están automatizadas ni adaptadas a los cambios rápidos del negocio” (Comentario del entrevistado, Scrum Master y Facilitador de DevOps, 2025).

Este tipo de brechas impacta no solo los tiempos de entrega, sino también la oportunidad con la que se detectan defectos, debilitando la capacidad preventiva del área de QA, desde otro punto de vista, se advierte que estas barreras no se circunscriben únicamente a deficiencias metodológicas, más bien:

“[...] las principales barreras no radican únicamente en la ejecución operativa del testing, sino en aspectos estructurales que afectan el flujo de valor. En PDT, todavía enfrentamos cuellos de botella causados por silos funcionales, una débil integración de QA desde las fases iniciales y una falta de visibilidad en los criterios de calidad” (Comentario del entrevistado, Agile Coach y Líder de Portafolio Estratégico, 2025).

En este contexto, las barreras son sistémicas, atraviesan múltiples niveles del ciclo de vida del software y reflejan una gobernanza débil de los criterios de calidad desde la conceptualización hasta la entrega final.

involucramiento de todos los roles —desde producto hasta infraestructura— bajo criterios comunes de calidad, como los OKRs vinculados a prácticas sostenibles.

En segundo lugar, se destaca la co-responsabilidad como una estrategia cultural efectiva: “[...] cuando los equipos de desarrollo y QA entienden que su trabajo impacta directamente en la satisfacción del cliente, hay un cambio radical en la cultura” (Comentario del entrevistado, Gerente de Producto, 2025).

Esta perspectiva se traduce en resultados tangibles al promover indicadores de calidad visibles como el porcentaje de regresiones o la cobertura de pruebas, facilitando así una rendición de cuentas más equitativa y alineada entre equipos. En tercer lugar, se advierte la necesidad de liderazgo visible y reconocimiento explícito del valor del QA. Según lo expresado: “fortalecer la cultura de calidad requiere una combinación de liderazgo visible, empoderamiento técnico y reconocimiento del valor del QA en la cadena de entrega” (Comentario del entrevistado, Chapter Lead de Calidad, 2025).

En este orden de ideas posiciona la calidad como un eje articulador del proceso de desarrollo, además promueve el empoderamiento profesional y el sentido de pertenencia dentro del equipo técnico. Un cuarto aspecto clave es la integración temprana del QA, incluso desde las etapas de diseño, lo cual se describe como una buena práctica orientada a anticipar problemas y fomentar la formación continua. Finalmente, se resalta la utilidad de los “checkpoints de calidad” en cada definición de listo (Definition of Done), así como la automatización de pruebas como expresión concreta de esta cultura, mostrando que una cultura de calidad eficaz no depende exclusivamente del discurso, sino de acciones prácticas insertas en los procesos clave.

Figura 19*Documentación y Trazabilidad*

Nota. La nube de palabras representa los principales elementos asociados a la documentación y trazabilidad en procesos de calidad de software. Atlas.Ti.25

En lo referente a la categoría documentación y trazabilidad, el análisis de las entrevistas revela una percepción compartida de que ambas prácticas, lejos de representar un apoyo estructurado para la calidad, se han convertido en una carga operativa para los equipos. Esto se evidencia cuando se afirma que: “[...] en este momento, la documentación representa más una carga que un apoyo” (Comentario del entrevistado, Chapter Lead de Calidad, 2025).

Lo cual obedece principalmente a la falta de centralización y automatización de los procesos de registro y seguimiento. Uno de los principales obstáculos identificados radica en la dispersión de los repositorios y la ausencia de herramientas integradas:

“Documentamos en múltiples repositorios: algunos en Confluence, otros en Excel, y muchos quedan en herramientas de gestión que no se conectan con los reportes técnicos. Esto fragmenta la trazabilidad. Desde DevOps, proponemos dashboards integrados con métricas en tiempo real, reportes automáticos desde herramientas de testing y documentación viva que se actualice con cada commit” (Comentario del entrevistado, Scrum Master y Facilitador de DevOps, 2025)

Esta fragmentación genera dificultades también para la capacidad de los equipos de auditar errores o validar funcionalidades, especialmente en proyectos con alta rotación o cambios frecuentes. Asimismo, la trazabilidad se describe como intermitente, y en muchos casos, insuficiente:

“Esto se vuelve un problema cuando hay que hacer seguimiento a defectos reportados por los usuarios o auditar los criterios con los que se validaron ciertas funcionalidades. Si no hay trazabilidad, no hay forma de saber si un requerimiento fue realmente probado o si el defecto se relaciona con un vacío en la validación” (Comentario del entrevistado, Gerente de Producto, 2025).

Tal diagnóstico sugiere una debilidad en los mecanismos de verificación, que se acentúa cuando no existe una guía de ejecución clara ni comentarios estructurados en los scripts de prueba. En contextos altamente regulados como los de PDT, esta problemática adquiere mayor gravedad: “Algunos equipos ven la documentación como una carga porque no está integrada al flujo de trabajo ni automatizada” (Comentario del entrevistado, Agile Coach y Líder de Portafolio Estratégico, 2025).

Esta afirmación resalta un problema sustancial: la transferencia de saber no documentado, especialmente en contextos donde la experiencia individual sustituye parcialmente la estandarización de procesos. La nube de palabras correspondiente refuerza esta percepción, destacando conceptos como *rotación*, *conocimiento*, *error*, *documentación*, *estabilidad*, *proceso* y *continuidad*, lo que evidencia que el impacto de la rotación no se limita a la dimensión humana o técnica, sino que atraviesa múltiples ejes organizativos.

En efecto, la entrevistada profundiza en este punto al señalar que:

“Cada vez que perdemos un recurso con experiencia, se rompen las cadenas de conocimiento dentro del flujo de valor. Esto se traduce en mayor tiempo para configurar entornos, redefinir pruebas automatizadas y entender las dependencias entre módulos.

Desde el punto de vista de la agilidad, también genera fricción en el equipo, porque implica rediseñar dinámicas de colaboración y reformular acuerdos internos”

(Comentario del entrevistado, Scrum Master y Facilitador de DevOps, 2025).

Lo cual genera cuellos de botella en etapas como la configuración de entornos, la ejecución automatizada y las pruebas de regresión. Por su parte, desde una visión transversal, se advierte que esta problemática compromete la capacidad operativa inmediata del equipo, como también la trazabilidad del aprendizaje acumulado; además, la pérdida de estos saberes tácitos y la ausencia de repositorios consolidados donde se centralice la experiencia previa derivan en un mayor tiempo de adaptación para nuevos integrantes y, por ende, en una menor eficiencia productiva. Otro elemento reiterado es el impacto sobre la automatización:

“Cada tester tiene un estilo de codificación, estructura de repositorios y convenciones propias. Cuando alguien se va, no siempre hay handover ni documentación actualizada de los scripts. Esto implica que quien llega debe hacer ingeniería inversa sobre lo ya

“automatización”, “criterio”, “refinamiento” y “cobertura” evidencian el uso de tecnologías y prácticas especializadas para optimizar procesos, garantizar trazabilidad y aumentar la eficiencia de los equipos técnicos. Atlas.Ti.25

El análisis de la categoría “Métodos y herramientas utilizadas” refleja una heterogeneidad en la adopción, uso y alineación estratégica de las herramientas de testing en PDT, evidenciada tanto en los comentarios textuales como en la nube de palabras y códigos asociados. Uno de los hallazgos más relevantes es que, a pesar de contar con herramientas potentes como Selenium, Jenkins o Cypress, su uso sigue siendo fragmentado y no estandarizado, como se señala en la afirmación:

“Tenemos herramientas potentes, pero su adopción no es homogénea. En algunos equipos utilizamos Jenkins, SonarQube, Selenium y herramientas de integración continua bastante maduras, pero en otros seguimos anclados a procesos manuales o herramientas obsoletas. Se necesita una estrategia corporativa de consolidación de herramientas con prácticas DevOps que fomenten el versionamiento de pruebas, pruebas en paralelo y validaciones automáticas por microservicio. También es clave invertir en entrenamiento del equipo para garantizar que se usen todas las funcionalidades disponibles, no solo las básicas.” (Comentario del entrevistado, Scrum Master y Facilitador de DevOps, 2025).

Esto sugiere que, si bien existen capacidades técnicas, no se ha logrado consolidar una estrategia corporativa que garantice prácticas mínimas comunes ni una cobertura transversal en los distintos equipos. Además, se identifican prácticas heredadas que no han sido actualizadas, como “procesos manuales o herramientas obsoletas”, lo que limita la eficiencia y genera retrabajos innecesarios, especialmente en validaciones y gestión de defectos.

La carencia de automatización en flujos críticos de pruebas también se menciona como una debilidad estructural, pues las pruebas se realizan aún de forma manual o sin criterios de entrada y salida claros, comprometiendo la trazabilidad de resultados y dificultando la replicabilidad. En este sentido, “no se aprovecha bien la integración entre herramientas” apuntando a la necesidad de sistemas interoperables que reduzcan los cuellos de botella y promuevan una trazabilidad continua desde el backlog hasta la validación final.

En efecto, esta desconexión entre artefactos, pipelines y tableros de control reduce la visibilidad y dificulta la toma de decisiones basada en evidencia. Por último, se destaca la urgencia de contar con “una batería mínima de pruebas automatizadas por sprint”, lo cual mejoraría tanto la cobertura y calidad de los entregables y por ende fortalecería la cultura de mejora continua, permitiendo focalizar esfuerzos en la validación de criterios de aceptación desde las primeras fases del desarrollo.

Figura 22

Planificación de Pruebas



Nota. La nube de palabras refleja los principales conceptos asociados a la planificación de pruebas, destacando términos como “planificación”, “prueba”, “criterio”, “refinamiento” y

“riesgo”, lo que evidencia la relevancia de una organización anticipada, colaborativa y orientada al valor en los procesos de calidad del software. Atlas.Ti.25

Se identifican dos ejes problemáticos principales: por un lado, la ausencia de planificación anticipada; por el otro, una ejecución fragmentada sin criterios claros ni momentos definidos. Tal como se indica en una de las citas más representativas:

“Planificar no es solo definir qué se va a probar, sino cuándo, con qué herramientas y en qué entorno. Cuando esto no ocurre, los testers entran sin claridad sobre qué validar, y eso nos obliga a hacer validaciones apuradas o incluso dejar pruebas críticas para ciclos posteriores” (Comentario del entrevistado, Scrum Master y Facilitador de DevOps, 2025).

Este tipo de omisión impacta en el valor mismo de las pruebas, pues genera ajustes tardíos y retrabajos innecesarios. Desde la perspectiva del rol de Producto, se señala que la planificación representa una inversión estratégica en la mitigación de riesgos, especialmente cuando se trata de productos complejos o funcionalidades críticas, donde las decisiones tardías pueden derivar en impactos significativos sobre los entregables.

Asimismo, se destaca que:

“Cuando no hay planificación en pruebas, las automatizaciones se desarrollan tarde, o se quedan sin ejecutar por falta de tiempo o ambientes. Esto genera deuda técnica y cobertura incompleta. Además, sin una estrategia de pruebas bien definida desde el Sprint 0, el testing se convierte en una carrera contrarreloj al final de cada iteración.” (Comentario del entrevistado, Analista Senior de QA Automatizado, 2025).

Lo que genera deuda técnica y disminuye la cobertura funcional. En este sentido, la ausencia de una estrategia de pruebas formal desde etapas tempranas —idealmente desde el Sprint 0—, representa no solo una debilidad operativa, sino también una pérdida de oportunidades para integrar calidad desde el diseño.

Finalmente, se reconoce que una planificación efectiva requiere una articulación con tableros de seguimiento, ceremonias ágiles y criterios de aceptación claros:

“Cuando no planificamos adecuadamente, lo que hacemos es desplazar la calidad hacia etapas tardías, donde ya no es posible prevenir, solo contener” (Comentario del entrevistado, Agile Coach y Líder de Portafolio Estratégico, 2025).

Sintetiza con precisión el valor preventivo de la planificación de pruebas en entornos ágiles y de alta demanda como el de PDT. Por tanto, los hallazgos sugieren que una mejora sustancial en esta categoría pasa por institucionalizar rutinas de planificación técnica desde fases tempranas, vinculadas a prácticas iterativas y colaborativas, y sustentadas en criterios técnicos compartidos.

Este ejercicio reveló patrones interdependientes que obstaculizan la eficiencia del área de testing, al tiempo que permitió identificar oportunidades viables de mejora mediante la integración de buenas prácticas, la estandarización de herramientas y el fortalecimiento de la documentación. Con base en estos hallazgos, se procedió a la elaboración de una matriz DOFA estratégica, que sintetiza las variables internas y externas más significativas y establece el punto de partida para la formulación de planes de acción sostenibles.

Tabla 18*Matriz DOFA*

Fortalezas	Debilidades
Alto nivel técnico de testers y automatizadores con dominio de herramientas clave (Selenium, Postman, etc.).	Fragmentación de procesos y roles por falta de alineación y tableros compartidos.
Cultura de corresponsabilidad en el aseguramiento de calidad emergente.	Baja cobertura de automatización en flujos críticos y regresiones recurrentes.
Disposición hacia la mejora continua y la adopción de estándares de calidad.	Alta rotación de personal que implica pérdida de conocimiento técnico y contextual.
Integración parcial de prácticas ágiles y sesiones de refinamiento conjunto.	Deficiencias metodológicas: falta de planificación de pruebas, métricas e indicadores formales.
Visión compartida sobre la necesidad de institucionalizar la trazabilidad.	Documentación disgregada, no automatizada, sin trazabilidad estructurada.
Oportunidades	Amenazas
Consolidación de una estrategia corporativa de herramientas interoperables y automatizadas.	Incremento en la complejidad de los proyectos sin adaptación de procesos QA.
Fortalecimiento del liderazgo técnico en prácticas de calidad.	Competencia externa con mayor madurez en frameworks y pipelines CI/CD.
Capacitación y formación avanzada en nuevas metodologías de prueba (PI Planning, OKRs técnicos).	Cambios frecuentes en los requerimientos del cliente sin planificación compartida.
Adopción de prácticas de calidad desde etapas tempranas del ciclo (Shift Left).	Pérdida de conocimiento acumulado por la alta rotación e inexistencia de lecciones aprendidas sistematizadas.

Nota. La tabla resume el análisis DOFA realizado para el área de calidad de software de PDT, identificando las principales fortalezas, debilidades, oportunidades y amenazas que afectan el desempeño de los procesos de aseguramiento de calidad. Construcción propia, 2025.

De este análisis surgen las estrategias FO, DO, FA y DA, las cuales articulan los componentes identificados con líneas de acción concretas que permitirán robustecer la cultura de calidad, consolidar herramientas interoperables, reducir la dependencia del conocimiento tácito y fortalecer la trazabilidad y planificación como ejes habilitadores de la mejora continua en PDT.

FO – Estrategias que aprovechan fortalezas para impulsar oportunidades: Crear comunidades técnicas internas orientadas a prácticas de calidad y automatización; Implementar sesiones colaborativas entre desarrollo, QA y producto para consensuar estándares de pruebas; Formalizar una política institucional de “automatización continua” con alcance progresivo.

DO – Estrategias que corrigen debilidades aprovechando oportunidades: Diseñar un plan de trazabilidad digital interconectado (Jira–Confluence–CI/CD); Desarrollar un plan de formación continua en planificación de pruebas y uso de métricas; Establecer procesos de integración temprana (Sprint 0) con lineamientos estandarizados.

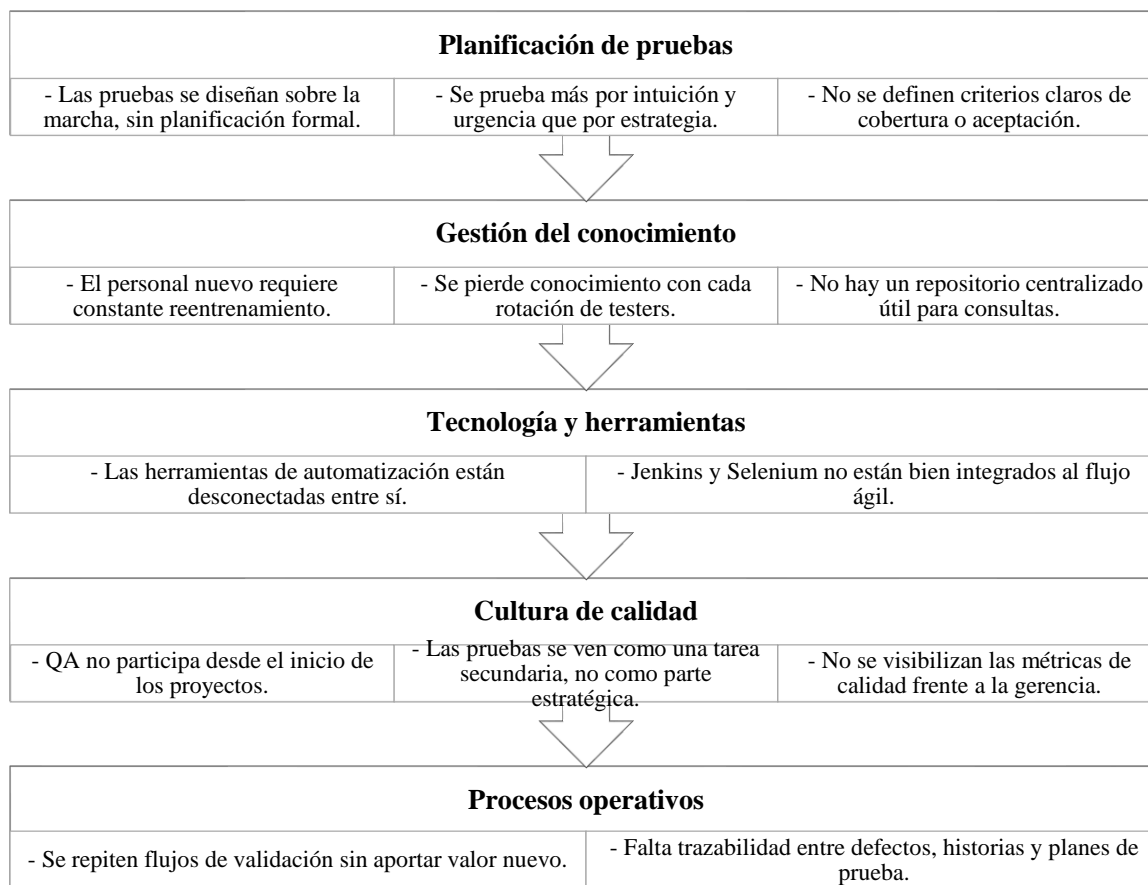
FA – Estrategias que utilizan fortalezas para mitigar amenazas: Potenciar la visibilidad del rol QA en espacios estratégicos para anticipar cambios de alcance o arquitectura; Escalar a nivel corporativo la cultura de corresponsabilidad para hacer frente a entornos cambiantes; Invertir en prácticas DevOps unificadas para garantizar adaptabilidad técnica ante la competencia.

DA – Estrategias que reducen debilidades y neutralizan amenazas: Sistematizar el aprendizaje colectivo mediante repositorios de lecciones aprendidas versionadas; Diseñar un sistema de rotación controlada del talento con transferencia estructurada del conocimiento; Reestructurar las fases de planificación con foco en el valor, la prioridad del negocio y la previsión de riesgos.

Procesos de Calidad Dentro del Ciclo de Vida del Desarrollo de Software

Tras identificar en la fase de diagnóstico múltiples deficiencias estructurales en el área de pruebas, se implementó una sesión de tormenta de ideas con el propósito de detectar problemas recurrentes, percepciones compartidas y oportunidades de mejora desde la experiencia directa de los actores involucrados. De acuerdo con las buenas prácticas señaladas por el Project Management Institute (PMI, 2021, p. 174), la tormenta de ideas se enmarca como una herramienta efectiva para la generación inicial de soluciones y el diagnóstico de causas cuando se busca integrar múltiples perspectivas.

Para este fin, se aplicaron diversas herramientas de la Guía del PMBOK® (2021), como los diagramas de Ishikawa y Pareto, y el diagrama de afinidad, las cuales permitieron organizar la información obtenida, priorizar causas críticas y visualizar las relaciones entre los factores técnicos, organizacionales y humanos que impactan la calidad del producto entregado. A continuación, se presentan los principales resultados de este análisis estructurado, acompañados por su respectiva interpretación.

Figura 23*Tormenta de Ideas*

Nota. La figura expone una cadena de debilidades identificadas en el proceso de aseguramiento de calidad del software.

Como parte del análisis de los procesos de calidad en el ciclo de vida del desarrollo de software, se elaboró un Diagrama de Afinidad con base en los hallazgos derivados de la tormenta de ideas; para agrupar conceptos dispersos en categorías temáticas y fue complementado con instrumentos como el diagrama de Ishikawa y el diagrama de Pareto, los cuales permitieron una evaluación integral de causas, efectos y prioridades críticas.

Figura 24

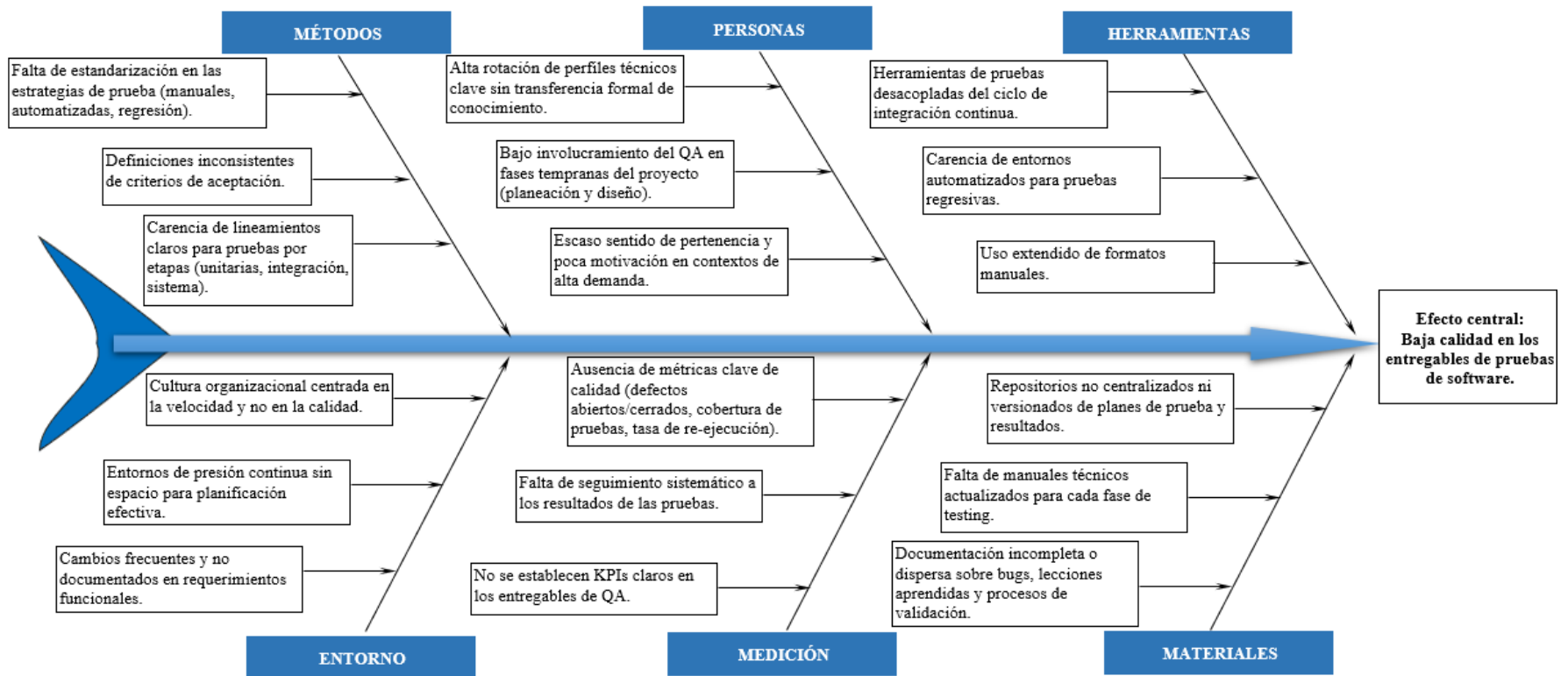
Diagrama de Afinidad



Nota. El diagrama resume las principales debilidades identificadas en el proceso de aseguramiento de calidad del software, categorizadas en cuatro áreas críticas: planificación de pruebas, gestión del conocimiento, tecnología y herramientas, y cultura de calidad.

Figura 25

Diagrama de Ishikawa



Nota. La figura presenta un diagrama de causa-efecto (Ishikawa) que identifica los factores críticos que contribuyen a la baja calidad en los entregables de pruebas de software.

El análisis del Diagrama de Ishikawa presenta fallas sistémicas que operan de forma simultánea tanto en el plano operativo como en el nivel estratégico, configurando un escenario de debilidad estructural dentro del ciclo de aseguramiento de calidad. Desde una perspectiva crítica, se advierte que las deficiencias metodológicas —en particular, la ausencia de estandarización en los procesos de prueba y la carencia de criterios homogéneos para la aceptación de entregables— constituyen un eje transversal que compromete la consistencia del sistema, generando resultados dispares, procesos poco repetibles y dificultades para el escalamiento técnico del área.

A esta situación se suma la alta rotación de personal técnico, que provoca desarticulación del conocimiento acumulado, afecta la curva de aprendizaje del equipo y dificulta la consolidación de una cultura organizacional orientada a la mejora continua; en consecuencia, la transferencia de buenas prácticas se ve limitada, y los tiempos de adaptación se incrementan, reduciendo la eficiencia general. En paralelo, la obsolescencia y fragmentación de las herramientas utilizadas restringe la capacidad del equipo para alcanzar niveles adecuados de automatización, integridad en la trazabilidad y control efectivo de versiones; como resultado, se generan reprocesos recurrentes, se extienden innecesariamente los tiempos de validación y se limita la visibilidad sobre el desempeño real del sistema bajo prueba.

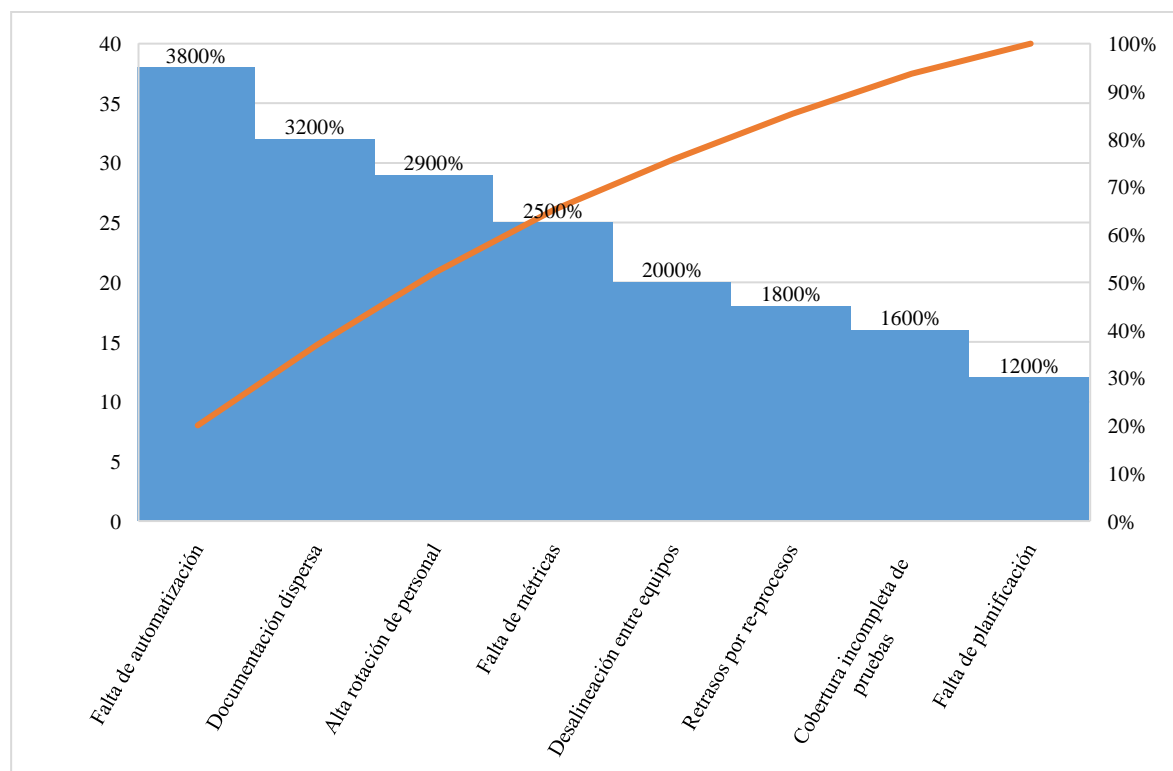
En el plano organizacional, la presión constante por cumplir cronogramas ajustados, combinada con una baja priorización de la calidad como valor estratégico, refuerza dinámicas reactivas que desplazan la planificación rigurosa por acciones correctivas de corto plazo, lo cual compromete la estabilidad técnica de los productos y genera una cultura centrada en la entrega más que en la excelencia. Asimismo, la ausencia de métricas objetivas y la falta de centralización documental obstaculizan la evaluación crítica de los resultados obtenidos; por tanto, se limita la

capacidad del sistema para retroalimentarse con base en datos verificables, impidiendo cerrar el ciclo de mejora con evidencia empírica que oriente la toma de decisiones futuras.

Para complementar el análisis de causas identificado en el Diagrama de Ishikawa, se empleó un Diagrama de Pareto con el propósito de jerarquizar los principales problemas que inciden en la baja calidad de los entregables del área de pruebas de software, en función de su frecuencia de aparición según los resultados obtenidos tanto en las encuestas como en las entrevistas realizadas. Este enfoque se fundamenta en el principio 80/20, que postula que una proporción reducida de causas suele concentrar la mayoría de los efectos.

Figura 26

Diagrama de Pareto



Nota. La figura muestra un diagrama de Pareto que permite identificar y priorizar las principales causas que impactan negativamente la calidad en los procesos de prueba de software.

Tabla 19*Análisis de Frecuencia de Problemas Críticos*

Problema	Q	%
Falta de automatización	38	20%
Documentación dispersa	32	17%
Alta rotación de personal	29	15%
Falta de métricas	25	13%
Desalineación entre equipos	20	11%
Retrasos por re-procesos	18	9%
Cobertura incompleta de pruebas	16	8%
Falta de planificación	12	6%

Nota. La tabla muestra la distribución porcentual de los principales problemas identificados en el proceso de calidad de software en PDT, donde la falta de automatización y la dispersión de la documentación se posicionan como las causas más relevantes de ineficiencia, seguidas por la alta rotación de personal y la falta de métricas de control.

En ese sentido, el diagrama anterior permitió identificar qué factores como la falta de automatización de pruebas, la documentación dispersa, la alta rotación del personal y la ausencia de métricas operativas acumulan más del 20% de las menciones, lo que evidencia su peso crítico en el deterioro del proceso de calidad.

Una vez identificadas y priorizadas las causas que afectan la calidad de los entregables en el área de pruebas de software mediante los diagramas de Ishikawa y Pareto, fue necesario complementar el análisis con una evaluación estructurada de los factores internos y externos que inciden en el desempeño del área. Para ello, se aplicaron las matrices MEFE (Evaluación de Factores Externos) y MEFI (Evaluación de Factores Internos), adaptadas a los hallazgos

específicos obtenidos en el diagnóstico. El objetivo de estas herramientas fue determinar la posición estratégica de la organización en relación con su entorno y sus capacidades internas.

Tabla 20

Matriz MEFE (Matriz de Evaluación de Factores Externos)

Factor Externo Clave de Éxito	Ponderación	Calificación	Puntuación
Oportunidades			
Entrenamiento de nuevas y mejores metodologías	0.30	4	1.20
Estabilidad del personal	0.15	4	0.60
Control del equipo	0.10	2	0.20
Certificaciones	0.10	1	0.10
Subtotal oportunidades			2.10
Amenazas			
Modificación de estándares	0.10	4	0.40
Mejores ingresos en empresas competidoras	0.06	4	0.24
Mejor clima laboral en otras organizaciones	0.10	1	0.10
Nuevos requerimientos sin planificación	0.06	4	0.24
Competidores fuertes en el sector tecnológico	0.03	2	0.06
Subtotal amenazas			1.04
Puntuación total	1.00		3.14

Nota. La tabla presenta los factores externos clave que afectan al área de testing de software en PDT, analizados mediante la Matriz de Evaluación de Factores Externos (MEFE).

Una puntuación total de 3,14 indica que la organización se encuentra en una posición favorable frente a su entorno externo, con capacidad de capitalizar oportunidades clave como el entrenamiento metodológico y la estabilización del personal. Sin embargo, se advierten amenazas latentes en relación con los cambios tecnológicos y la presión competitiva que requieren vigilancia constante.

Tabla 21*Matriz MEFI (Matriz de Evaluación de Factores Internos)*

Factor Interno Clave de Éxito	Ponderación	Calificación	Puntuación
Fortalezas			
Equipo altamente calificado	0.07	4	0.28
Colaboración en línea efectiva	0.09	4	0.36
Relaciones sólidas con otros departamentos	0.08	2	0.16
Compromiso de los integrantes del equipo	0.05	1	0.05
Subtotal fortalezas			0.85
Debilidades			
Alta rotación del personal	0.10	4	0.40
Procesos exigentes que desgastan al equipo	0.05	4	0.20
Procesos manuales sin automatización	0.06	1	0.06
Falta de documentación centralizada	0.20	4	0.80
Deficiencias metodológicas en el ciclo de pruebas	0.30	2	0.60
Subtotal debilidades			2.06
Puntuación total	1.00		2.91

Nota. La tabla presenta los factores externos clave que impactan el desempeño del área de testing de software en la empresa PDT, evaluados a través de la Matriz de Evaluación de Factores Externos (MEFE).

La puntuación total de 2,91 significa que, a pesar de contar con algunas fortalezas técnicas y colaborativas, el área de testing presenta debilidades significativas en procesos y metodologías, especialmente en lo referente a documentación y estandarización. Estas debilidades deben ser abordadas de manera prioritaria mediante estrategias internas de mejora continua.

Ambas matrices evidencian que el área de pruebas de software en la organización posee un entorno externo propicio para evolucionar, siempre y cuando logre fortalecer sus capacidades internas. Las principales alertas estratégicas se centran en la deficiente documentación, la rotación del personal y la falta de metodologías consolidadas, todas ellas problemáticas críticas ya destacadas en los diagramas anteriores. En consecuencia, los resultados de MEFÉ y MEFI permiten definir planes de acción alineados con la matriz DOFA previamente desarrollada, facilitando la selección de estrategias FO, DO, FA y DA orientadas a la mitigación de riesgos y el aprovechamiento de ventajas competitivas en entornos de calidad altamente regulados como PDT.

Estrategias de Optimización para los Procesos de Calidad

A partir del diagnóstico cualitativo y cuantitativo realizado en fases anteriores, se identificaron disrupciones significativas en la operación del área de aseguramiento de calidad (QA), las cuales no se limitan exclusivamente a factores técnicos (como la baja automatización o documentación fragmentada), sino que evidencian debilidades en la estructura sistémica del proceso. Se detectó una falta de madurez en componentes clave del ciclo de vida del software como la trazabilidad, la retención del conocimiento, la estandarización metodológica y la cultura organizacional hacia la calidad.

En respuesta a estos hallazgos, se llevaron a cabo sesiones de ideación bajo las metodologías SCAMPER y Design Thinking, priorizadas mediante una matriz de impacto-esfuerzo y contrastadas con modelos de madurez como TMMi y DevOps Research and Assessment (DORA). El resultado fue el diseño de cinco estrategias integradas y de alto impacto orientadas a lograr una transformación profunda, escalable y sostenible del área de QA.

Tabla 22*Estrategias*

Estrategia	Objetivo	Componentes Técnicos / Clave	Beneficios o Impactos Esperados
Ecosistema CI/CD Unificado y Gobernado	Optimizar la automatización de pruebas y despliegues, reduciendo reprocesos y cuellos de botella en entornos productivos.	Jenkins (orquestración de pipelines) Selenium (automatización funcional) SonarQube (calidad del código) Allure (reportería automatizada) GitHub Actions / GitLab CI (versionamiento y ejecución)	Reducción del tiempo de pruebas regresivas hasta en un 45%. Mejora en frecuencia de releases. Trazabilidad automatizada desde el commit hasta el release.
Plan de Onboarding de Calidad	Minimizar el impacto negativo de la rotación mediante el fortalecimiento del aprendizaje técnico y contextual de nuevos miembros.	Rutas de formación por perfiles (manual, automatizado, QA lead) Repositorio de buenas prácticas (Confluence) Mentores por célula Simulaciones con datos reales	Reducción del tiempo de estabilización de 4 a 2 semanas. Incremento del 60% en retención del conocimiento técnico.
Marco Estratégico de Pruebas Ágiles	Establecer acuerdos inter-roles que alineen la calidad desde la definición hasta la entrega continua.	Alineación en criterios de aceptación desde refinamiento Dashboards compartidos QA presente en PI Planning y demos	Reducción del 40% en ambigüedad de requisitos. Mejora del 55% en cobertura de pruebas documentadas.

Estrategia	Objetivo	Componentes Técnicos / Clave	Beneficios o Impactos Esperados
		Historias con atributos mínimos de calidad	
Automatización de la Trazabilidad QA	Construir una línea de tiempo digital y dinámica que documente los procesos de QA para auditorías y continuidad operativa.	Scriptado semántico con Jira Repositorios automatizados (Zephyr / Xray) Tableros Power BI	Trazabilidad funcional $\geq 90\%$. Auditorías de defectos con duración ≤ 2 días.
2.5. Programa de Gobernanza de Calidad	Establecer un marco de gobierno organizacional que promueva la calidad como eje transversal del valor entregado.	Indicadores de madurez por célula (TMMi adaptado) Auditorías internas trimestrales OKRs/KPIs con enfoque en calidad Comité de calidad interfuncional	Alineación estratégica entre roles técnicos y funcionales. Detección temprana de riesgos desde Sprint 0.

Nota. La tabla sintetiza las estrategias de optimización diseñadas para fortalecer los procesos de aseguramiento de calidad en PDT, especificando los objetivos, componentes técnicos clave y beneficios esperados de su implementación.

Las estrategias formuladas a partir del análisis de procesos y causas se integran de manera coherente en la matriz de impacto, herramienta que permite visualizar su contribución diferenciada con base en dos variables clave: el alcance organizacional y el grado de transformación generado; este cruce analítico posibilita establecer una clasificación estratégica

que facilita, por una parte, la identificación de aquellas intervenciones con impacto estructural elevado —como el Ecosistema CI/CD Unificado y el Programa de Gobernanza de Calidad— y, por otra parte, la priorización de soluciones orientadas a resolver problemáticas críticas de forma inmediata, como el Plan de Onboarding Técnico o los esquemas de Automatización de la Trazabilidad, los cuales responden a cuellos de botella específicos detectados durante el análisis de causa raíz.

Figura 27

Matriz de Impacto - Esfuerzo de las Estrategias

Estrategia	Impacto Esperado	Esfuerzo	Prioridad
Ecosistema CI/CD unificado	Muy alto	Alto	Alta
Plan de Onboarding de Calidad	Alto	Medio	Alta
Marco Estratégico de Pruebas Ágiles	Muy alto	Medio	Alta
Automatización de la trazabilidad	Alto	Medio	Alta
Programa de Gobernanza de Calidad	Alto	Media	Alta

Nota. La figura presenta la priorización estratégica de cinco iniciativas clave orientadas a mejorar la calidad del software. Todas las estrategias evaluadas presentan una prioridad alta, destacando especialmente el “Ecosistema CI/CD unificado” y el “Marco Estratégico de Pruebas Ágiles” por su impacto muy alto, pese al esfuerzo requerido.

Para priorizar estratégicamente la implementación de las estrategias propuestas, se elaboró un Mapa de Impacto de Alternativas que relaciona el nivel de impacto organizacional con el esfuerzo requerido.

Tabla 23*Matriz de Impacto vs. Esfuerzo*

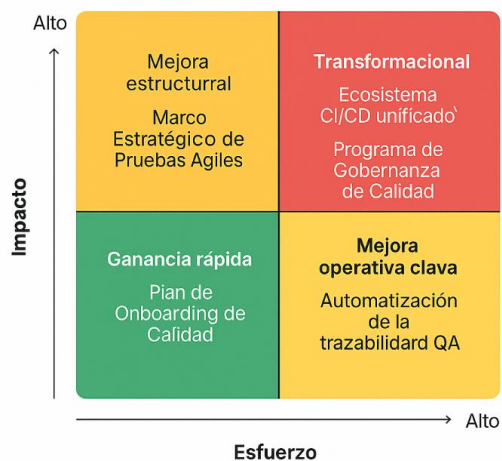
Estrategia	Clasificación	Justificación
Ecosistema CI/CD unificado	Transformacional	Aunque requiere inversión inicial, su impacto técnico-operativo es transversal y a largo plazo reduce hasta 45% de los reprocesos.
Plan de Onboarding de Calidad	Ganancia rápida	Tiene un impacto inmediato en la curva de aprendizaje y retención del conocimiento con bajo esfuerzo.
Marco Estratégico de Pruebas Ágiles	Mejora estructural	Fortalece la calidad desde etapas tempranas de planeación, mejora la comunicación y previene defectos.
Automatización de la trazabilidad QA	Mejora operativa clave	Permite continuidad del conocimiento, auditoría ágil y transparencia. Es fundamental para entornos regulados.
Programa de Gobernanza de Calidad	Transformacional	Su alcance es organizacional e involucra cultura, métricas y liderazgo; impacta estratégicamente, pero requiere continuidad y sostenibilidad.

Nota. La tabla clasifica las estrategias propuestas de acuerdo con su naturaleza de impacto (transformacional, estructural, operativa o de ganancia rápida), justificando su priorización estratégica con base en criterios de viabilidad, profundidad de cambio organizacional e impacto en los procesos de calidad de software.

Esta matriz facilita la toma de decisiones al clasificar las iniciativas según su viabilidad y valor, permitiendo identificar rápidamente aquellas que representan ganancias rápidas, mejoras estructurales u oportunidades transformacionales.

Figura 28

Mapa de Impacto vs. Esfuerzo de las Alternativas



Nota. La figura representa una matriz de impacto vs. esfuerzo para clasificar las estrategias de mejora en calidad de software. Se identifican cuatro cuadrantes: iniciativas transformacionales, operativas clave, estructurales y de ganancia rápida, lo cual facilita la priorización de acciones según su retorno y complejidad de implementación.

A partir del análisis estratégico, se diseñó un Roadmap de Implementación distribuido por trimestres, con el fin de organizar la ejecución secuencial y sostenible de cada iniciativa. Este plan temporal contempla responsables, objetivos específicos y metas de impacto medibles, garantizando una hoja de ruta coherente que facilite el seguimiento y la evaluación continua de los avances.

Tabla 24*Roadmap de Implementación por Trimestres*

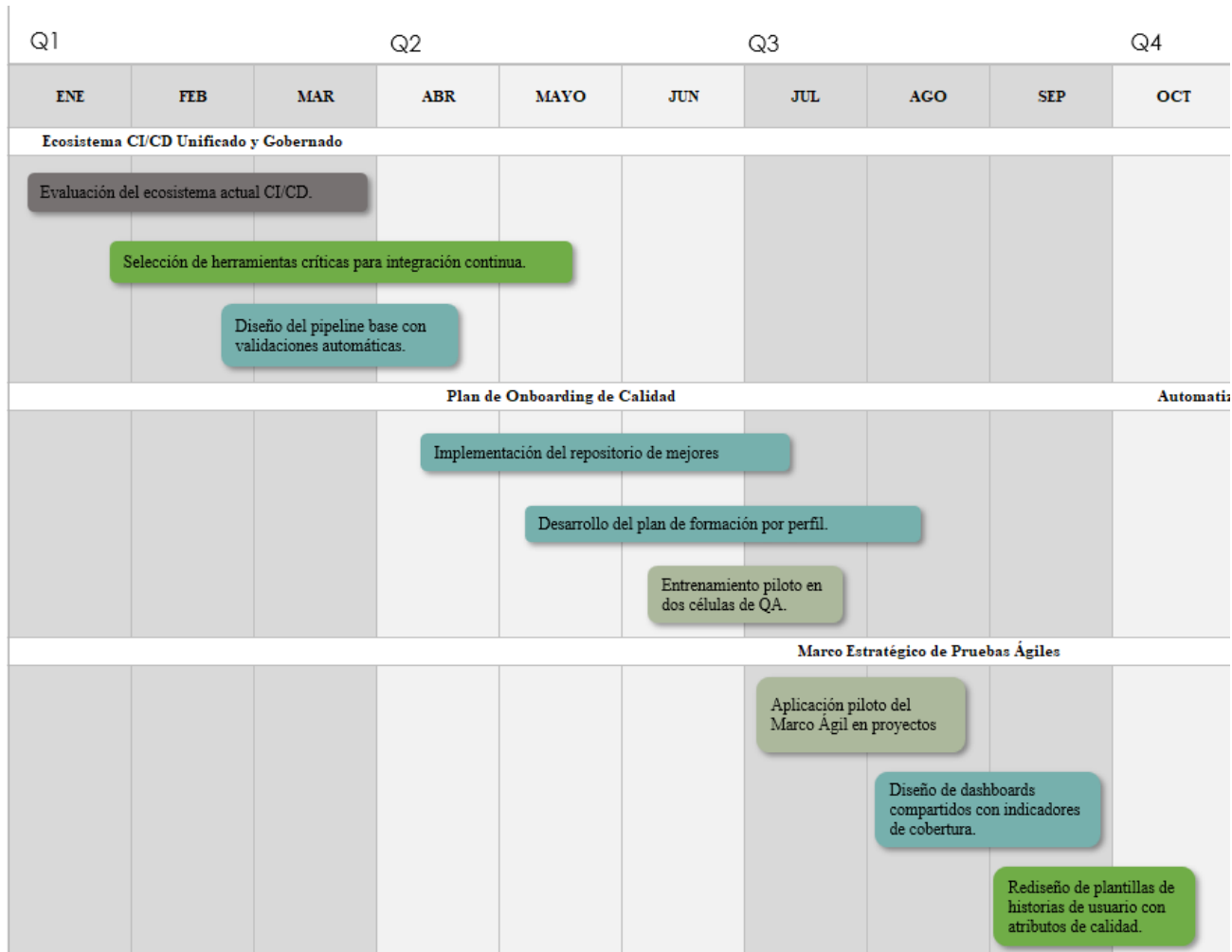
Trimestre	Actividades Principales	Indicadores Esperados	Estrategia Asociada
Q1 (Enero - Marzo)	Evaluación del ecosistema actual	Tiempos de ejecución	Ecosistema
	CI/CD	↓ 15%	CI/CD Unificado
	Selección de herramientas críticas para integración continua	Errores en despliegues ↓ 20%	y Gobernado
Q2 (Abril - Junio)	Diseño del pipeline base con validaciones automáticas	Estabilidad en ambientes ↑ 10%	
	Desarrollo del plan de formación por perfil	Tiempo de estabilización de nuevo personal ↓ 50%	Plan de Onboarding de Calidad
	Implementación del repositorio de mejores prácticas	Participación en simulaciones ≥ 80%	
Q3 (Julio - Septiembre)	Entrenamiento piloto en dos células de QA	Uso del repositorio ↑ 60%	
	Aplicación piloto del Marco Ágil en proyectos prioritarios	Cobertura de pruebas ↑ 30%	Marco Estratégico de Pruebas Ágiles
	Diseño de dashboards compartidos con indicadores de cobertura	Ambigüedad de historias ↓ 40%	
Q4 (Octubre - Diciembre)	Rediseño de plantillas de historias de usuario con atributos de calidad	Participación del QA en PI Planning ≥ 90%	
	Desarrollo de scripts de trazabilidad QA (Jira + Zephyr/Xray)	Trazabilidad documentada ≥ 90%	Automatización de la Trazabilidad QA
	Integración con visualización dinámica en Power BI	Tiempo promedio de auditoría ↓ 60%	
	Validación de cobertura funcional automatizada	Alertas tempranas de errores ↑ 25%	

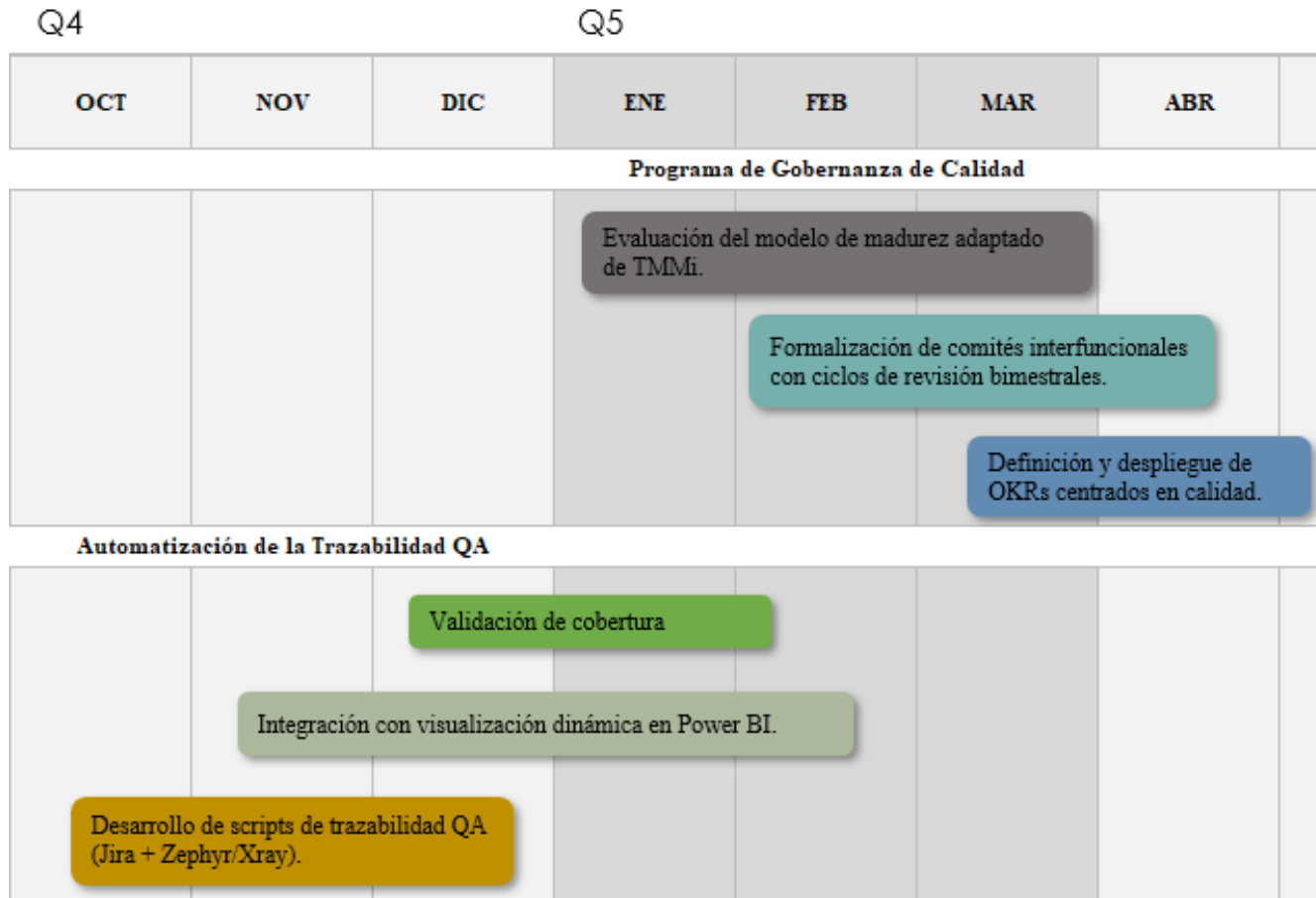
Trimestre	Actividades Principales	Indicadores Esperados	Estrategia Asociada
Q5 (Enero - Marzo siguiente año)	<p>Evaluación del modelo de madurez adaptado de TMMi</p> <p>Definición y despliegue de OKRs centrados en calidad</p> <p>Formalización de comités interfuncionales con ciclos de revisión bimestrales</p>	<p>Inclusión de calidad en 100% de OKRs del área</p> <p>Auditorías técnicas realizadas (≥ 2)</p> <p>Satisfacción de stakeholders con entregables de calidad</p> <p>$\uparrow 20\%$</p>	Programa de Gobernanza de Calidad

Nota. La tabla presenta el roadmap de implementación estratégico para la optimización de los procesos de calidad en PDT, estructurado en cinco trimestres consecutivos. Cada fase prioriza actividades específicas, indicadores de desempeño cuantificables y su correspondiente alineación con las estrategias clave definidas.

Figura 29

Roadmap de Implementación





Nota. La figura presenta las estrategias priorizadas en el plan de mejora para la calidad del software, distribuido por trimestres (Q1–Q5) y organizado en cinco ejes de acción: CI/CD, Onboarding, Pruebas Ágiles, Trazabilidad y Gobernanza. Se detallan actividades específicas, responsables y duración estimada para cada componente estratégico.

Es relevante subrayar que el diseño estratégico propuesto articula de forma coherente un conjunto de iniciativas de alto impacto, orientadas específicamente hacia la transformación organizacional del área de testing; en este sentido, las estrategias formuladas —evaluadas con base en criterios de impacto, viabilidad operativa y retorno de inversión— configuran una hoja de ruta progresiva y sostenible, cuya estructuración se encuentra alineada tanto con los marcos de referencia establecidos por la Guía del PMBOK® como con las necesidades reales y actuales de la empresa PDT.

A partir de dicha articulación, se consolida un plan de mejora integral que incorpora dimensiones técnicas, metodológicas y culturales, lo que permite garantizar no únicamente la optimización operativa de los procesos de pruebas, sino también el fortalecimiento de una cultura de calidad transversal, entendida como un principio organizacional que permea los distintos niveles de decisión y ejecución. En consecuencia, se establecen las condiciones necesarias para la construcción de un modelo de aseguramiento de calidad robusto, proactivo y adaptable, capaz de responder con eficacia a los desafíos propios de proyectos de alta complejidad tecnológica y de acelerar, de manera sostenida, la entrega de valor en entornos caracterizados por su alta exigencia, variabilidad e innovación constante.

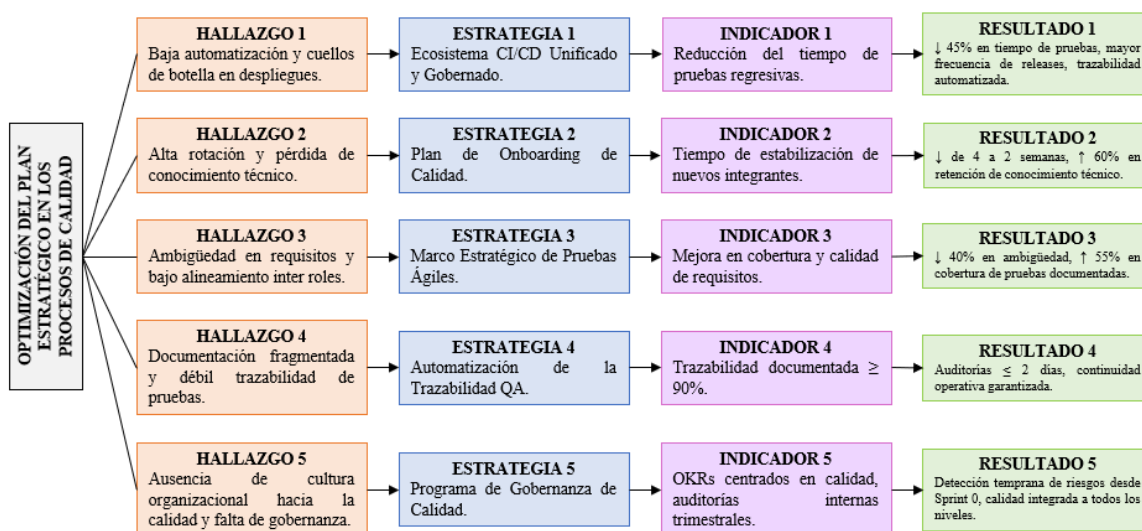
Evidencia de Optimización

En línea con el objetivo específico propuesto, puede afirmarse que el plan estratégico fue optimizado a partir del diseño e implementación progresiva de cinco estrategias que responden a un diagnóstico detallado de los procesos del área de calidad en software. Estas estrategias surgieron como respuesta directa a problemáticas estructurales previamente identificadas y se formularon empleando enfoques colaborativos como SCAMPER, Design Thinking y la matriz impacto-esfuerzo, lo que permitió priorizar acciones viables y bien fundamentadas en términos

técnicos, organizacionales y culturales. El desarrollo de un roadmap con indicadores trimestrales sirvió como guía para llevar estas acciones a la práctica, permitiendo pasar de un modelo operativo desarticulado a un sistema mucho más ordenado, medible y alineado con los principios de la Guía del PMBOK®. Este avance no solo mejora aspectos como la automatización, la trazabilidad o la formación técnica, también fortalece la colaboración entre roles, el gobierno de la calidad y, sobre todo, fomenta una cultura institucional comprometida con la mejora continua y con la entrega de valor sostenido en el tiempo.

Figura 30

Optimización del Plan Estratégico



Nota. La figura anterior sistematiza la forma en que los hallazgos del diagnóstico inicial fueron transformados en estrategias optimizadoras del plan estratégico, evidenciando mejoras tangibles mediante indicadores medibles, lo cual refleja una alineación clara con los principios del PMBOK® y las buenas prácticas de gestión de calidad en software.

Socialización de Resultados con los Involucrados

Con el objetivo de socializar los hallazgos del proyecto y validar su aplicabilidad, se organizaron diferentes espacios de divulgación y evaluación con los principales actores del ecosistema tecnológico de la empresa PDT, incluyendo líderes técnicos, analistas de QA, representantes de producto, DevOps, equipo de gestión de talento, y miembros de la PMO. La presentación de resultados se realizó mediante ponencias formales en dos eventos clave: una conferencia virtual dirigida por ACIEM (Asociación Colombiana de Ingenieros), con enfoque en herramientas de la Guía del PMBOK® para la calidad, y una participación institucional en la décima edición de la Expotech 2025, en el marco del Encuentro Nacional de Semilleros de Investigación ACOFI.

Estas presentaciones permitieron contextualizar las estrategias propuestas dentro del entorno operativo de PDT, evidenciando los beneficios esperados en términos de automatización, trazabilidad, cultura de calidad y eficiencia operativa. Las sesiones estuvieron acompañadas de afiches tipo póster, materiales de apoyo visual y un resumen ejecutivo entregado a los asistentes.

Como parte del enfoque cualitativo adoptado para garantizar la pertinencia de las soluciones planteadas, se recurrió al juicio de expertos mediante entrevistas semiestructuradas y retroalimentación especializada. Se contó con la participación de profesionales externos con experiencia en ingeniería de calidad, DevOps, automatización de pruebas y gestión estratégica de proyectos, quienes analizaron tanto los entregables técnicos como las estrategias de mejora propuestas. Este proceso se desarrolló en tres rondas:

Primera ronda: Evaluación individual de las estrategias por parte de los expertos a partir de un formulario de valoración con criterios de pertinencia, viabilidad técnica, impacto esperado y alineación con tendencias de la industria.

Segunda ronda: Taller colaborativo de retroalimentación cruzada mediante la metodología Delphi, que permitió ajustar elementos críticos como la secuencia de implementación, dependencias entre estrategias y capacidades organizacionales.

Tercera ronda: Revisión de los indicadores de éxito propuestos, sugiriendo mejoras en la forma de medir el retorno sobre la inversión (ROI), la madurez del ecosistema CI/CD, y la evolución de la trazabilidad QA.

Durante las sesiones de retrospectiva con los equipos internos de PDT, se aplicaron técnicas de facilitación para recoger percepciones, alertas tempranas, y oportunidades de mejora. La metodología empleada combinó: A) Dinámicas tipo “Start-Stop-Continue” para categorizar los procesos a ajustar, eliminar o mantener; B) Mapas de empatía para entender las emociones y desafíos vividos por los actores del proceso; C) Rondas abiertas de reflexión estructurada con los equipos técnicos y funcionales.

A partir de estas sesiones se generaron las siguientes lecciones aprendidas:

La falta de estandarización en las herramientas y prácticas de testing impide la consolidación de una cultura de calidad.

La rotación de personal sin mecanismos de transferencia de conocimiento produce interrupciones críticas en los flujos de valor.

La documentación, cuando no está automatizada ni alineada al ciclo de vida del software, representa más una carga que un activo.

La planificación de pruebas y su integración en el PI Planning sigue siendo una deuda pendiente en la mayoría de las células de desarrollo.

El modelo de madurez del testing necesita estar articulado con los OKRs de los equipos técnicos para asegurar su evolución continua.

Como parte del proceso de socialización de resultados, y en concordancia con los objetivos de visibilización científica y transferencia de conocimiento, se generaron varios productos que documentan la participación activa del proyecto en escenarios académicos y profesionales de alto impacto. Se incluye el diseño del póster presentado en la Expotech 2025, titulado “Herramientas para mejorar el plan estratégico de calidad”.

Figura 31

Poster - Conferencia Empresarial

CONFERENCIA EMPRESARIAL

HERRAMIENTAS PARA MEJORAR EL PLAN ESTRATÉGICO DE CALIDAD

TEMAS

- Calidad de software y aplicación de herramientas PMBok.
- Problemática de la empresa PDT y resultados en la optimización.

SINDY VANESSA VERGEL

- Ingeniera en Telecomunicaciones y candidata a magister en gerencia de proyectos, Universidad Nacional Abierta y a Distancia (UNAD).
- Especialización Tecnológica. SENA.

MILTON ALFREDO GÓMEZ

- Ingeniero Industrial, Universidad Central.
- Candidato a Especialista y magister gerencia de proyectos, Universidad Nacional Abierta y a Distancia (UNAD).

CONFERENCISTAS

19 SEPTIEMBRE | **3:00 P.M. VIRTUAL** | **¡INSCRÍBETE SIN COSTO!** | **ACIEM** Asociación Colombiana de Ingenieros

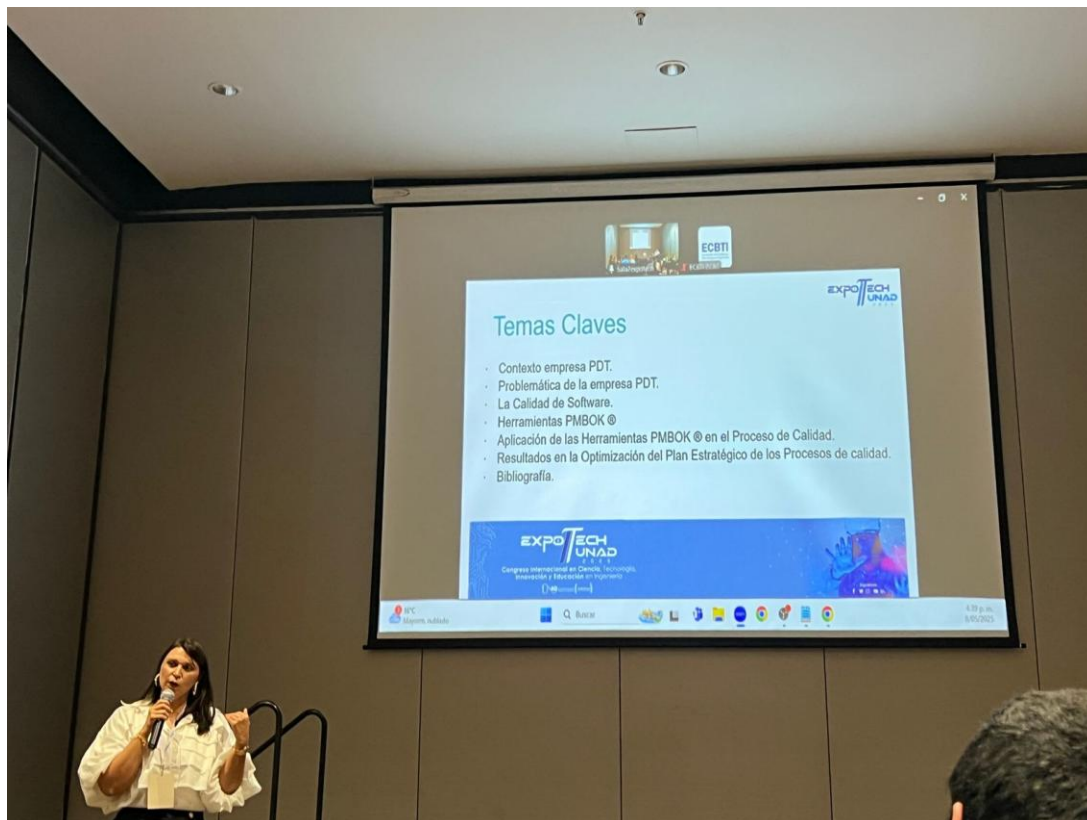
Nota. Este póster fue diseñado bajo lineamientos gráficos institucionales y expone, de manera visual y sintética, los hallazgos del diagnóstico, las estrategias desarrolladas y los beneficios esperados.

Se anexa la certificación oficial de la participación como conferencista, otorgada por la organización del evento (Anexo D). Además, el proyecto fue seleccionado para su divulgación en la décima edición del evento nacional EXPOTECH – UNAD, dentro del Eje 1: Productividad

y Sostenibilidad Territorial desde la Innovación, en la línea de “Desarrollo de Software”; bajo este link de acceso: <https://eventos.unad.edu.co/eventos/2025/expotech-2025/index.html>.

Figura 32

Conferencia - Expotech 2025



Nota. La imagen muestra la presentación de los temas clave desarrollados durante la ponencia de ExpoTech UNAD, en el Congreso Internacional de Ciencia, Tecnología e Innovación.

Este cartel presenta la información oficial del proyecto desarrollado en el marco de EXPOTECH 2025, evento académico organizado por la UNAD que visibiliza investigaciones de alto impacto en ingeniería y tecnología.

Figura 33

Información Oficial de EXPOCH 2025

MEJORAMIENTO DEL PLAN ESTRATÉGICO DE LOS PROCESOS DE CALIDAD EN EL CICLO DE VIDA DEL DESARROLLO DE SOFTWARE EN EL ÁREA DE PRUEBAS Y CERTIFICACIÓN PARA LA EMPRESA PDT: POR MEDIO DE LAS HERRAMIENTAS DE LA GUÍA DEL PMBOK®

AUTORES:
 SINDY VANESSA VERDEZ LAFAURE, sverdegel@unadvirtual.edu.co
 Grupo de Investigación SIGCENY Similitero Projects Decada UNAD, EC211
 MILTON ALFREDO DOMÍZ, migomez@unadvirtual.edu.co
 Grupo de Investigación SIGCENY Similitero Projects Decada UNAD, EC211

RESUMEN:
 La investigación titulada "Mejoramiento del plan estratégico de los procesos de calidad en el ciclo de vida del desarrollo de software" aborda el posible alcanzar la excelencia operativa en el desarrollo de software en una estrategia rigurosa de gestión de calidad. El estudio se aplicó al área de pruebas y certificación en una empresa colombiana de tecnología financiera. Se analizaron factores críticos como deficiencias en los procesos de testing, alta incidencia de defectos en entornos productivos, insatisfacción en la descentralización del conocimiento y recursos elevados asociados operativos por pruebas manuales redundantes.
 Los entendimientos contextualizan la evaluación de los modelos de calidad en la industria, identificando tendencias en la gestión de pruebas y certificación. Se analizaron metodologías como CHAI, Six Sigma, Agile y DevOps para evaluar su aplicabilidad en la optimización de los procesos de testing.
 En relación a los objetivos, planteamos los 3 más relevantes:
 1. Diagnosticar las deficiencias operacionales del área de pruebas de software.
 2. Analizar los procesos de calidad dentro del ciclo de vida del desarrollo de software.
 3. Diseñar estrategias de optimización para los procesos de calidad en el área de pruebas y certificación.
MATERIALES Y MÉTODOS:
 Este estudio utiliza un enfoque de investigación mixto, integrando técnicas cuantitativas y cualitativas para una comprensión holística de los procesos de calidad en el ciclo de vida del desarrollo de software, con énfasis en pruebas y certificación. El diseño metodológico es no experimental, descriptivo y exploratorio, analizando fenómenos en su contexto natural sin manipular variables independientes.
 El mejoramiento de calidad se implementó mediante múltiples niveles de validación técnica, garantizando la integridad funcional, estructural y operativa del producto final. Las metodologías utilizadas y su contribución a la mejora del producto fueron:
 1. **Pruebas unitarias:** Verificaron la cohesión y precisión algorítmica de los componentes individuales, detectando anomalías en módulos discretos desde fases tempranas del ciclo de vida (Ponies, 2024).
 2. **Pruebas de integración:** Validaron la funcionalidad de las interfaces internas entre módulos, asegurando la interoperabilidad y mitigando defectos por dependencias estructurales y comportamientos inesperados en las interfaces (Ponies, 2024; Pastori et al., 2021).
 3. **Pruebas de registro y estrés:** Mantuvieron la estabilidad funcional y el desempeño (optino bajo escenarios críticos, como actualizaciones del código base o condiciones extremas de carga transaccional y uso concurrenente (Lim et al., 2022).
 4. **Pruebas de aceptación:** Garantizaron la conformidad del producto con los requisitos funcionales, técnicos y normativos establecidos para la entrega final, mediante criterios definidos por especialistas técnicos y usuarios expertos (Mustafa et al., 2021; Lim et al., 2022).

ANÁLISIS Y RESULTADOS:
ANÁLISIS
 El análisis de datos se estructuró según la Sección 4.5 de la Guía del PMBOK®, que especifica métodos para recopilar y evaluar datos para una comprensión profunda de una situación (PM, 2021). Se utilizó una muestra no probabilística de 24 empleados seleccionados por su conocimiento y experiencia en los procesos analizados.
 El estudio demostró que es posible optimizar el plan estratégico de calidad en el ciclo de vida del desarrollo de software en la empresa PDT mediante herramientas de PMBOK®. Las sus estrategias mejoraron a necesidades concretas y se contrastaron con estándares legales y análisis comparativos, creando un modelo flexible y escalable. Se abordaron temas prácticos como la aplicación imparcial de pruebas, la ausencia de planes formalizados, la implementación de tipos de datos y la descentralización entre equipos técnicos y funcionales, impactando costos, tiempos de entrega y percepción de clientes.
RESULTADOS
 La aplicación de las herramientas de la Guía del PMBOK® en el área de pruebas y certificación de la empresa PDT generó los siguientes resultados:
 El diagnóstico de deficiencias y la optimización de estrategias mejoraron la calidad del software entregado, con menor defectos y una cobertura de pruebas más robusta. La identificación de causas de fondo y la automatización de pruebas incrementó la eficiencia operativa, reduciendo tiempos y costos. Las herramientas del PMBOK® proporcionaron un marco estructurado para decisiones basadas en datos, facilitando la identificación de riesgos y su mitigación. Además, mejoraron la comunicación entre equipos de desarrollo y pruebas, creando una cultura de trabajo más colaborativa y eficiente.
CONCLUSIONES:
 La aplicación de las herramientas PMBOK® en el área de pruebas y certificación de software ha tenido un impacto positivo significativo en la optimización de los procesos de calidad. La combinación de técnicas estructuradas para el diagnóstico, análisis de procesos y diseño de estrategias de mejora ha permitido a la organización identificar deficiencias operacionales, optimizar sus procesos y tomar decisiones más informadas. A través de esta metodología, se ha logrado no solo una mayor calidad en el software entregado, sino también una mayor eficiencia operativa, mejor comunicación entre los equipos y una cultura organizacional más alineada con la mejora continua.

BIBLIOGRAFÍA
 Ponies, S. (2024). Fundamentals of software testing (2nd ed., Revised and Updated). ISTE Ltd. & John Wiley & Sons. ISBN 978-1-7852-592-2.
 International Software Testing Qualifications Board (ISTQB). (2024). Welcome to ISTQB®. <https://www.istqb.org/>
 Pastori, M., Sarano, M., Pérez-Castillo, R., Petersen, G., & Heria, J. L. (2021). Toward a quantum software engineering. IT Professional, 23(1), 55.
 Materials & Continua, 6(8/2), 1542-1555. <https://doi.org/10.32664/mtc.2021.014391>
 Project Management Institute. (2021). El estándar para la dirección de proyectos y Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®) (7ª ed.). Project Management Institute. <https://www.pmi.org/>

VÍDEO DE SOCIALIZACIÓN
 Socialización Proyecto Grupo UNAD
 Dirección de las Herramientas PMBOK® en el Área de Pruebas y Certificación de la Empresa PDT
 Este video muestra la socialización del proyecto y los resultados obtenidos. Incluye una presentación de diapositivas y una sesión de preguntas y respuestas con los participantes.

Nota. La imagen corresponde al póster científico presentado en ExpoTech UNAD (2025).

Se documenta la reunión de cierre realizada con los equipos internos de la empresa PDT, donde se presentaron los entregables finales del proyecto, se recibieron comentarios por parte de líderes funcionales, y se establecieron acuerdos para la posible adopción progresiva de las estrategias definidas. Esta acta registra tanto los participantes como las conclusiones y compromisos derivados.

Tabla 25*Resumen de Acta de Socialización de Resultados*

Acta de socialización interna de resultados
Empresa: PDT – Pruebas y Desarrollo Tecnológico Proyecto: Mejoramiento del plan estratégico de los procesos de calidad en el ciclo de vida del desarrollo de software Fecha: _____ Lugar: Sala de conferencias PDT – Bogotá / conexión remota (Teams) Hora de inicio: 10:00 a. m. Hora de finalización: 12:15 p. m. Modalidad: Híbrida Elaborada por: Coordinación de Calidad e Innovación
Objetivo de la reunión Socializar los resultados, hallazgos y estrategias derivadas del proyecto de mejoramiento del plan estratégico de calidad en el área de pruebas, validar los entregables con los actores clave y definir los pasos a seguir para su adopción progresiva dentro del ciclo de vida del desarrollo de software.
Comentarios relevantes de los participantes El enfoque CI/CD unificado es una deuda histórica; su implementación nos ayudará a romper cuellos de botella entre desarrollo y QA” — Arquitecto de Software “El onboarding estructurado es vital; hoy estamos perdiendo hasta un mes en adaptaciones cuando un nuevo tester entra” — QA Automatizador Senior “Me interesa mucho que integremos calidad a nuestros OKRs desde ya. Podemos hacer un piloto en mi célula el próximo trimestre” — Product Manager “El roadmap tiene sentido. Requiere apoyo gerencial para asegurar tiempo y recursos. Lo podemos gestionar por fases” — Coordinadora de Gestión del Cambio
Temas abordados

Acta de socialización interna de resultados

Presentación de los resultados clave del diagnóstico (Fase 1): problemas críticos, causas estructurales y patrones de deficiencia recurrentes.

Revisión de las estrategias diseñadas (Fase 3) para cada eje temático: automatización, onboarding, trazabilidad, marco ágil y gobernanza.

Evaluación de impactos esperados y ROI estimado.

Socialización del roadmap de implementación por trimestres.

Espacio de retroalimentación abierta entre equipos.

Conclusiones

La socialización fue bien recibida por los equipos y permitió validar la pertinencia de las estrategias diseñadas. Se identificó un alto nivel de compromiso por parte de los líderes técnicos y funcionales, quienes valoraron la propuesta como una hoja de ruta clara para profesionalizar y optimizar el área de calidad de software. La reunión permitió consolidar acuerdos claves y sentó las bases para una implementación progresiva y escalonada de las acciones priorizadas.

Nota. El acta documenta la reunión de cierre en la que se socializaron los resultados del proyecto de mejoramiento de calidad de software en PDT, incluyendo la validación de estrategias, la retroalimentación de expertos y la concertación de compromisos para la implementación progresiva de las acciones definidas.

El análisis desarrollado permitió identificar que las debilidades estructurales presentes en el área de pruebas de software de la empresa PDT tienen un impacto profundo en la calidad del producto entregado y en la eficiencia de los procesos internos; en efecto, se reconoció que la falta de metodologías formalizadas, la baja integración de herramientas de automatización, la alta rotación de personal técnico, la fragmentación documental y la escasa trazabilidad funcional afectan directamente la estabilidad operativa, ralentizan la capacidad de respuesta ante

requerimientos dinámicos y debilitan la consolidación de una cultura organizacional orientada a la mejora continua; por lo tanto, fue necesario ir más allá de un diagnóstico estático y avanzar hacia una propuesta técnica y gerencial que permita tomar decisiones basadas en datos, articulando las acciones a los objetivos estratégicos de la organización.

Desde el enfoque metodológico propuesto, basado en la Guía del PMBOK® (2021), se aplicaron herramientas como el análisis de causa raíz, los diagramas de Ishikawa, Pareto y afinidad, y sesiones estructuradas de ideación con juicio de expertos, permitiendo construir una línea de base confiable sobre los principales factores que comprometen la calidad de los entregables y la eficiencia del flujo de pruebas; estos hallazgos, sistematizados y validados con los equipos técnicos, revelaron una desconexión entre los ciclos de desarrollo y los criterios de calidad, así como una débil alineación entre los diferentes roles involucrados en la gestión de proyectos, lo que conllevó a identificar oportunidades claras de mejora en gobernanza, planificación y automatización.

A partir de esta base se diseñaron estrategias centradas en la construcción de un ecosistema CI/CD unificado, la implementación de un plan estructurado de onboarding técnico, la formalización de un marco de pruebas ágiles, la automatización de la trazabilidad de QA y la creación de un programa de gobernanza de calidad, todo esto con el propósito de fortalecer la continuidad operativa, garantizar la trazabilidad de los procesos críticos y alinear la calidad con los objetivos de entrega continua y mejora progresiva; el impacto de estas estrategias fue proyectado mediante indicadores de valor como la reducción de reprocesos, la mejora en cobertura de pruebas, el incremento de trazabilidad automatizada y la inclusión de métricas de calidad en la evaluación de desempeño, permitiendo validar su viabilidad técnica, económica y organizacional a corto y mediano plazo.

En relación con la pregunta problema, el estudio permitió demostrar que es posible optimizar el plan estratégico de los procesos de calidad en el ciclo de vida del desarrollo de software en la empresa PDT mediante la aplicación estructurada de herramientas propuestas en el PMBOK®; cada uno de los ejes estratégicos planteados respondió a una necesidad concreta identificada en la fase diagnóstica y fue construido con base en evidencia empírica y análisis colaborativos, generando un modelo flexible y escalable que puede ser replicado y adaptado a otras áreas de la organización; las respuestas a las preguntas secundarias también fueron abordadas con rigor, pues se evidenciaron las malas prácticas más recurrentes como la ejecución improvisada de pruebas, la ausencia de planes formalizados, la dependencia excesiva de hojas de cálculo y la desarticulación entre los equipos técnicos y funcionales, aspectos que impactan directamente en los costos, en los tiempos de entrega y en la percepción del cliente sobre el producto final.

El diseño de las estrategias consideró no solo las causas técnicas, metodológicas y organizativas, también integró elementos de sostenibilidad como la formación progresiva del talento interno, la institucionalización de roles de liderazgo en calidad y la medición constante del impacto a través de indicadores asociados a los OKRs del área, lo que permitirá no solo corregir fallas puntuales, también construir una cultura de calidad basada en evidencia, colaboración y responsabilidad compartida; además, la implementación de un roadmap escalonado por trimestres facilitará la ejecución ordenada de las acciones, asegurando la viabilidad y el control de los recursos, alineando cada paso con la visión estratégica de la empresa y permitiendo monitorear de manera continua el cumplimiento de metas.

Por lo cual, este trabajo no se limita a una propuesta técnica, más bien está diseñado para ser una hoja de ruta fundamentada en principios de gestión de proyectos, orientada a transformar

los procesos de calidad desde su raíz, integrando lo aprendido a lo largo de la maestría con la realidad operativa de una empresa en crecimiento, y aportando herramientas que permiten fortalecer la toma de decisiones, elevar el estándar de calidad de los servicios prestados y consolidar una base sólida para la evolución del área de pruebas dentro del ciclo de vida del desarrollo de software.

Discusión

En concordancia con los fundamentos conceptuales establecidos en el marco teórico, los resultados obtenidos en la investigación evidencian que la implementación de un enfoque integral de aseguramiento de calidad en la empresa PDT ha sido progresiva y estructurada, incorporando de manera articulada tanto principios estratégicos del PMBOK® (2021) como prácticas normativas técnicas derivadas de organismos internacionales como ISTQB e IEEE. La apropiación de estos lineamientos ha permitido avanzar desde un modelo fragmentado de calidad, centrado en acciones reactivas, hacia un modelo más robusto de mejora continua, sustentado en prácticas preventivas, automatización de pruebas y trazabilidad operativa. Esta evolución responde de manera directa a los planteamientos de Pressman (2019) y Sommerville (2016), quienes coinciden en que la calidad en software debe ser abordada desde una perspectiva sistémica, en la que confluyan atributos técnicos, satisfacción del usuario y sostenibilidad del producto en el tiempo.

De igual forma, se constató en los resultados que la empresa PDT ha consolidado un modelo híbrido de gestión de proyectos, integrando marcos predictivos y ágiles conforme a las recomendaciones de la Guía del PMBOK® (2021). Esta adopción no solo se alinea con el enfoque iterativo e incremental descrito por el PMI, sino que responde adecuadamente a la naturaleza cambiante y compleja del entorno fintech en el que opera la organización. En este sentido, la aplicación de Scrum, DevOps y Kanban en combinación con la planificación detallada de entregables a través de estructuras EDT refuerza lo planteado por Piattini et al. (2021) y Homès (2024), quienes argumentan que la sinergia entre metodologías ágiles y herramientas estructuradas favorece la eficiencia operativa, reduce el lead time y mejora sustancialmente la calidad de los productos tecnológicos desarrollados. En consecuencia, la

gestión de calidad deja de ser una actividad aislada, para convertirse en una práctica transversal que atraviesa todos los ciclos del desarrollo del software, como lo propone la misma Guía del PMBOK®.

En lo que concierne al marco normativo, los hallazgos confirman que la empresa PDT ha iniciado un proceso de alineación progresiva con las disposiciones legales vigentes en Colombia sobre protección de datos, ciberseguridad y derechos de autor. No obstante, los resultados también permiten identificar brechas relacionadas con la implementación operativa de normas como la Ley 1581 de 2012 y el Decreto 1078 de 2015, especialmente en lo que refiere a la trazabilidad documental de pruebas de cumplimiento y al fortalecimiento de protocolos de seguridad en los entornos de despliegue. Esta situación confirma lo advertido por Román (2024) y Naranjo (2024), quienes señalan que la adecuación normativa en empresas tecnológicas requiere no solo la comprensión jurídica de los marcos regulatorios, sino también su traducción en procedimientos técnicos verificables y auditablemente documentados. En este contexto, la integración normativa no debe limitarse a una exigencia jurídica, sino asumirse como un componente esencial del sistema de calidad organizacional, tal como se propone en el documento CONPES 3854 de 2016 sobre ciberseguridad y en la Circular 002 de la SIC (2024) sobre tratamiento de datos en inteligencia artificial.

Ahora bien, uno de los hallazgos más relevantes es la identificación de una relación directa entre la adopción de estándares internacionales y la competitividad organizacional, en concordancia con lo expuesto en el apartado 2.2 del marco teórico. Específicamente, la reducción de errores en producción, la mejora en los tiempos de entrega y el fortalecimiento de la confianza del cliente reflejan de manera práctica lo planteado por autores como Madhavaram et al. (2023) y Dorđević (2017), quienes sostienen que el aseguramiento de calidad impacta de manera directa la

ventaja competitiva de las empresas tecnológicas. En el caso de PDT, este impacto es particularmente visible en productos críticos como las pasarelas de pago y los sistemas de autenticación biométrica, donde la eficiencia y la confiabilidad son no solo deseables, sino indispensables. El hecho de que la empresa haya adoptado herramientas como Jenkins, Selenium y Azure DevOps, reportadas en los resultados como parte de su ecosistema técnico, confirma la relevancia de integrar herramientas adaptativas con capacidad de automatización, auditoría y trazabilidad, tal como lo sugiere la Guía del PMBOK® (2021) en el uso de artefactos especializados.

Asimismo, la incorporación de análisis de causa raíz y técnicas como el diagrama de Ishikawa en la resolución de defectos, prácticas reportadas en los resultados, evidencia una apropiación operativa de los métodos recomendados por el PMI (2021) para comprender los factores subyacentes en la aparición de fallas técnicas. Este aspecto es especialmente importante, ya que conecta directamente con la dimensión de "medición" del dominio de desempeño propuesto en el PMBOK® y confirma que la calidad en PDT no es un concepto abstracto, sino una práctica activa de análisis, control y mejora. En este sentido, se ratifica la idea de que la gestión de calidad efectiva requiere no solo marcos conceptuales sólidos, sino también herramientas operativas que traduzcan esos conceptos en acciones verificables.

Finalmente, es importante resaltar que la cultura organizacional identificada en PDT, caracterizada por la búsqueda de la mejora continua, el aprendizaje organizacional y la adaptación metodológica, se alinea de forma coherente con los principios de la Guía del PMBOK® y con las teorías contemporáneas de calidad formuladas por autores como Cuatrecasas (2012) y Senlle (2004). Esta cultura posibilita que la empresa no solo responda a los retos técnicos del entorno, sino que también se prepare estratégicamente para enfrentar nuevas

exigencias regulatorias y tecnológicas en el futuro inmediato. En este sentido, la discusión confirma que la calidad, más que un objetivo puntual, es un proceso evolutivo que requiere compromiso institucional, decisiones informadas y una visión estratégica integral.

Contribuciones

La Maestría en Gerencia de Proyectos de la Universidad Nacional Abierta y a Distancia (UNAD), constituyó la base académica que sustentó el desarrollo del presente proyecto; debido a que a lo largo de la formación se adquirieron competencias tales como planeación estratégica, formulación de proyectos, gestión de calidad y análisis económico, las cuales permitieron abordar el diagnóstico situacional, el análisis de procesos y la construcción de estrategias de manera integral y rigurosa; en particular, asignaturas como Diseño, Planeación y Formulación de Proyectos, y Gerencia de Proyectos I y II, proporcionaron herramientas para estructurar los entregables siguiendo principios de alcance, tiempo, costo y calidad, fortaleciendo así la coherencia metodológica de todo el trabajo.

De manera complementaria, el curso de Aspectos Legales aportó un enfoque jurídico fundamental que permitió integrar al proyecto consideraciones sobre protección de datos personales, derechos de propiedad intelectual y normativas aplicables en el sector tecnológico, lo que aseguró la pertinencia ética y legal de los entregables; simultáneamente, la formación en Análisis Económico de Proyectos proporcionó las bases para realizar evaluaciones de rentabilidad y sostenibilidad de las estrategias propuestas, mediante el uso de indicadores como el Retorno de Inversión (ROI) y el Valor Actual Neto (NPV), elementos imprescindibles para sustentar la viabilidad económica de las acciones planteadas.

Asimismo, las electivas disciplinares y los Seminarios de Investigación Aplicada contribuyeron a la profundización en técnicas de investigación cualitativa y cuantitativa, así como en la formulación de soluciones basadas en la evidencia, habilidades que se vieron reflejadas en la aplicación de métodos como encuestas, entrevistas, análisis de interesados, tormentas de ideas y la construcción de matrices como DOFA, Ishikawa, Pareto y mapas de

afinidad; en este sentido, cada fase del proyecto integró conocimientos teóricos y prácticos, permitiendo generar un enfoque sistémico que abordó los problemas desde múltiples dimensiones.

En cuanto a la contribución del proyecto hacia la disciplina de la Gerencia de Proyectos, resulta pertinente señalar que se demostró la importancia de la integración temprana de la calidad en el ciclo de vida de los proyectos, aplicando los principios del “Shift Left” para mejorar la detección temprana de defectos y optimizar la gestión de riesgos; igualmente, se consolidó la relevancia del involucramiento activo de los stakeholders en todas las fases, evidenciándose que la gestión eficaz de la comunicación y la participación fortalece la sostenibilidad de las acciones propuestas y facilita la adaptación organizacional al cambio.

Desde una perspectiva estratégica, el proyecto aportó a la gerencia de proyectos un modelo de intervención estructurado que combina el rigor técnico con la flexibilidad adaptativa, orientado al fortalecimiento de la madurez organizacional en aseguramiento de calidad, automatización de procesos, consolidación de prácticas ágiles y construcción de cultura de mejora continua; en consecuencia, el conocimiento generado no solo contribuye a la práctica profesional, sino que también se constituye en un insumo para el perfeccionamiento académico de la UNAD, enriqueciendo su enfoque curricular hacia la formación de gerentes de proyectos capaces de liderar transformaciones sostenibles en entornos dinámicos y altamente competitivos.

Bibliografía

- Ababneh, O. M. A. (2020). The impact of organizational culture archetypes on quality performance and total quality management: The role of employee engagement and individual values. *International Journal of Quality & Reliability Management*.
- Abad, D. N., & Ko Cerna, S. E. (2023). *Impacto de la automatización en el desarrollo de ventajas competitivas en empresas de almacenamiento de Lima Metropolitana* [Tesis de licenciatura, Universidad de Lima]. Repositorio Institucional de la Universidad de Lima. https://repositorio.ulima.edu.pe/bitstream/handle/20.500.12724/19143/T018_72084748_T.pdf?sequence=6&isAllowed=y
- Alarcón, A. C., González, J. S., & Rodríguez, S. L. (2011). Guía para pymes desarrolladoras de software, basada en la norma ISO/IEC 15504. *Revista Virtual Universidad Católica del Norte*, (34), 285-313. <https://www.redalyc.org/articulo.oa?id=194222473013>
- AlOmar, E. A., AlOmar, S. A., & Mkaouer, M. W. (2023). *Linting in software engineering education: Effectiveness in raising students' awareness of code anti-patterns*. In *2023 IEEE/ACM 45th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)* (pp. XX-XX). IEEE.
- Álvarez, Y. (2025). *Optimización de procesos de garantía de calidad de software (Quality Assurance, QA): Innovando la calidad en soluciones digitales en Object Edge a través del análisis de procesos y la automatización de la gestión de procesos de QA para transformar y agilizar las pruebas en la consultoría tecnológica* [Trabajo de pregrado, Universidad de Antioquia]. Repositorio Institucional Universidad de Antioquia. <https://hdl.handle.net/10495/44888>

- Antony, J., McDermott, O., & Sony, M. (2021). Quality 4.0 conceptualisation and theoretical understanding: A global exploratory qualitative study. *The TQM Journal*.
- Benavides, F. A. (2025). *Plan de optimización de costos aplicando la Guía del PMBOK® para mejorar la eficiencia y rentabilidad en la producción de dulce de mano de res, de la fábrica Dulce Tradición Benavides* [Proyecto aplicado, Universidad Nacional Abierta y a Distancia]. Repositorio UNAD. <https://repository.unad.edu.co/handle/10596/66697>
- Berényi, L. (2023). Project quality management in a competency-based approach. *Gradus*, 10(1). <https://doi.org/10.47833/2023.1.ECO.009>
- Callejas, M., Alarcón, A. C., & Álvarez, A. M. (2017). Modelos de calidad del software, un estado del arte. *Entramado*, 13(1), 236-250. <https://doi.org/10.18041/entramado.2017v13n1.25125>
- Campos, K. E. (2022). *Aplicación de técnicas computacionales en la predicción de procesos industriales críticos para optimizar la calidad del producto, un paso hacia adelante en la industria 4.0* [Tesis de maestría o doctorado, Nombre de la Universidad]. Repositorio Institucional. <http://hdl.handle.net/11317/2307>
- Castillo, I., Losavio, F., Boegh, J., & Matteo, A. (2010). REquirements, aspects and software quality: The REASQ model. *Journal of Object Technology*, 9(4), 69–91. https://www.jot.fm/contents/issue_2010_07/article4.html
- Chaudhry, K., & Chaudhry, M. (2023). *Competitiveness through technology and innovation* [Master's thesis, Northumbria University]. ResearchGate. <https://doi.org/10.13140/RG.2.2.18707.86568>
- Comisión Económica para América Latina y el Caribe (CEPAL). (2021). *Tecnologías digitales para un nuevo futuro* (LC/TS.2021/43). Naciones Unidas. <https://www.cepal.org>

Congreso de Colombia. (1982, 28 de enero). *Ley 23 de 1982: Sobre derechos de autor*. Diario Oficial. Recuperado de

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=3431>

Congreso de Colombia. (1987, 26 de octubre). *Ley 33 de 1987: Por medio de la cual se aprueba el "Convenio de Berna para la Protección de las Obras Literarias y Artísticas", del 9 de septiembre de 1886*. Diario Oficial. Recuperado de [https://www.suin-](https://www.suin-juriscol.gov.co/viewDocument.asp?ruta=Leyes/1589977)

[juriscol.gov.co/viewDocument.asp?ruta=Leyes/1589977](https://www.suin-juriscol.gov.co/viewDocument.asp?ruta=Leyes/1589977)

Congreso de Colombia. (1993). *Ley 44 de 1993: Por la cual se modifica y adiciona la Ley 23 de 1982 y se modifica la Ley 29 de 1944*. Recuperado de

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=34492>

Congreso de Colombia. (2008). *Ley 1266 de 2008: Por la cual se dictan disposiciones generales del Habeas Data y se regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países, y se dictan otras disposiciones*. Recuperado de

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=34488>

Congreso de Colombia. (2009, 5 de enero). *Ley 1273 de 2009: Por medio de la cual se modifica el Código Penal*. Diario Oficial. Recuperado de

<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=34492>

Congreso de Colombia. (2012, 17 de octubre). *Ley 1581 de 2012: Por la cual se dictan disposiciones generales para la protección de datos personales*. Diario Oficial No.

48.587. <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>

Congreso de Colombia. (2014). *Ley 1712 de 2014: Por medio de la cual se crea la Ley de*

Transparencia y del Derecho de Acceso a la Información Pública Nacional y se dictan

- otras disposiciones*. Recuperado de <https://www.minjusticia.gov.co/Documents/home/ABECE-transparencia-2019.pdf>
- Consejo de Europa. (2001). *Convenio sobre ciberdelincuencia*. Organización de los Estados Americanos (OEA). Recuperado de https://www.oas.org/juridico/english/cyb_pry_convenio.pdf
- Consejo Nacional de Política Económica y Social [CONPES]. (2016). *Política Nacional de Seguridad Digital*. Departamento Nacional de Planeación, Ministerio de Tecnologías de la Información y las Comunicaciones, Ministerio de Defensa Nacional y Dirección Nacional de Inteligencia. Recuperado de <https://colaboracion.dnp.gov.co/CDT/Conpes/Econ%C3%B3micos/3854.pdf>
- Dorđević, N. D. (2017). *Software quality standards*. *Vojnotehnički Glasnik / Military Technical Courier*, 65(1), 102. <https://doi.org/10.5937/vojtehg65-10668>
- Eisty, N. U., & Carver, J. C. (2022). Testing research software: A survey. *Empirical Software Engineering*, 27, Article 138.
- Fewster, M., & Graham, D. (1999). *Software test automation: Effective use of test execution tools*. Addison-Wesley.
- Gatto, D. D. de O., Evangelista, J. R. G., da Silva, R. C., & Sassi, R. J. (2023). Six Sigma methodology based on the PMBOK guide in the release management process in software developer enterprise. *Cuadernos*, 15(10). <https://doi.org/10.55905/cuadv15n10-035>
- Gupta, N., & Sahu, M. (2024). Problem faced during the software development cycle. *International Journal of Scientific Research in Engineering and Management*, 8(6). <https://doi.org/10.55041/IJSREM35471>

- Homès, B. (2024). *Fundamentals of software testing* (2nd ed., Revised and Updated). ISTE Ltd. & John Wiley & Sons. ISBN 978-1-78630-992-2.
- Institute of Electrical and Electronics Engineers (IEEE). (2021). *IEEE software standards*. IEEE.
<https://segoldmine.ppi-int.com/node/44387>
- International Software Testing Qualifications Board (ISTQB). (2024). *Welcome to ISTQB®*.
<https://www.istqb.org/>
- Irurita, J., & Villanueva, P. M. (2012). *Sistemas de gestión de la calidad* [Proyecto de fin de carrera, Universidad Pública de Navarra]. Universidad Pública de Navarra
- Jaakkola, H., & Thalheim, B. (2005, September). *Software quality and life cycles*. In Proceedings of the 9th East-European Conference on Advances in Databases and Information Systems (ADBIS 2005). Tallinn, Estonia. DBLP.
- Jones, C. (1997). *Software quality: Analysis and guidelines for success*. Thomson Learning.
- Lim, W. M., Ciasullo, M. V., Douglas, A., & Kumar, S. (2022). Environmental social governance (ESG) and total quality management (TQM): A multi-study meta-systematic review. *Total Quality Management & Business Excellence*.
<https://doi.org/10.1080/14783363.2022.2048952>
- Liu, Y., Tyszberowicz, S., Liu, Z., & Liu, B. (2022). *iTrustEval: A framework for software trustworthiness evaluation with an intelligent AHP-based method*. In Proceedings of the 2022 IEEE International Conference on Systems, Man and Cybernetics (SMC). IEEE.
- López, A. M., Cabrera, C., & Valencia, L. E. (2008). *Introducción a la calidad de software*. Scientia Et Technica, 14(39), 326-331. Universidad Tecnológica de Pereira. Recuperado de <http://www.redalyc.org/articulo.oa?id=84920503058>

- Madhavaram, S., Appan, R., Manis, K. T., & Browne, G. J. (2023). *Building capabilities for software development firm competitiveness: The role of intellectual capital and intra-firm relational capital*. *Information & Management*, 60(2), 103744.
<https://doi.org/10.1016/j.im.2022.103744>
- Martínez, S., Bogner, J., Franch, X., Oriol, M., Siebert, J., Trendowicz, A., Vollmer, A. M., & Wagner, S. (2022). Software engineering for AI-based systems: A survey. *ACM Transactions on Software Engineering and Methodology*, 31(2), Article 37e, 1-59.
<https://doi.org/10.1145/3487043>
- Meso, K., Peña, S., & Larrondo, A. (Eds.). (2021). *Desinformación y credibilidad en el ecosistema digital: Actas del XII Congreso Internacional de Cyberperiodismo*. Universidad del País Vasco / Euskal Herriko Unibertsitatea. <https://www.ehu.eus>
- Ministerio de Tecnologías de la Información y las Comunicaciones. (2022). *Resolución 00460 de 2022: Por la cual se establecen lineamientos en materia de seguridad digital en entidades públicas*. Recuperado de https://www.mintic.gov.co/portal/715/articles-198952_resolucion_00460_2022.pdf
- Momaya, K. S., & Ambastha, A. (2005). Technology management and competitiveness: Is there any relationship? *International Journal of Technology Transfer and Commercialisation*, 4(4). <https://doi.org/10.1504/IJTTC.2005.006702>
- Mustafa, A., Wan, W. M. N., Ibrahim, N., Shah, M. A., Younas, M., Khan, A., Zareei, M., & Alanazi, F. (2021). Automated test case generation from requirements: A systematic literature review. *Computers, Materials & Continua*, 68(2), 1543-1565.
<https://doi.org/10.32604/cmc.2021.014391>

Nabot, A., & Al-Qerem, A. (2024). *Impact of software quality on organizational performance*.

SSRN. <https://doi.org/10.2139/ssrn.5036455>

Naranjo, R. (2024, 10 de junio). *Normatividad Colombiana relacionada con información, privacidad de datos personales y propiedad intelectual*.

<https://www.skinait.com/normatividad-colombiana-relacionada-con-informaci%C3%B3n-privacidad-de-datos-personales-y-propiedad-intelectual-Escritos-87/>

Okpala, C. O. R., & Korzeniowska, M. (2021). Understanding the relevance of quality management in the agro-food product industry: From ethical considerations to assuring food hygiene quality safety standards and its associated processes. *Food Reviews International*, 39(4), 1879-1952. <https://doi.org/10.1080/87559129.2021.1938600>

Ozkaya, I. (2023). Application of large language models to software engineering tasks: Opportunities, risks, and implications. *IEEE Software*.

<https://doi.org/10.1109/MS.2023.3248401>

Pérez Velázquez, R. (2011). *Desarrollo de un simulador conductual para la formación en gestión empresarial basada en LEAN* [Proyecto final de carrera, Universitat Politècnica de Catalunya]. UPCommons.

https://upcommons.upc.edu/bitstream/handle/2099.1/12316/PFC_Raul_Perez_Velazquez.pdf

Piattini, M., Serrano, M., Pérez, R., Petersen, G., & Hevia, J. L. (2021). Toward a quantum software engineering. *IT Professional*, 23(1). IEEE.

Presidencia de la República de Colombia. (1989, 23 de junio). *Decreto 1360 de 1989: Por el cual se reglamenta la inscripción de soporte lógico (software) en el Registro Nacional*

- del Derecho de Autor*. Diario Oficial. Recuperado de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=10575>
- Presidencia de la República de Colombia. (2015). *Decreto 1078 de 2015: Por medio del cual se expide el Decreto Único Reglamentario del sector de Tecnologías de la Información y las Comunicaciones*. Recuperado de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=77888>
- Project Management Institute. (2021). *El estándar para la dirección de proyectos y Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK®) (7ª ed.)*. Project Management Institute. <https://www.pmi.org/>
- Rasheed, A., Zafar, B., Shehryar, T., Aslam, N. A., Sajid, M., Ali, N., Dar, S. H., & Khalid, S. (2021). Requirement engineering challenges in agile software development. *Mathematical Problems in Engineering, 2021*, Article 6696695. <https://doi.org/10.1155/2021/6696695>
- Román, E. (2024, 30 de agosto). *Entendiendo la calidad del software: qué es, qué la compone y estándares principales*. [Entrada de blog]. Innevo. Recuperado de <https://innevo.com/blog/calidad-en-el-desarrollo-de-software>
- Sader, S., Husti, I., & Daroczi, M. (2021). A review of Quality 4.0: Definitions, features, technologies, applications, and challenges. *Total Quality Management & Business Excellence, 33*(9-10), 1164-1182. <https://doi.org/10.1080/14783363.2021.1944082>
- Schwaber, K., & Sutherland, J. (2020, noviembre). *La guía de Scrum: La guía definitiva de Scrum. Las reglas del juego*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>

Sebestyén, Z., Erdei, J., & Hajdu, M. (2023). A fundamental overview of the concept of quality in project management. En M. J. Skibniewski & M. Hajdu (Eds.), *Proceedings of the Creative Construction Conference (2023)*. Keszthely, Hungary.

<https://doi.org/10.3311/CCC2023-092>

Serna M, E., Martínez M, R., & Tamayo O, P. (2019). Una revisión a la realidad de la automatización de las pruebas del software. *Computación y Sistemas*, 23(1), 169-182.

<https://doi.org/10.13053/cys-23-1-2782>

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education.

[https://dn790001.ca.archive.org/0/items/bme-vik-](https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf)

[konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf](https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf)

Spillner, A., & Linz, T. (2021). *Software testing foundations: A study guide for the Certified Tester Exam (5th ed.)*. dpunkt.verlag GmbH.

St. Quintin, M. (1993). Competitive advantage in the software industry. *Transactions on Information and Communications Technologies*, 4. WIT Press.

Sunkel, G., & Trucco, D. (Eds.). (2012). *Las tecnologías digitales frente a los desafíos de una educación inclusiva en América Latina: Algunos casos de buenas prácticas*. Comisión Económica para América Latina y el Caribe (CEPAL). Recuperado de

<https://repositorio.cepal.org/>

Superintendencia de Industria y Comercio (SIC). (2024). *Circular Externa No. 002 del 21 de agosto de 2024: Protección de datos personales*. Recuperado de

<https://sedeelectronica.sic.gov.co/sites/default/files/normativa/Circular%20Externa%20No.%20002%20del%2021%20de%20agosto%20de%202024.pdf>

- Tavakol, M., & Dennick, R. (2011). *Making sense of Cronbach's alpha*. *International Journal of Medical Education*, 2, 53–55. <https://doi.org/10.5116/ijme.4dfb.8dfd>
- Trigo, L. G. (2021). *La relevancia del marco teórico (MT) en la iniciación científica. Una aproximación desde la ciencia política y el estudio del desarrollo histórico institucional*. PANORAMA, 15(29). Politécnico Granacolombiano. <https://doi.org/10.15765/pnrm.v15i29.2536>
- Tyagi, S., Kumar, D., & Kumar, S. (2019). *Open source software: Analysis of available reliability models keeping security in the forefront*. *International Journal of Information Technology*, 14, 1041–1050.
- Urbano, M. R., & Kamran, O. (2023). *Essentials of software engineering* (5th ed.). Jones & Bartlett Learning.
- Vaniček, J. (2006). *Software quality requirements*. *Agricultura Economics – Czech*, 52(4), 177–185.
- Villa, G. (2024). *Plan de mejoramiento del sistema de gestión de la calidad de la empresa Provigasa Medical S.A.S. de conformidad con la resolución 4002 de 2007 e ISO 9001:2015. Semestre de industria* [Trabajo de pregrado, Universidad de Antioquia]. Repositorio Institucional Universidad de Antioquia. <https://hdl.handle.net/10495/42804>
- Xue, B., & Zou, L. (2023). Knowledge graph quality management: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(5), 4969-4988.

Apéndices

Apéndice A

Acta de Reunión N.1

ACTA DE REUNIÓN N.º 1	
Nombre del Proyecto	Optimización del Plan Estratégico de Calidad en el Área de Pruebas – PDT
Tipo de Reunión	Reunión de apertura del diagnóstico situacional

DATOS GENERALES

ELEMENTO	DETALLE
FECHA	-----
HORA DE INICIO	9:00 AM
HORA DE FINALIZACIÓN	10:30 AM
LUGAR	Sala virtual – Microsoft Teams (PDT)
ASISTENTES	Analistas de pruebas, líderes técnicos, testers, líderes de desarrollo, analistas de requerimientos, representantes de gestión de calidad

ORDEN DEL DÍA

Presentación del proyecto y sus objetivos.

Alineación con el enfoque la Guía del PMBOK® y herramientas metodológicas.

Socialización de cronograma de la fase de diagnóstico.

Aprobación de instrumentos.

Roles, tiempos y entregables.

DESARROLLO

OBJETIVO	Contextualizar al equipo sobre el alcance del proyecto de mejora, la metodología aplicada, los instrumentos de recolección de datos y la agenda de trabajo durante la fase de diagnóstico.	
DISCUSIÓN DE LOS PUNTOS DEL ORDEN DEL DÍA	1	Se presentó el marco general del proyecto, el cual busca diagnosticar deficiencias operacionales en el área de pruebas y proponer estrategias de optimización bajo estándares de la Guía del PMBOK®, ISTQB e IEEE.
	2	Se explicó que el enfoque será mixto: se combinarán encuestas, entrevistas, revisión documental y DOFA para identificar puntos críticos.
	3	Se validó la pertinencia de aplicar una encuesta en línea a los colaboradores de QA, desarrollo y liderazgo técnico.
	4	Se propuso agendar entrevistas con representantes de cada equipo (QA funcional, automatización, desarrollo, Scrum) para capturar conocimiento tácito.
	5	Se revisó el cronograma de trabajo para la fase de diagnóstico.

CONCLUSIONES Y ACUERDOS

Se realizará envío de la encuesta con un plazo de respuesta de una semana.

Las entrevistas estarán agendadas en una franja de 15 minutos.

Se espera lograr un análisis DOFA que sustente lo encontrado en esta fase de diagnóstico.

PROXIMAS ACCIONES

ACCIÓN	RESPONSABLE	FECHA LÍMITE
Se elaborará una nueva reunión de seguimiento	Sindy Vanessa Milton Alfredo	-----
-----	-----	-----
-----	-----	-----

Apéndice B

Acta de Reunión N.1

ACTA DE REUNIÓN N.º 2	
Nombre del Proyecto	Optimización del Plan Estratégico de Calidad en el Área de Pruebas – PDT
Tipo de Reunión	Reunión de validación de hallazgos del diagnóstico
 DATOS GENERALES	
ELEMENTO	DETALLE
FECHA	-----
HORA DE INICIO	3:00 AM
HORA DE FINALIZACIÓN	4:30 AM
LUGAR	Sala virtual – Microsoft Teams (PDT)
ASISTENTES	Analistas de pruebas, líderes técnicos, testers, líderes de desarrollo, analistas de requerimientos, representantes de gestión de calidad
 ORDEN DEL DÍA	
Presentación de hallazgos de encuestas y entrevistas.	
Revisión de patrones comunes en fallos operacionales.	
Validación de matriz DOFA preliminar.	
Aportes del equipo sobre brechas críticas.	
Recomendaciones iniciales y próxima fase.	

DESARROLLO

OBJETIVO	Socializar los resultados preliminares del diagnóstico, validar colectivamente las principales problemáticas y priorizar focos de mejora en el área de pruebas.
DISCUSIÓN DE LOS PUNTOS DEL ORDEN DEL DÍA	1 Se presentó una síntesis de resultados obtenidos en las 54 encuestas aplicadas, destacando los mayores puntos de dolor: baja automatización, ausencia de planes de prueba, alta rotación en proveedores y documentación dispersa.
	2 Se compartieron los fragmentos más representativos de las entrevistas, los cuales evidencian debilidades estructurales y conocimiento tácito no capitalizado.
	3 Se expuso la matriz DOFA, la cual fue validada y complementada por los asistentes.
	4 Se acordó priorizar en la siguiente fase los siguientes focos: Automatización de flujos críticos. Trazabilidad de planes de prueba. Implementación de métricas e indicadores.

CONCLUSIONES Y ACUERDOS

El entregable del diagnóstico final incluirá:

Actas de reunión

Análisis cuantitativo y cualitativo

Matriz DOFA**Documento técnico de diagnóstico****PROXIMAS ACCIONES**

ACCIÓN	RESPONSABLE	FECHA LÍMITE
Se elaborará borrador de las estrategias de mejora.	Sindy Vanessa Milton Alfredo	-----
-----	-----	-----

Apéndice C

Respuestas de Entrevistas Semiestructuradas

ENTREVISTA - Chapter Lead de Calidad (CLDC)

¿Cuáles considera que son las principales barreras actuales que enfrenta el área de pruebas de software en PDT para garantizar la calidad en los entregables?

Actualmente, el área de pruebas presenta un desempeño funcional aceptable, pero no óptimo. Aunque logramos cumplir con la mayoría de los compromisos establecidos en términos de entregables, existen cuellos de botella estructurales que impactan la eficiencia. Por ejemplo, los procesos manuales siguen siendo predominantes en etapas donde la automatización debería ser la norma. Además, nuestra cobertura funcional aún no es lo suficientemente robusta como para garantizar una trazabilidad completa, lo cual limita la capacidad para prevenir regresiones en releases sucesivos. El cumplimiento de objetivos se da más por el esfuerzo operativo que por la madurez del proceso, lo cual no es sostenible en el tiempo.

Desde su rol, ¿cómo percibe la alineación entre los equipos de desarrollo, calidad y producto respecto al entendimiento del alcance y prioridades del proyecto?

En la práctica, esa alineación es parcial. Aunque trabajamos bajo un marco ágil, con ceremonias y artefactos compartidos, muchas veces el entendimiento del alcance no está completamente sincronizado entre los roles. Esto se debe, en gran medida, a una definición inicial ambigua del alcance por parte de los stakeholders del negocio. Lo que observamos es que los equipos de desarrollo priorizan entregables tangibles, mientras que QA tiende a enfocarse en estabilidad y cobertura. Esa diferencia de enfoque, si no se gestiona adecuadamente desde la planificación, deriva en entregas funcionales pero débiles en términos de calidad estructural.

La ausencia de planes de pruebas documentados y validados por el equipo de desarrollo y negocio es una de las principales deficiencias. Asimismo, existe una práctica recurrente de ejecutar pruebas sin criterios de entrada o salida bien definidos, lo que da lugar a resultados ambiguos y reejecuciones sin valor agregado. La falta de revisión cruzada de scripts y de métricas centralizadas también genera una percepción errónea del estado real del producto. Finalmente, se observa una tendencia a no aprovechar las capacidades del versionamiento automatizado en herramientas de integración continua, lo que expone al sistema a errores evitables.

¿Qué efectos ha observado que genera la rotación de personal en los procesos de pruebas? ¿Cómo afecta esto la calidad y la eficiencia?

La rotación tiene un impacto negativo directo en la continuidad operativa. Cuando un tester con conocimiento del dominio del negocio o de los procesos específicos de validación abandona el equipo, se pierde una parte importante del conocimiento tácito. Esto obliga a los nuevos miembros a atravesar curvas de aprendizaje largas, ralentizando los ciclos y aumentando los errores por desconocimiento. También se afecta el diseño de los casos de prueba, que en muchos casos quedan desactualizados o duplicados. Además, afecta el clima del equipo y genera inseguridad en los procesos de revisión cruzada y mejora continua.

Uno de los mayores desafíos es la rotación constante del personal, especialmente en los perfiles de automatización. Esto afecta la consolidación del conocimiento y obliga a repetir ciclos de entrenamiento, lo que ralentiza el avance hacia prácticas maduras. También enfrentamos resistencia al cambio cuando se intenta implementar frameworks de automatización o herramientas nuevas.

Muchas veces, el foco del equipo está puesto en “salir del paso” y no en pensar la calidad como un proceso transversal desde el inicio del proyecto. A esto se suma la limitada participación del área

de calidad en fases tempranas de planificación, lo cual nos relega a un rol de control al final del ciclo, en lugar de ser parte activa en la construcción de soluciones robustas.

En cuanto a los métodos y herramientas utilizadas, ¿cree que están alineadas con las necesidades reales de los proyectos? ¿Qué mejoras considera necesarias?

Primero, es indispensable adoptar un enfoque Shift Left real, en el que el área de calidad participe desde la etapa de análisis de requerimientos. Esto permitirá definir criterios de aceptación y escenarios de prueba antes del desarrollo. Segundo, recomendaría institucionalizar una batería mínima de pruebas automatizadas por sprint, orientada a los flujos de negocio críticos. Tercero, urge establecer tableros de control con métricas accionables —tales como cobertura de pruebas, defectos por tipo, tiempos de ciclo— que sirvan de base para la toma de decisiones. Finalmente, se deben fortalecer los procesos de retroalimentación con el cliente y las áreas funcionales, para validar continuamente que lo que se prueba responde a sus necesidades reales y no solo a los entregables contractuales.

Hay herramientas robustas, pero su uso no siempre es consistente o integral. Contamos con plataformas para gestión de pruebas, automatización y análisis, pero muchas veces no se integran en un flujo continuo. Además, hay equipos que siguen utilizando plantillas manuales o procesos heredados, lo que impide una trazabilidad efectiva. Necesitamos avanzar hacia una arquitectura de QA unificada, con herramientas interoperables y enfocadas en la experiencia del usuario interno. También urge fortalecer las competencias del equipo en metodologías como TDD, BDD o pruebas basadas en modelos, que son poco exploradas en el contexto actual.

¿Qué papel juega la planificación en la calidad de las pruebas? ¿Qué consecuencias ha traído su ausencia o deficiencia?

La comunicación inter-áreas es un eje transversal en la calidad del software. En mi experiencia, los errores más críticos no surgen por fallas técnicas, sino por malentendidos entre requerimientos, diseño y validación. Cuando la información no fluye de forma clara entre los Product Owners, analistas de requerimientos, desarrolladores y testers, se generan desviaciones que se detectan tarde, aumentando los costos de corrección. Una práctica que ha dado buenos resultados es realizar sesiones de refinamiento cruzado, donde QA valida en vivo los criterios de aceptación junto con desarrollo y negocio. Estas sesiones ayudan a alinear expectativas, anticipar riesgos y evitar reprocesos innecesarios.

¿Cómo evalúa el nivel de documentación y trazabilidad en los procesos actuales de QA?

¿Representa esto un apoyo o una carga?

En este momento, la documentación representa más una carga que un apoyo. Si bien existe la intención de dejar trazabilidad, el volumen de documentación requerida muchas veces sobrepasa la capacidad del equipo, especialmente en proyectos de alta velocidad. Además, al no estar centralizada o automatizada, consultar el histórico se vuelve complejo y poco práctico. Una posible solución sería adoptar estrategias de documentación ligera, basada en evidencias automatizadas y reportes ejecutivos generados a partir de dashboards, lo cual permitiría mantener la trazabilidad sin comprometer la agilidad.

Desde su experiencia, ¿cuáles son las estrategias más efectivas para fortalecer la cultura de calidad en entornos de alta demanda como el de PDT?

Fortalecer la cultura de calidad requiere una combinación de liderazgo visible, empoderamiento técnico y reconocimiento del valor del QA en la cadena de entrega. Una estrategia efectiva ha sido involucrar a los testers desde las etapas iniciales de diseño, incluso en sesiones de refinamiento con producto. Otra clave ha sido la formación continua en buenas prácticas y el establecimiento de comunidades internas de práctica, donde se comparten errores, aprendizajes y técnicas. Finalmente, visibilizar el impacto de la calidad en métricas de negocio —como disminución de incidencias post-producción o mejora en tiempos de entrega— ha sido fundamental para que otras áreas valoren el rol estratégico de QA.

ENTREVISTA - Scrum Master y Facilitador de DevOps

¿Cuáles considera que son las principales barreras actuales que enfrenta el área de pruebas de software en PDT para garantizar la calidad en los entregables?

Desde mi rol, lo que observo con mayor frecuencia es una desconexión entre la visión técnica del área de QA y las necesidades del negocio. Hay una falta de integración efectiva entre los pipelines de entrega continua y los ciclos de validación, lo que genera cuellos de botella cuando las pruebas no están automatizadas ni adaptadas a los cambios frecuentes del producto. También existe una carga operativa considerable derivada del exceso de pruebas manuales, especialmente en regresiones, lo cual es insostenible en un entorno de despliegue frecuente. La falta de visibilidad sobre el estado real de la calidad en los entornos de integración también limita la toma de decisiones informadas en tiempo real.

Desde su rol, ¿cómo percibe la alineación entre los equipos de desarrollo, calidad y producto respecto al entendimiento del alcance y prioridades del proyecto?

Hay avances importantes, pero aún enfrentamos desafíos en términos de entendimiento compartido. Muchas veces, el equipo de desarrollo empieza a construir funcionalidades antes de tener claros los criterios de aceptación, lo que implica ajustes posteriores y, en algunos casos, retrabajo. Las prioridades cambian con rapidez, y cuando no se gestionan adecuadamente desde los tableros de planificación o desde las ceremonias Scrum (como el refinamiento o el sprint planning), esto genera un desbalance entre entregables y calidad. La clave está en fortalecer la colaboración tripartita: producto–dev–QA, desde una lógica de co-creación, no de silos.

¿Qué efectos ha observado que genera la rotación de personal en los procesos de pruebas? ¿Cómo afecta esto la calidad y la eficiencia?

La rotación impacta particularmente la estabilidad del ciclo DevOps. Cada vez que perdemos un recurso con experiencia, se rompen las cadenas de conocimiento dentro del flujo de valor. Esto se traduce en mayor tiempo para configurar entornos, redefinir pruebas automatizadas y entender las dependencias entre módulos. Desde el punto de vista de la agilidad, también genera fricción en el equipo, porque implica rediseñar dinámicas de colaboración y reformular acuerdos internos. A nivel técnico, afecta los pipelines, ya que quienes ingresan requieren tiempo para familiarizarse con los scripts, herramientas y escenarios críticos, generando retrasos.

En cuanto a los métodos y herramientas utilizadas, ¿cree que están alineadas con las necesidades reales de los proyectos? ¿Qué mejoras considera necesarias?

Tenemos herramientas potentes, pero su adopción no es homogénea. En algunos equipos utilizamos Jenkins, SonarQube, Selenium y herramientas de integración continua bastante maduras, pero en otros seguimos anclados a procesos manuales o herramientas obsoletas. Se necesita una estrategia

corporativa de consolidación de herramientas con prácticas DevOps que fomenten el versionamiento de pruebas, pruebas en paralelo y validaciones automáticas por microservicio.

También es clave invertir en entrenamiento del equipo para garantizar que se usen todas las funcionalidades disponibles, no solo las básicas.

¿Qué papel juega la planificación en la calidad de las pruebas? ¿Qué consecuencias ha traído su ausencia o deficiencia?

Planificar no es solo definir qué se va a probar, sino cuándo, con qué herramientas y en qué entorno. Cuando esto no ocurre, los testers entran al sprint tarde, sin claridad sobre qué validar, y eso nos obliga a hacer validaciones apuradas o incluso dejar pruebas críticas para ciclos posteriores. He visto proyectos donde la falta de planificación provocó regresiones no controladas en ambientes de producción. En DevOps, esto también genera fallos en los pipelines de integración y despliegue, porque los test scripts no están listos o no cubren los escenarios de riesgo. La calidad no debe improvisarse.

¿Cómo evalúa el nivel de documentación y trazabilidad en los procesos actuales de QA?

¿Representa esto un apoyo o una carga?

Actualmente representa más una carga que un valor agregado, principalmente porque no se ha digitalizado ni automatizado. Documentamos en múltiples repositorios: algunos en Confluence, otros en Excel, y muchos quedan en herramientas de gestión que no se conectan con los reportes técnicos. Esto fragmenta la trazabilidad. Desde DevOps, proponemos dashboards integrados con métricas en tiempo real, reportes automáticos desde herramientas de testing y documentación viva

que se actualice con cada commit. Esa sería la forma de convertir la trazabilidad en una ventaja competitiva, no en una burocracia.

Desde su experiencia, ¿cuáles son las estrategias más efectivas para fortalecer la cultura de calidad en entornos de alta demanda como el de PDT?

La más efectiva es integrar la calidad como responsabilidad compartida. El QA no es un rol, es una mentalidad que debe permear desde el Product Owner hasta el último desarrollador. En nuestra práctica, una buena estrategia ha sido incluir checkpoints de calidad en cada definición de listo (“Definition of Done”), promover pruebas automáticas desde el inicio del desarrollo (shift-left testing), y fomentar pair testing entre devs y testers. Además, el uso de retrospectivas enfocadas en errores y métricas de mejora continua ha fortalecido el compromiso del equipo. Pero, sobre todo, la calidad se construye con confianza, visibilidad y autonomía técnica.

ENTREVISTA - Gerente de Producto

¿Cuáles considera que son las principales barreras actuales que enfrenta el área de pruebas de software en PDT para garantizar la calidad en los entregables?

La mayor barrera que percibo está en la desconexión entre los criterios de valor del negocio y los criterios técnicos de calidad. Muchas veces, el enfoque de pruebas se orienta únicamente a lo funcional, dejando de lado dimensiones clave como la experiencia del usuario, la escalabilidad o la robustez del producto en condiciones reales. Además, el área de QA suele ingresar de forma tardía en los ciclos de desarrollo, cuando ya hay presión por entregar, lo cual limita su capacidad de intervención. Esto genera una dinámica reactiva, en lugar de preventiva, y termina comprometiendo el valor percibido por el cliente.

Desde su rol, ¿cómo percibe la alineación entre los equipos de desarrollo, calidad y producto respecto al entendimiento del alcance y prioridades del proyecto?

Existe un esfuerzo continuo por alinear las prioridades, pero aún se presentan inconsistencias, sobre todo cuando las definiciones de alcance no se actualizan de manera iterativa. En ocasiones, se asume que todos comprenden lo mismo por “alcance”, pero no siempre hay un documento fuente, como una historia bien refinada o un criterio de aceptación claro. Esto afecta el entendimiento común entre Producto, Desarrollo y QA, y puede generar retrabajo. La alineación mejora considerablemente cuando se emplean herramientas visuales, sesiones de refinamiento bien estructuradas y tableros vivos con trazabilidad de requerimientos.

¿Qué efectos ha observado que genera la rotación de personal en los procesos de pruebas? ¿Cómo afecta esto la calidad y la eficiencia?

La rotación tiene un impacto profundo en la estabilidad de los proyectos. Desde el punto de vista del producto, afecta la consistencia en la comprensión del negocio y el dominio funcional de los testers sobre los flujos críticos. También implica perder contextos valiosos que no están documentados, lo que genera puntos ciegos en los casos de prueba. Esto se traduce en errores en producción que pudieron haberse evitado, además de un aumento en los tiempos de onboarding y una menor eficiencia en los sprints. Es un problema que se agrava cuando no hay una cultura sólida de documentación y transferencia de conocimiento.

En cuanto a los métodos y herramientas utilizadas, ¿cree que están alineadas con las necesidades reales de los proyectos? ¿Qué mejoras considera necesarias?

Desde mi perspectiva, las herramientas son adecuadas, pero la forma en que se utilizan podría mejorar. Por ejemplo, tenemos frameworks de automatización que no se aprovechan al máximo, o herramientas de seguimiento que no están integradas entre áreas. Lo ideal sería tener una suite integrada que permita conectar historias de usuario, criterios de aceptación, casos de prueba, evidencias y métricas de cobertura en un solo lugar. Además, hace falta mayor capacitación sobre la lógica de valor del negocio para que las pruebas no solo validen lo técnico, sino también lo que realmente espera el usuario final.

¿Qué papel juega la planificación en la calidad de las pruebas? ¿Qué consecuencias ha traído su ausencia o deficiencia?

La planificación es fundamental. Cuando no se hace con la anticipación debida, el equipo de QA termina recibiendo funcionalidades en fases demasiado avanzadas, con poco margen de maniobra. Esto fuerza a priorizar pruebas por intuición o urgencia, en lugar de seguir un plan basado en riesgos y valor. Desde Producto, esto representa un riesgo estratégico porque se pueden liberar versiones con defectos en componentes que impactan directamente la experiencia del cliente. La falta de planificación también debilita la confianza del negocio en el proceso técnico.

¿Cómo evalúa el nivel de documentación y trazabilidad en los procesos actuales de QA?

¿Representa esto un apoyo o una carga?

Actualmente, la trazabilidad es intermitente. En algunos proyectos está muy bien definida, pero en otros la documentación es insuficiente o dispersa. Esto se vuelve un problema cuando hay que hacer seguimiento a defectos reportados por los usuarios o auditar los criterios con los que se validaron ciertas funcionalidades. Si no hay trazabilidad, no hay forma de saber si un requerimiento

fue realmente probado o si el defecto se relaciona con un vacío en la validación. Necesitamos fortalecer la documentación como un instrumento de transparencia, no como una carga operativa.

Desde su experiencia, ¿cuáles son las estrategias más efectivas para fortalecer la cultura de calidad en entornos de alta demanda como el de PDT?

En mi experiencia, la más efectiva es la co-responsabilidad. Cuando los equipos de desarrollo y QA entienden que su trabajo impacta directamente en la satisfacción del cliente, hay un cambio radical en la cultura. También hemos tenido buenos resultados promoviendo indicadores visibles de calidad, como el porcentaje de regresiones, la cobertura de pruebas automatizadas y los tiempos de resolución de defectos. A esto se suma la figura del “champion de calidad” dentro de cada célula, que actúa como referente y mentor. Finalmente, el reconocimiento público a las buenas prácticas es una herramienta poderosa para motivar al equipo.

ENTREVISTA - Analista Senior de QA Automatizado

¿Cuáles considera que son las principales barreras actuales que enfrenta el área de pruebas de software en PDT para garantizar la calidad en los entregables?

Desde mi experiencia en QA Automatizado, uno de los mayores desafíos es la falta de cobertura automatizada sobre los flujos críticos de negocio. Aunque hemos avanzado en frameworks y pipelines CI/CD, no todos los proyectos cuentan con suites completas o mantenidas. Además, aún existe resistencia a considerar la automatización como parte del desarrollo, y se deja como una tarea secundaria. Esto se agrava cuando hay entregas sin tiempo suficiente para estabilizar los scripts o validar ambientes. También enfrentamos dificultades cuando no hay sincronía entre QA y desarrollo desde etapas tempranas.

Desde su rol, ¿cómo percibe la alineación entre los equipos de desarrollo, calidad y producto respecto al entendimiento del alcance y prioridades del proyecto?

La alineación mejora cuando hay ceremonias como refinamiento bien estructuradas y Definition of Done claras. Sin embargo, todavía hay muchas historias que llegan a QA con criterios ambiguos o incompletos, lo cual genera interpretaciones distintas entre los equipos. Esto impacta directamente en la definición de los casos de prueba y en la automatización. En algunos casos, desarrollamos pruebas para flujos que luego cambian por decisiones de producto que no se comunican a tiempo. Necesitamos que el alcance se mantenga sincronizado y visible para todos.

¿Qué efectos ha observado que genera la rotación de personal en los procesos de pruebas? ¿Cómo afecta esto la calidad y la eficiencia?

La rotación nos afecta gravemente, sobre todo en automatización. Cada tester tiene un estilo de codificación, estructura de repositorios y convenciones propias. Cuando alguien se va, no siempre hay handover ni documentación actualizada de los scripts. Esto implica que quien llega debe hacer ingeniería inversa sobre lo ya construido, lo cual ralentiza los procesos y puede generar errores. Además, muchos de los problemas resueltos en el pasado no están registrados como lecciones aprendidas, lo que genera una repetición constante de errores.

En cuanto a los métodos y herramientas utilizadas, ¿cree que están alineadas con las necesidades reales de los proyectos? ¿Qué mejoras considera necesarias?

Usamos herramientas potentes como Selenium, Postman, Jenkins y TestRail, pero hay una falta de estandarización en su uso. Cada célula tiene implementaciones distintas, y no siempre hay un marco común de automatización por capas. Además, hace falta inversión en formación y en

herramientas más avanzadas como Cypress, Playwright o soluciones de performance testing.

También se requieren licencias para herramientas de testeo integradas en CI/CD, que permitan acelerar los ciclos de validación y liberar versiones más estables con menos intervención manual.

¿Qué papel juega la planificación en la calidad de las pruebas? ¿Qué consecuencias ha traído su ausencia o deficiencia?

La planificación es crítica. Cuando no hay planificación en pruebas, las automatizaciones se desarrollan tarde, o se quedan sin ejecutar por falta de tiempo o ambientes. Esto genera deuda técnica y cobertura incompleta. Además, sin una estrategia de pruebas bien definida desde el Sprint 0, el testing se convierte en una carrera contrarreloj al final de cada iteración. Hemos tenido casos en los que se liberan funcionalidades sin haber pasado por pruebas automatizadas porque “no hay tiempo”. Esto pone en riesgo la calidad y mina la confianza en los procesos de QA.

¿Cómo evalúa el nivel de documentación y trazabilidad en los procesos actuales de QA?

¿Representa esto un apoyo o una carga?

En general, la documentación aún no está al nivel que necesitamos. Muchos scripts automatizados carecen de comentarios, no están versionados correctamente o no tienen una guía de ejecución.

Aunque usamos herramientas como Jira y Confluence, la trazabilidad entre los requisitos, casos de prueba y defectos no siempre es clara ni consistente. Esto representa una carga en el momento en que se necesita rastrear el origen de un fallo o justificar una cobertura. La documentación debe verse como parte del proceso de calidad, no como un documento aparte.

Desde su experiencia, ¿cuáles son las estrategias más efectivas para fortalecer la cultura de calidad en entornos de alta demanda como el de PDT?

La más efectiva ha sido integrar QA en los squads desde el inicio, con voz y voto en todas las decisiones del producto. También ayuda mucho tener pipelines con “gates” de calidad obligatorios antes de hacer merge o release. La cultura de revisión cruzada entre testers (code review y test review) también eleva el estándar técnico. Por último, mostrar métricas de calidad (como tasa de defectos evitados, cobertura de regresiones y tiempos de ejecución de suites) ayuda a visibilizar el valor de QA y a posicionarlo como un actor clave, no como un paso posterior.

ENTREVISTA - Agile Coach y Líder de Portafolio Estratégico

¿Cuáles considera que son las principales barreras actuales que enfrenta el área de pruebas de software en PDT para garantizar la calidad en los entregables?

Desde una visión sistémica, las principales barreras no radican únicamente en la ejecución operativa del testing, sino en aspectos estructurales que afectan el flujo de valor. En PDT, todavía enfrentamos cuellos de botella causados por silos funcionales, una débil integración de QA desde las fases iniciales y una falta de visibilidad en los criterios de calidad a nivel de portafolio. La calidad sigue viéndose como una responsabilidad del equipo de pruebas, cuando en realidad debe ser una propiedad compartida por todo el sistema organizacional. El reto es evolucionar hacia un modelo donde la calidad esté embebida en la cultura, no tercerizada a un equipo aislado.

Desde su rol, ¿cómo percibe la alineación entre los equipos de desarrollo, calidad y producto respecto al entendimiento del alcance y prioridades del proyecto?

Aún existe fragmentación entre los roles, especialmente cuando los artefactos de valor no están visibilizados en tableros compartidos, o cuando los objetivos estratégicos no se traducen correctamente en épicas e historias funcionales. Si bien hemos avanzado con prácticas ágiles como PI Planning, refinamiento conjunto y roadmaps colaborativos, falta consistencia. Muchos equipos trabajan bajo presión y con backlog técnico acumulado, lo que lleva a priorizar la velocidad sobre la calidad. La alineación real exige conversaciones difíciles sobre trade-offs, participación activa de QA en la definición del alcance y un entendimiento común de qué es valor.

¿Qué efectos ha observado que genera la rotación de personal en los procesos de pruebas? ¿Cómo afecta esto la calidad y la eficiencia?

La rotación tiene un costo de alineación muy alto. Cada vez que un colaborador se va, no solo se pierde conocimiento técnico, sino también entendimiento contextual del negocio, de las restricciones no documentadas y de las relaciones tácitas entre módulos. Esto impacta directamente la calidad, porque se rompe la continuidad del aprendizaje y se reinicia la curva de madurez de los equipos. Hemos intentado mitigar esto con pairing, mentoring cruzado y documentación colaborativa, pero mientras no abordemos la causa raíz —que está ligada al bienestar laboral, la sobrecarga y la falta de carrera profesional clara—, la rotación seguirá generando deuda organizacional.

En cuanto a los métodos y herramientas utilizadas, ¿cree que están alineadas con las necesidades reales de los proyectos? ¿Qué mejoras considera necesarias?

Desde el portafolio vemos una heterogeneidad considerable. Algunas células tienen frameworks maduros y pipelines robustos, mientras que otras aún operan con procesos manuales y herramientas

no integradas. Esta brecha técnica no solo genera ineficiencia, sino también desigualdad en la calidad de los entregables. Necesitamos un marco de referencia compartido, con mínimos técnicos comunes y prácticas de calidad adoptadas a nivel organizacional. La mejora no pasa por comprar más herramientas, sino por definir estándares evolutivos, asegurar que los equipos tengan espacio para el aprendizaje continuo y promover la automatización como una inversión en resiliencia, no como un lujo.

¿Qué papel juega la planificación en la calidad de las pruebas? ¿Qué consecuencias ha traído su ausencia o deficiencia?

La planificación es el puente entre intención y ejecución. Cuando no planificamos adecuadamente las actividades de QA, lo que hacemos es desplazar la calidad hacia etapas tardías, donde ya no es posible prevenir, solo contener. La ausencia de una estrategia de pruebas por release, una gestión de dependencias bien mapeada y una capacidad de carga bien calculada genera improvisación. Esto se traduce en entregas reactivas, bugs en producción y retrabajo. Hemos identificado que donde hay planificación anticipada (por ejemplo, mediante roadmaps de calidad o backlog técnico sincronizado), los equipos entregan valor más sostenible y predecible.

¿Cómo evalúa el nivel de documentación y trazabilidad en los procesos actuales de QA?

¿Representa esto un apoyo o una carga?

La trazabilidad es clave en entornos regulados y en productos de misión crítica, como los que desarrolla PDT. Sin embargo, su aplicación aún es inconsistente. Algunos equipos ven la documentación como una carga porque no está integrada al flujo de trabajo ni automatizada. La clave está en adoptar herramientas que permitan una trazabilidad viva: conectar historias con casos

de prueba, defectos, despliegues y métricas en tiempo real. Esto transforma la documentación de un requisito burocrático en una fuente estratégica para la mejora continua, auditoría y toma de decisiones.

Desde su experiencia, ¿cuáles son las estrategias más efectivas para fortalecer la cultura de calidad en entornos de alta demanda como el de PDT?

La primera estrategia es elevar la calidad al nivel de conversación estratégica. No debe ser un tema operativo, sino una prioridad de negocio. La segunda es transversalizarla: todos los roles —desde producto hasta infraestructura— deben tener OKRs vinculados a prácticas de calidad. Tercero, construir comunidad técnica: comunidades de práctica, espacios de aprendizaje, guilds de automatización, intercambio de lecciones aprendidas. Y, por último, construir con propósito.

Cuando los equipos entienden cómo su trabajo impacta al usuario final, la calidad deja de ser una obligación para convertirse en un compromiso.

Apéndice D

Certificado de Participación como Conferencistas



EL DIRECTOR DE LA OFICINA DE SERVICIOS AL INGENIERO DE ACIEM

Asociación
Colombiana de
Ingenieros

CERTIFICA QUE:

La Ingeniera en Telecomunicaciones, **SINDY VANESSA VERGEL**, identificada con la cédula de ciudadanía 52.987.999 de Bogotá y el Ingeniero Industrial **MILTON GOMEZ** identificado con la cédula de ciudadanía 11.445.507 de Facatativá; compartieron sus valiosos conocimientos y experiencias con la comunidad de Ingeniería durante la conferencia virtual titulada "Herramientas para el Mejoramiento del Plan Estratégico de Calidad".

Dado en Bogotá D.C., a los dieciocho días (18) del mes de marzo de dos mil veinticinco (2025).

Calle 70 No. 9 - 10
PBX: (57)3127393
www.aciem.org
Bogotá D.C.

Cordialmente

MAURICIO HERNANDEZ GÓMEZ

Director

Departamento de Servicios al Ingeniero

Cuerpo Técnico Consultivo
del Gobierno Nacional
Ley 51 de 1986