UNIVERSIDAD Rafael Belloso Chacín
Revista Electrónica de Estudios Telemáticos
TÉLÉMATIQUE

# Aseguramiento de la Calidad en Proyectos Informáticos: Una Nueva Dimensión desde la Ingeniería de Software

# Quality Assurance in Computer Projects: A New Dimension from Software Engineering

Díaz Alonso Ricardo Alejandro
Correo: od.ricardodiaz@gmail.com
Orcid: 0000-0002-8245-7906
**Universidad José Antonio Páez, Venezuela.**

**Resumen**

La gestión tecnológica corporativa ha sido impulsada por la demanda de esquemas funcionales avanzados, con énfasis en el aseguramiento de la calidad en el desarrollo informático. No obstante, se han identificado escenarios en el sector de tecnologías de la información carentes de estándares metodológicos, lo que ha planteado desafíos significativos. En respuesta, se estableció como objetivo general del estudio, desarrollar un modelo metodológico basado en ingeniería de software para el aseguramiento de la calidad en el proceso de desarrollo informático, orientado a las empresas productoras de sistemas. De este modo, se precisó un discurso teórico en la autoría de Pressman y Maxim (2021), Blandon y Jaramillo (2023), así como la ISO (2020), enfocado en la ingeniería de software, procesos, metodologías, incluyéndose la administración y factores de la calidad del área. La investigación planteó un enfoque cuantitativo, aplicado, proyectivo, de campo y no experimental. En orden, se ejecutaron las técnicas de la observación y encuesta, para la aplicación de una lista de verificación y cuestionario, previamente validados por juicio de expertos y con un índice de confiabilidad KR-20 de 0,89, para una muestra de 45 participantes, provenientes de 3 empresas del sector. Finalmente, los resultados evidenciaron el alejamiento pronunciado de las entidades respecto a los principios y prácticas de la ingeniería de software, y estándares de calidad informática, lo que fortificó el desarrollo del Modelo InsideOut como un marco metodológico replicable y escalable, capaz de transformar y optimizar procesos informáticos en la industria, combinando especialización, calidad y practicidad, ofreciendo un modelo robusto que impulsa la mejora continua y establece un referente para la excelencia en el desarrollo de software.

**Palabras clave:** Calidad, Ingeniería, Metodología, Modelo, Software.

**Abstract**

Corporate technology management has been driven by the demand for advanced functional schemes, with an emphasis on quality assurance in IT development. However, scenarios have been identified in the information technology sector lacking methodological standards, which has posed significant challenges. In response, the general objective of the study was established to develop a methodological model based on software engineering for quality assurance in the IT development process, aimed at systems production companies. Thus, a theoretical discourse was specified by Pressman and Maxim (2021), Blandon and Jaramillo (2023), as well as ISO (2020), focused on software engineering, processes, methodologies, including management and quality factors in the area. The research proposed a quantitative, applied, projective, field and non-experimental approach. In order, observation and survey techniques were executed, for the application of a checklist and questionnaire, previously validated by expert judgment and with a KR-20 reliability index of 0.89, for a sample of 45 participants, from 3 companies in the sector. Finally, the results showed the pronounced distancing of the entities from the principles and practices of software engineering, and IT quality standards, which strengthened the development

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

of the InsideOut Model as a replicable and scalable methodological framework, capable of transforming and optimizing IT processes in the industry, combining specialization, quality and practicality, offering a robust model that drives continuous improvement and establishes a benchmark for excellence in software development.

**Keywords:** Quality, Engineering, Methodology, Model, Software

## Introducción

La tecnología ha transformado de múltiples maneras la sociedad actual, de igual modo se ha convertido en un elemento de vital importancia para el crecimiento y desarrollo de una era cada vez más globalizada. La comprobación de su notable relevancia la afirman las diversas aplicaciones establecidas para la generación de cambios estructurales dentro de las grandes áreas de la industria latente; salud, economía, educación, política, comercio y negocios, han propiciado nuevos escenarios interrelacionados gracias a la sistematización de sus procesos.
Aunado a lo expuesto, el desarrollo de sistemas de información y aplicaciones tecnológicas, ha permitido una amplia apertura para la constitución de nuevos enfoques de trabajo, destacando positivos índices de progreso en la consecución de una práctica profesional automatizada. De esta manera, se han brindado numerosas soluciones empresariales tras la implantación de tecnologías para el soporte de la gestión corporativa.

Partiendo de lo expuesto, es válido acotar que los requerimientos de las tecnologías de la información demandados por individuos, negocios y gobiernos se han tornado más complejos en los últimos años, así lo describen Pressman y Maxim (2021), asintiendo que, se ha direccionado el análisis de la práctica hacia la conformación de equipos profesionales que introdujeran consigo, la inserción de enfoques sinérgicos, en donde yacía previamente, una orientación individual del trabajo. Esta creciente exigencia ha generado, a su vez, una dependencia continua hacia el software, en entornos para la toma de decisiones estratégicas y tácticas, así como para las operaciones y controles cotidianos; tomando la falla del software, como una experiencia de compleja dimensión, lo que implica una afección que viaja desde un impacto menor hasta niveles adversos.

Ecuménicamente, corporaciones como IBM, Microsoft, Apple, han incursionado profundamente en el desarrollo de modelos efectivos para la construcción de software de calidad. Y es que, dentro del mundo hoy vivido, se ha permitido la edificación de la ingeniería de software, como una disciplina de fácil inmersión en el argot establecido por los practicantes de la industria. De allí que, Díaz y Somaza (2020:26) en referencia a Jordán (2014), doctor honoris causa por la Universidad Politécnica de Madrid, España, y actualmente considerado uno de los mayores expertos en ingeniería de software a nivel mundial, declara:

> **Software tiene un amplio significado económico. Software es el principal material en que se basan los negocios y la infraestructura. Software es ahora la fuente de ventaja competitiva en sectores que van desde los servicios financieros y la sanidad a las telecomunicaciones, logística, transporte, espectáculos, y empresas de servicios públicos. El software es el fundamento de la civilización moderna. Constituye y controlará productos, servicios, y toda la infraestructura de la que depende el ser humano. No creo que esta definición sea exagerada.**

Por lo tanto, la aceptación del software, como herramienta para el exponenciación de las dimensiones de un negocio, y por extensión para el mejoramiento de la calidad de vida, ha de tornarse ineludible dentro el pensamiento del individuo común. Sin embargo, la obtención de soluciones informáticas, no ha concebido en su totalidad, la alineación y diferenciación necesaria

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

entre los términos práctica y proceso dentro de la ingeniería (Carrizo y Alfaro, 2018). Lo que supone, la presencia de un síntoma de inestabilidad e imprecisión inicial en el ejercicio tecnológico. Ante ello, Serna (2021), ha declarado la existencia de una clara diferenciación entre las prácticas y los procesos en el ámbito de la ingeniería de software. Las primeras representan las actividades o tareas a ejecutar, mientras que los últimos establecen la secuencia lógica y la interrelación entre estas actividades.

La observación de la experiencia desarrollada ha implicado una reflexión oportuna en relación al empleo de la práctica profesional, en concordancia con el seguimiento de los procesos estructurados dentro de la ingeniería de software. Si bien se ha permitido el ingreso de metodologías de acompañamiento y soporte, el mercado global ha mostrado un paradigma flexible y en ocasiones difuso para la ejecución y el desempeño, debido a la creciente desinformación, en unión con la desculturización del sector en cuanto a la práctica real de la ingeniería. Se habla entonces, de la presencia de un marco de resistencia o alejamiento de las bases sistemáticas, disciplinadas y cuantificables del software. Este panorama es avalado en la opinión de Santos (2009), referenciado por Peláez et al. (2019:117) quienes permiten en su disertación, un segundo acercamiento hacia lo que pudiese catalogarse como un potencial escenario de necesidad:

> **No existe una forma de organización global definida para la industria del software que rija su desarrollo o forma de implementación. Cada país, según sus características y posibilidades adopta la vía que considera más favorable en ese sentido. La industria de las tecnologías de la información y comunicaciones está fuertemente sujeta al desarrollo económico de cada nación. Por consiguiente, la industria del software como apéndice de la industria de las nuevas tecnologías se encuentra bajo las mismas condiciones.**

No obstante, la creciente demanda que impulsa el mercado consumidor ha exigido, a quienes en su ejercicio insertan la producción tecnológica, adoptar esquemas funcionales superiores, procurando el aseguramiento de la calidad como un punto medular inmerso en la construcción informática. Sin embargo, dentro de la progresista era digital, se han comprobado escenarios de desarrollo con un limitado acompañamiento metodológico, lo que ha simbolizado un entorno problemático comúnmente detectado, y a su vez integrado al sector de tecnologías de información. Este hecho ha sido sustentado en la aparición de un número inexacto de organizaciones y desarrolladores particulares, dedicados a la producción informática, incluyendo perspectivas formales e informales de trabajo; otorgándose, un plano secundario a la admisión de metodologías, así como paradigmas de trabajo para el aseguramiento de la calidad del software como producto, y con ello, cada uno de los atributos fundamentales que persiguen su producción (Sinisterra et al., 2021).

La premisa planteada ha concentrado su origen en los escenarios anteriormente definidos, en unión con las prácticas tomadas como pilares de construcción de proyectos de software, las cuales datan de hace más de 40 años, lo que revela una planeación poco atractiva y de limitada trascendencia. De igual forma, las técnicas implementadas para el análisis y diseño de sistemas han propiciado la aparición de problemas de seguridad por el grado de obsolescencia que las últimas reportan. Históricamente, Blandon y Jaramillo (2023), han manifestado la precedencia en los proyectos informáticos, de niveles de elevada complejidad difícilmente manejables desde una visión fiable, lo que ha estimulado el fracaso en la implementación de la ingeniería de software.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

Este estamento ha apuntado a la consecución de entornos de incertidumbre en el mercado generador de este tipo de bien digital. Se han percibido índices de desconfianza en el producto propiamente dicho, al mantenerse el oscurantismo reinado en la forma de producción de software, infraestructuras y sistemas de información. Es así como, se ha perfilado un profesional difidente, cuyos parámetros de evaluación de la calidad, cálculo de costes, así como el dimensionamiento de resultados han presentado imprecisiones, reservando una mirada periférica a los estándares y patrones que contemplan el ejercicio profesional. Como último elemento, ha tomado posición la crisis del usuario, la cual alude al fenómeno relacionado con el desajuste entre las capacidades o características del software desarrollado y las demandas emergentes o cambiantes de los usuarios (Serna, 2021).

Se ha producido entonces, la caracterización de una situación problemática, a partir de los elementos causales anteriormente definidos en el sector de tecnologías de información, reflejada en la limitada aplicación de métodos efectivos, basados en los principios de la ingeniería de software, para el aseguramiento de la calidad durante las etapas que conforman el proceso de desarrollo informático.

Como medida general para el control del pronóstico trazado, es insertada la incorporación de modelos integradores que combinen, dentro de su composición, las buenas prácticas y herramientas que han precedido la construcción informática a nivel mundial. Para lo propio, se hace oportuna la adaptación de estándares y parámetros dedicados al seguimiento y validación del proceso de desarrollo según cada dominio de aplicación de software, en unión con la resignificación, reeducación y concienciación generalizada en torno a la vitalidad e importancia que impregna la correcta implementación de la ingeniería de software. De cualquier modo, dentro de la gestión informática integral debe permanecer un fuerte compromiso, tomando como foco la sensibilización del profesional sobre los requerimientos del usuario, atendiendo como misión primordial, la resolución de necesidades en calidad, efectividad, transformando el software-producto en ingeniería.

## Objetivo

El objetivo general de la investigación fue desarrollar un modelo metodológico basado en ingeniería de software para el aseguramiento de la calidad en el proceso de desarrollo informático, orientado a las empresas productoras de sistemas pertenecientes al sector de tecnologías de información.

## Fundamentación Teórica

La revisión e incorporación de la literatura ha simbolizado un impacto altamente favorable dentro de los estudios que, dentro de su estructura, persiguen la aplicación de adecuadas metodologías de investigación. De este modo, se verifica la introducción de la perspectiva teórica, como una introducción a la visión sobre la cual se sitúa el planteamiento propuesto dentro del campo de conocimiento adoptado como escenario de investigación.

### Ingeniería de Software

Una definición completa es proporcionada por Pressman y Maxim (2021:11), citando a su vez al IEEE (1993) indicando que el elemento representa “la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software”. La ingeniería de software es entonces, una forma de tecnología con varias capas, las cuales comprenden entornos adaptables y ágiles. Su enfoque está también basado en un compromiso organizacional, incorporando la administración total de la calidad, Six Sigma y demás filosofías afines, las cuales alimentan la cultura de mejora continua.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

En ese orden, la capa proceso es el fundamento de la ingeniería de software, proporcionando un marco de trabajo que integra los aspectos tecnológicos, de gestión y de calidad. El proceso de ingeniería de software define entonces, una estructura formal que guía el desarrollo de software, desde la concepción hasta la entrega, a través de la definición de métodos, productos de trabajo y puntos de control, el proceso asegura la consistencia, la trazabilidad y la mejora continua (Blandon y Jaramillo, 2023).

De allí que, los métodos de la ingeniería de software proporcionan la experiencia técnica para elaborar productos informáticos, incluyendo un conjunto amplio de tareas, tales como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo (Pressman y Maxim, 2021). Por otra parte, dentro de la disciplina, es incluida una serie de herramientas, las cuales proporcionan un apoyo automatizado o semiautomatizado para el proceso y los métodos. La integración de herramientas de software que permiten el intercambio de información constituye la base de la ingeniería de software asistida por computadora (CASE). Esta disciplina proporciona un conjunto de herramientas y metodologías que automatizan y soportan las actividades del desarrollo de software.

Por tanto, la ingeniería de software, como campo en constante evolución, transciende la mera creación de código. Es una disciplina interdisciplinaria que abarca desde la concepción de ideas hasta la gestión de sistemas complejos. Considerando que, la disciplina dualmente no solo produce el software en sí, sino que también establece los cimientos metodológicos y tecnológicos para su desarrollo, operación y mantenimiento, la convierte en un escenario multifactorial sumergido en desafíos y oportunidades, donde la innovación es norma para el abordaje de las demandas cada vez más sofisticadas del mundo digital.

**Proceso del Software**

El proceso del software, citando a Pressman y Maxim (2021:12), “destaca la definición de un grupo de elementos involucrados en el proceso de desarrollo informático, antes de realizar una conceptualización del término referido”. Los autores hacen alusión al proceso como un conjunto de actividades, acciones y tareas ejecutadas durante la creación de algún producto de trabajo. Es así como una actividad, busca lograr un objetivo amplio, y es desarrollada sin importar el dominio de la aplicación, tamaño del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería de software.

Por otra parte, se incluye la arquitectura del diseño, a partir de una acción, esta centra su atención en un objetivo de corto alcance, pero debidamente definido. Con la introducción de los elementos anteriores, es posible una aproximación avanzada a la concepción del proceso del software, nuevamente Pressman y Maxim (2021:12), indican que, “un proceso no es una prescripción rígida de cómo elaborar el software de cómputo. Por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo busquen y elijan el conjunto apropiado de acciones y tareas”. Ante tal aseveración, es posible la admisión de una terminología ágil, traducida en un ejercicio profesional flexible y adecuado.

En ese orden de ideas, el proceso del software busca incorporar las mejores prácticas adaptadas a la realidad del desarrollo, procurando la entrega del producto en forma oportuna y con la calidad suficiente para satisfacer a quienes patrocinaron su creación, así como aquellos que fungirán como usuarios. Por tanto, la esquematización del proceso establece un fundamento para el desarrollo completo de la ingeniería de software, por medio de la introducción de un grupo limitado de tareas aplicables a todos los proyectos informáticos, sin mesuras en cuanto su

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

tamaño o complejidad (Serna, 2021). De acuerdo a lo planteado, la estructura del proceso identifica las actividades estructurales definidas a partir del Cuadro 1.

**Cuadro 1. Actividades Estructurales**

| PROCESO DEL SOFTWARE | |
|---|---|
| **Actividad** | **Detalle** |
| Comunicación | Destaca la importancia crítica de la comunicación con el cliente, antes del inicio de cualquier trabajo técnico. Se busca entender los objetivos de los involucrados respecto al proyecto, para la generación de los requerimientos que ayuden a definir las características y funciones del software a desarrollar. |

**Cuadro 1. (Cont.)**

| PROCESO DEL SOFTWARE | |
|---|---|
| **Actividad** | **Detalle** |
| Planeación | Alimentada por el proceso de comunicación. Se establece el plan o mapa guía del proyecto de software, se define el trabajo de ingeniería, al describir las tareas técnicas por realizar, los riesgos probables, los recursos requeridos, los productos del trabajo que se obtendrán y una programación de las actividades propuestas. |
| Modelado | Se crea un bosquejo del objeto con el fin de entender el panorama general. El modelado del sistema permite obtener una comprensión de cómo se verá arquitectónicamente el producto terminado, y cómo ajustan entre sí las partes constituyentes de dicho producto, entre otros aspectos inmersos, con el fin de lograr una traducción real y acentuada de los requerimientos del software y el diseño que los satisfará. |
| Construcción | Se permite un enlace entre la generación del código, manual o automatizada, y las pruebas que se requieren para descubrir los errores en éste. Se producen continuos procesos de programación, alimentados por procedimientos metodológicos para la inserción de códigos y patrones de verificación de la funcionalidad, dando respuesta a los requerimientos de usuario. |
| Despliegue | Se propicia la entrega del software según las especificaciones acordadas en las actividades predecesoras. Por tanto, el software, como entidad completa o como un incremento parcialmente terminado, se entrega al consumidor para su evaluación y oportuna retroalimentación. De esta forma, se permite el despacho del producto incorporando las capacidades de instalación, adecuación y configuración para una entrega enfocada en la calidad del desarrollo. |

**Fuente:** Adaptado de Pressman y Maxim (2021).

Las actividades estructurales definidas representan un compendio genérico usado tanto en el desarrollo de programas de corto alcance, para la creación de grandes aplicaciones Web, e incluso en la ingeniería de sistemas enormes basados en computadoras (Muñoz, 2019). No obstante, los detalles del proceso del software, estarán en función del caso o escenario presentado, conservando en esencia la estructura mostrada. Para muchos proyectos de software, las actividades estructurales se aplican en forma iterativa a medida que avanza dicho proyecto. Es decir, la comunicación, planeación, modelado, construcción y despliegue son

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

ejecutadas a través de un cierto número de repeticiones. Cada una de ellas produce un incremento del software, lo cual provee a los participantes un subconjunto de características y funcionalidades generales del software. Conforme se produce cada incremento, el producto se hace más completo (Pin et al., 2020).

**Factores de la Calidad ISO/IEC 25000:2014**

La Organización Internacional para la Normalización (International Organization for Standarization), en adelante ISO, como organismo encargado de promover el desarrollo, así como la estandarización de normas internacionales para la fabricación y seguridad de productos y servicios, ha introducido el estándar un conjunto de factores de calidad informática proviniendo inicialmente desde el correlativo 9126. En torno a ello, Pressman y Maxim (2021:343), amplía, "el estándar ISO 9126 se desarrolló con la intención de identificar los atributos clave del software de cómputo".

Siendo esta norma publicada en 1992, bajo la denominación Information Technology-Software product evaluation-quality characteristics and guidelines for their use (Evaluación de las características de calidad en los productos de software de tecnologías de la información y lineamientos para su uso), permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software.

No obstante, una versión mejorada de la normativa fue anunciada por la ISO con el fin de optimizar los procesos de aseguramiento de la calidad dentro del desarrollo informático, bajo el nombre de Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) (Ingeniería de sistemas y software - Requisitos de calidad y evaluación de sistemas y software). En ese orden de ideas, ISO (2020), publicó, ISO/IEC 25000:2014 proporciona una guía para la organización, enriquecimiento y unificación de las series que cubren dos procesos principales, especificación de requisitos de calidad y evaluación de la calidad del software, soportada por el proceso de medición de calidad. Por tanto, el modelo coordina el tratamiento de dos métricas, interna y externa, detallados seguidamente:

1. En la calidad interna, se aprecia, la totalidad de las características del producto de software desde una visión interna, componentes y productos intermedios. Es decir, es medida y evaluada contra los requisitos de calidad también internos, considerando que los detalles de calidad del producto pueden ser mejorados durante la implementación, revisión y pruebas del código. Sin embargo, la naturaleza fundamental de la calidad del producto informático representada por la calidad interna, debe permanecer sin cambios a menos que se lleven a cabo actividades de rediseño (ISO, 2020).
2. La calidad externa, por su parte implica la inserción de la visión externa, diseminada en el producto final. Esta es la calidad percibida en el software cuando es ejecutado, el cual es típicamente medido y evaluado durante las pruebas en un entorno simulado con datos simulados utilizando métricas externas (ISO, 2020). Por lo que, el tópico citado es referido por excelencia a un ambiente de validación funcional, tras la incorporación de un conjunto de casos de prueba asociados al proceso.

De manera que, durante las pruebas, la mayoría de los fallos deben ser cubiertos y eliminados; no obstante, es posible conservar algún remante después de ser aplicadas las pruebas. En este caso, el diseño fundamental del software debe permanecer invariable durante el proceso, por la complejidad que implica algún cambio en el mismo. Habiéndose definido las

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

estructuras medulares del estándar, es posible la incorporación de los seis atributos clave de calidad que presenta el modelo dentro de su contenido, a partir de la visualización del Cuadro 2.

**Cuadro 2. Atributos de Calidad del Software**

| FACTORES DE CALIDAD DEL SOFTWARE | |
|---|---|
| **Atributo** | **Detalle** |
| Funcionalidad | Grado en el que el software satisface las necesidades según los sub-atributos adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad. Esta característica es expresada en términos de cumplimiento operativo y adaptativo del producto informático. |

**Cuadro 2. (Cont.)**

| FACTORES DE CALIDAD DEL SOFTWARE | |
|---|---|
| **Atributo** | **Detalle** |
| Confiabilidad | Cantidad de tiempo que el software se encuentra disponible para su uso, de acuerdo a los sub-atributos: madurez, tolerancia a fallas y recuperación. Por lo tanto, un software con esta característica es capaz de asegurar un nivel de funcionamiento adecuado cuando es utilizado en condiciones específicas. Así, la confiabilidad se amplía al sostener un nivel especificado de funcionamiento y no una función requerida del sistema. |
| Usabilidad | Grado en el que el software es fácil de usar, a partir de los sub-atributos entendible, aprendible y operable. Responde a la capacidad del software de ser comprendido y usado en forma fácil y atractiva. Así mismo, está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido. |
| Eficiencia | Nivel en el que el software emplea óptimamente los recursos del sistema, en línea con los sub-atributos comportamiento del tiempo y de los recursos. Por lo que, la eficiencia está estrechamente vinculada a la obtención de una forma de desempeño adecuado, de acuerdo al número de recursos utilizados según un conjunto de condiciones previamente especificadas. A ello, se atañen aspectos como la configuración del hardware, sistemas operativos, entre otros componentes involucrados. |
| Mantenibilidad | Facilidad con la que pueden efectuarse reparaciones al software, en función de los sub-atributos analizable, cambiable, estable, susceptible de someterse a pruebas. De allí que, la capacidad de mantenimiento es la cualidad que tiene el software para ser modificado; incluyendo a su vez, correcciones o mejoras al software, cambios en el entorno, y especificaciones de requerimientos funcionales de acuerdo al contexto desarrollado. |
| Portabilidad | Versatilidad con la que el software puede llevarse de un ambiente a otro según lo resuelven los sub-atributos adaptable, instalable, conformidad y sustituible. Por tanto, se centran las propiedades que posee el producto informático de ser trasladado o ejecutado en diferentes plataformas y/o sistemas operativos. |

**Fuente:** Adaptado de ISO (2020); Pressman y Maxim (2021).

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

De este modo, la norma ISO/IEC 25000:2014 estableció un marco conceptual y metodológico para la evaluación rigurosa y sistemática de la calidad del software. Es importante destacar que, su implementación efectiva requiere un enfoque integral que incorpore la participación activa de todos los actores involucrados en el proceso de desarrollo. Considerando a su vez, la necesidad de adaptación a diversos contextos organizacionales y la definición de métricas específicas y precisas para cada atributo de calidad.

## Metodología

El estudio implementó un esquema metodológico con un enfoque cuantitativo, aplicado, proyectivo, de campo y con un diseño no experimental. En este marco, se desarrollaron tres fases metodológicas orientadas a: (a) el diagnóstico, (b) el diseño de la propuesta y (c) el desarrollo de la propuesta. Las técnicas de recolección de datos incluyeron la observación y la encuesta, para las cuales se emplearon una lista de verificación y un cuestionario como instrumentos asociados.

En ese orden de ideas, el estudio reconoció como principal limitación dentro del diseño no experimental, la imposibilidad de establecer relaciones de causalidad directa, dado que no se ejerció control sobre las variables. No obstante, esta necesidad fue mitigada a partir de la triangulación de las técnicas e instrumentos citados, lo cual permitió incrementar la validez de los datos recolectados y profundizar en el análisis de los hallazgos.

Partiendo de la naturaleza del estudio, y en función del alcance del investigador sobre el escenario donde confluyen los hechos, se propuso la selección de una muestra no probabilística, de tipo censal conformada por 45 participantes, provenientes de tres empresas del sector de tecnologías de información, en la ciudad de Valencia, estado Carabobo, Venezuela; respondiendo, a la necesidad de inclusión de la totalidad del universo de participantes directamente relacionados con los procesos de aseguramiento de calidad en las empresas seleccionadas. Esta estrategia garantizó que los resultados reflejaran con fidelidad las perspectivas y realidades del contexto investigado, contribuyendo así al fortalecimiento de la validez externa del estudio.

De este modo, los instrumentos fueron validados por un panel de expertos, garantizando su pertinencia, claridad y alineación con los objetivos de investigación. El índice de confiabilidad obtenido fue de 0,89, calculado mediante la Fórmula 20 de Kuder-Richardson, lo que indicó un nivel de consistencia interna elevado. Esta alta confiabilidad respaldó la solidez de los instrumentos para captar información precisa y confiable, asegurando que los datos recolectados reflejaran adecuadamente las características del fenómeno estudiado.

## Resultados

La estructuración del proceso de análisis e interpretación de los datos condujo a la generación de los resultados de la investigación. Basándose en la disertación teórica de Pressman y Maxim (2021), la función de calidad en el proceso de desarrollo de software se entiende como un elemento clave que asegura la transformación efectiva de las necesidades y expectativas del usuario en estándares concretos que garanticen la calidad tanto del proceso como del producto final. Este enfoque implica la identificación y satisfacción de los requerimientos

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

del usuario, en paralelo con la implementación de prácticas y herramientas que permitan mantener la calidad a lo largo de todo el ciclo de desarrollo, desde la planificación inicial hasta la entrega del software.

De esta manera, se busca establecer una correlación directa entre las demandas del usuario y los criterios de calidad aplicados, asegurando que cada etapa del desarrollo contribuya a un resultado óptimo y funcional. En ese orden, la variable mencionada, fue evaluada dentro de la investigación, a partir de sus dimensiones e indicadores, en respuesta a la aplicación del instrumento de recolección de datos, en la forma de la lista de verificación, cuyos resultados han sido reflejados en la Tabla 1, a continuación**.**

**Tabla 1.** *Medición de la variable función de calidad en el proceso de desarrollo de software*

| Dimensión | Indicador | Ítem | Distribución de Frecuencias | | Cierre Variable |
|---|---|---|---|---|---|
| | | | Si (%) | No (%) | |
| Calidad del software | Factores de calidad | Alinean sus productos a las normas de calidad definidas internacionalmente. | 8,89% | 91,11% | Sí = 26,14% No = 73,86% |
| | Funcionalidad | Miden el grado de satisfacción del software en función de su adaptabilidad, exactitud e interoperabilidad. | 33,33% | 66,67% | |
| | Confiabilidad | Arquean métodos para la detección de errores, tolerancia a fallas y recuperación de información. | 31,11% | 68,89% | |
| | Eficiencia | Valoran el grado de optimización de los recursos del sistema. | 27,78% | 72,22% | |
| | Mantenimiento | Centran su producción en el desarrollo de programas estables, sostenibles y con apertura a reparaciones. | 30,56% | 69,44% | |
| | Portabilidad | Generan productos informáticos adecuados para diferentes ambientes y estaciones de trabajo. | 25,18% | 74,82% | |

**Fuente:** Elaboración propia (2022)

En primer término, la evaluación del indicador factores de calidad en el proceso del software demostró en un 91,11%, la no alineación de las metodologías de desarrollo a las normas de calidad definidas internacionalmente. Tal proporción exacerbó notablemente el impacto generado en el producto informático, en torno a los parámetros generales de calidad aplicados durante su ciclo de vida. En ese sentido, la desestimación la trascendencia generada a partir de la instauración de los parámetros mencionados, implicó a su vez la minimización de la eficacia y el valor del software generado.

Estos componentes han sido abiertamente insertados, tras la teoría expuesta por Pressman Y Maxim (2021:340), “en el sentido más general la calidad del software se define como el proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y utilizan”. Por tanto, el alejamiento pronunciado de los parámetros de calidad, señaló un deficiente control y aseguramiento de las propiedades, así como las capacidades de satisfacción del bien digital reportadas por el usuario. Al hecho planteado, se adosa la separación de los elementos que componen la administración de la calidad, los cuales en el pensamiento de Maquera et al. (2016), abarcan un ciclo de vida desde

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

la captura de requisitos y la creación de modelos conceptuales, hasta la codificación, las pruebas unitarias, de integración y de sistema, y la entrega del producto final.

En continuidad, la evaluación del atributo funcionalidad arrojó en promedio un índice positivo del 33,33%, obteniéndose un importante 66,67% de inclinación negativa. Analizar el comportamiento de los valores presentados, conllevó a la observación de un punto álgido de desatención sobre las características del software, expresadas en términos de cumplimiento operativo y adaptativo del producto informático. En adición a lo planteado, es admisible inferir un incumplimiento de las tareas y objetivos propuestos por el sistema desarrollado, relegando el propósito fundamental del atributo. Por lo que, la escasa vigilancia de la propiedad mencionada incurrió en la inserción de procesos desprovistos de completitud funcional, corrección oportuna, y pertinencia operativa (Serna, 2021), en términos de interoperabilidad, seguridad, exactitud y grado de satisfacción del software.

Por su parte, el atributo confiabilidad reportó una media del 31,11% de efectividad en el tratamiento del indicador. El hallazgo evidenciado, apeló contra la seguridad esencial que el producto informático necesita ofrecer al usuario, en función de la madurez, tolerancia a fallas y recuperación. En ese sentido, la recesión del atributo estudiado significó una producción desprovista de mecanismos para la evaluación del grado de funcionamiento del software sin proclividad a fallar. Consecutivamente, la evaluación de la usabilidad del software demostró un 46,67% de inclusión del atributo en el ciclo de vida, reflejándose una discreta capacidad para el entendimiento, aprendizaje y uso del mismo en ambientes considerados amigables y atractivos por los usuarios.

Por tanto, subestimar el atributo estudiado, ordena la separación del usuario del correcto manejo del producto tecnológico, excluyendo adicionalmente, aspectos como flexibilidad, robustez y accesibilidad de los componentes funcionales del software. En esa línea, la atención del indicador de eficiencia reveló en un 27,78%, su grado de aplicación bajo los estándares internacionales que fomentan la calidad de la intervención. Este panorama manifestó una restringida vinculación del proceso hacia la obtención de una forma de desempeño adecuado, con base en un número de recursos utilizados, para la maximización funcional siguiendo un conjunto de condiciones previamente especificadas.

Desde este punto de vista, es considerada necesaria la adaptación de la praxis; de modo que, las propiedades mesurables del software puedan ser determinadas a partir de una evaluación objetiva, basándose en el desempeño expresado, así como en la experiencia del usuario sobre el producto considerado. Seguidamente, la medición del indicador mantenimiento fijó un promedio de ejecución del 30,56%, coartando la facilidad con la que pueden efectuarse reparaciones al software, en función del nivel de análisis integrativo, susceptibilidad a cambios y tolerancia a pruebas (Pressman y Maxim, 2021).

De allí que, se determinó que la capacidad de mantenimiento, como cualidad que posee el software para ser modificado, junto con sus correcciones o mejoras, resultó comprometida al destinar esfuerzos parcialmente suficientes, con el fin de proporcionar cambios en el entorno, y especificaciones adicionales de requerimientos funcionales. Esta situación contravino los principios de una gestión de software eficaz, donde los procesos establecidos deben garantizar el control y medir el nivel de riesgo, factores determinantes en la calidad del producto final.

Como cierre en la medición de los atributos del software, se calculó un promedio de ejecución efectiva del 25,18% en cuanto al nivel de portabilidad evidenciado en el ciclo. Tomando en

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

consideración la valoración porcentual, se estableció un producto desarrollado con escasa flexibilidad de movilización, exponiendo el grado de adecuación, conformidad, instalación y sustitución. Por lo tanto, dentro de esta característica, y basando el análisis en los hallazgos extraídos, se han relegado las propiedades poseídas por el producto informático, no únicamente para ser trasladado o ejecutado en diferentes plataformas y/o sistemas operativos, sino también para ser adaptado de acuerdo a las necesidades del usuario, incidiendo notablemente en la universalidad y capacidad de comercialización del producto. De este modo, con la evaluación de los atributos del software, se produjo la exploración de la segunda variable de la investigación determinada en las metodologías de desarrollo de software en la industria, a partir de la visualización de la tabla 2, cuyos resultados han sido producto de la aplicación del cuestionario.

**Tabla 2.** *Medición de la variable metodologías de desarrollo de software en la industria*

| Dimensión | Indicador | Ítem | Distribución de Frecuencias | | Cierre Variable |
|---|---|---|---|---|---|
| | | | Si (%) | No (%) | |
| Paradigma de desarrollo | Método de desarrollo | Grado de empleo y adaptación metodológica según cada tipo de software. | 27,40% | 72,60% | Sí = 23,70%<br>No = 72,30% |
| | Roles del equipo de desarrollo | Nivel de definición de las operaciones para cada rol desempeñado por el personal de la división. | 14,44% | 85,56% | |
| | Sistemas de referencia y proyección de calidad | Implementación de pruebas y parámetros para el aseguramiento de la calidad del software. | 29,27% | 70,73% | |

**Fuente:** Elaboración propia (2023)

En continuidad, el indicador método de desarrollo manejó una frecuencia promedio del 27,40% de aceptación y aplicación en términos de efectividad y continuidad. Este abordaje condujo a la determinación de espacios de trabajo desprovistos de esquemas organizativos y sistemáticos de la práctica establecida. Asegurar este hecho, condujo a la desestimación del enfoque medular del proceso de análisis y diseño de sistemas, cuya premisa principal consiste en que los sistemas se desarrollan mejor utilizando un ciclo específico de actividades del analista y el usuario (Carrizo y Alfaro, 2018). Por consiguiente, el desarrollo de un sistema informático carente de una metodología adecuada, es tendente a la generación de expectativas comúnmente insatisfechas, lo que con frecuencia deriva en el abandono del producto informático una vez establecido su despliegue (Pin et al., 2020).

Por su parte, la definición de operaciones para cada rol dentro del proceso, así como la dotación de actualizaciones y formación continua de los actores, comprobó un 14,44% de efectiva aplicación en el ciclo. Al respecto, la evolución del talento humano, debe permanecer estrechamente vinculada al comportamiento y gestión de los líderes que encabezan los proyectos emprendidos (Caicedo, 2023). Asimismo, la cohesión establecida entre los diferentes participantes del proyecto simboliza un punto indiscutible de éxito o fracaso en la ejecución de las tareas establecidas.

Complementariamente, Flores et al. (2018) han precisado que, los equipos con una fuerte identidad y objetivos comunes son más productivos y motivados. Por lo que, desestimar esta premisa puede desencadenar equipos disfuncionales, caracterizados por conflictos internos, baja moral y una disminución en la calidad del producto final.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

En último término, la valoración de los sistemas de referencia y proyección de la calidad obtuvo un promedio de ejecución favorable reflejado en el 29,27% de la muestra. Si bien, la calidad ha sido abiertamente referida a un conjunto de propiedades y capacidades de satisfacción inherentes a un objeto (Guevara et al., 2024), dentro del sector estudiado se ha establecido un alejamiento importante en torno a la concepción de mecanismos que administren el concepto presentado.

Estos resultados han sustentado la adopción parcial de los modelos de calidad, lo que restringe el acceso a herramientas y metodologías que pueden guiar a las organizaciones hacia la excelencia operativa. Este hecho, por extensión, ha truncado el proceso de seguimiento y control de los diferentes proyectos involucrados; así como también, la administración del riesgo, el establecimiento de revisiones técnicas y mediciones del software. Desencadenando, por consiguiente, un nulo aseguramiento de la calidad.

De este modo, la acentuación negativa detectada sobre los componentes sometidos a evaluación ha concretado, el alejamiento pronunciado de las empresas del sector de tecnologías de información respecto a los postulados, principios y prácticas proporcionadas por la ingeniería de software, lo que ha traído como principal efecto, la desestimación de la calidad evidenciada en la línea de productos informáticos desarrollada por las organizaciones consultadas.

En esa línea, el oscurantismo precedido dentro de la práctica de la ingeniería de software, ha tomado una evolución significativa, gracias al fortalecimiento de nuevos esquemas de trabajo, en pro de la construcción de escenarios legítimos para la aplicación de la disciplina. Así mismo, consolidar el esfuerzo reportado por un equipo interdisciplinario ha comprometido a su vez, una perspectiva ulterior sobre qué, cuándo y cómo debería desarrollarse el producto en atención.

Si bien, la definición del software no ha representado, para los diferentes especialistas de la industria, una sencilla labor, se ha hecho preciso reafirmar que el término comprende un conjunto estructurado de instrucciones, procedimientos, programas, normativas y documentación, diseñado para ser implementado en soportes digitales o físicos, con el propósito de permitir, gestionar y optimizar el funcionamiento de sistemas de procesamiento electrónico de datos (Ibarra y Muñoz, 2018). Lo que permite la fijación de la teoría de sistemas, indicándose que los inputs y outputs son virtualmente inmateriales, con casi nulos costos de transporte a elevada velocidad, como inferencia anidada. A pesar de que, el desarrollo de software y sus servicios relacionados han sido caracterizados como intensivos en trabajo calificado y con requerimientos generalmente bajos en términos de capital físico, siguen subsistiendo problemas de calidad, confiabilidad, cumplimiento de tiempos, entre otros factores que han reducido la proyección efectiva del producto confeccionado.

En ese orden de ideas, el enfoque de la ingeniería de software continúa aún en debate, generando diversas controversias en función de la naturaleza de su aplicación, lo que ha originado un estudio en contraste. Es por ello que, la introducción de estándares y parámetros especializados dentro del proceso del software, se ha convertido en una decisión trascendental dentro de la determinación de la calidad del producto. Este esfuerzo ha insertado, de igual modo, la asimilación de experiencias mejoradas, basadas en los estilos predecesores, y orientadas hacia la obtención de modelos de desarrollo efectivos.

Bajo esta premisa, y con base en el diagnóstico de la investigación, se produjo el desarrollo de un modelo metodológico, basado en ingeniería de software, para el aseguramiento de la calidad en el desarrollo informático. En orden, se otorgó la denominación InsideOut, siendo éste

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

un equivalente a la expresión en español, desde adentro hacia fuera, reforzando la propiedad cohesiva fundamental de la solución planteada. Desde esta perspectiva, la propuesta insertada estuvo caracterizada de acuerdo a los elementos considerados en el Cuadro 3.

**Cuadro 3.** Características del Modelo InsideOut

| MODELO INSIDEOUT | |
|---|---|
| **Características** | **Detalle** |
| Visión sinérgica | InsideOut persigue la instauración de espacios armónicos para el desarrollo integral del equipo profesional, fusionando el conjunto de factores colaborativos inmersos, con la intención de crear un efecto extra y comunicativo dentro del proceso del software. |
| Perspectiva concurrente | Se ofrece al usuario calificado una experiencia basada en la simultaneidad. Así, el equipo adiciona a su práctica la posibilidad de promover un conjunto de procesos o hilos de ejecución basados en múltiples tareas, combinando las bondades de los métodos prescriptivos y ágiles contemplados en el desarrollo de software. |
| Especialización | El modelo InsideOut garantiza la selección de la mejor ruta para iniciar el ciclo de vida del producto con la identificación oportuna del dominio de aplicación de software. |
| Practicidad | InsideOut no solo establece un marco estructural para la definición y caracterización del proceso productivo. Por demás, invita al usuario a la aplicación práctica de los métodos, estándares y nociones, de acuerdo a cada tipo de software. |
| Calidad absoluta | El aseguramiento de la calidad se ha convertido en uno de los focos principales reportados dentro de InsideOut. De allí que, el modelo se encuentra alineado a un conjunto de experticias mundiales, entre las que destaca fundamentalmente el estándar internacional ISO/IEC 25000:2014, como garantía de las propiedades reflejadas en el producto desarrollado. |
| Producción continua | En cada etapa del ciclo se organiza la elaboración de artefactos e incrementos parciales, hasta concurrir en la totalidad del producto terminado. Es por ello que, InsideOut se muestra leal a la idea de confeccionar documentos que articulen el ejercicio técnico-profesional alcanzado por el equipo de alto desempeño. |

**Fuente:** Elaboración propia (2022)

En continuidad, el Modelo InsideOut configuró la atención del proceso del software a través del desarrollo de cinco (5) estaciones para el desarrollo progresivo del ciclo de vida perteneciente al software. En ese sentido, se listan como estaciones del modelo, las siguientes:

1. Conversemos: Adoptando una propiedad comunicativa, y siendo su producto esperado el modelado del negocio, la etapa inicial promueve la concepción fundamental del escenario generado a partir de la interacción entre el cliente y el equipo de alto desempeño. Se consideran los componentes:

1.1. Involucrados: Identificación inicial de los participantes del proyecto, quiénes comandan el equipo de alto desempeño, o bien se ven afectados por el desarrollo del negocio. Se definen los medios de retroalimentación y rangos funcionales.

1.2. Visión del negocio: Se determinan las necesidades del proceso, lineamientos de colaboración, alcances y proyecciones.

1.3. Visión del software: Basada en la selección del domino de software apropiado para el acompañamiento del proceso general. Se indica el tipo de producto recomendado para el escenario, de los siguientes: (a) Software de sistemas, (b) Software de aplicación, (c) Software

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

de ingeniería y ciencias, (d) Software incrustado, (e) Software de línea de productos, (f) Aplicación Web, y (g) Software de inteligencia artificial.

1.4. Requerimientos: Se analiza el conjunto de características del producto, desde un enfoque tanto funcional, como no funcional, a fin de propiciar una aproximación cercana a la arquitectura de comportamiento general que tendrá el sistema potencial.

2. Prepara el trabajo: Promueve una propiedad proyectiva, y mantiene como producto esperado el mapa del proyecto, por lo que la organicidad de los recursos generales del proyecto respalda la ejecución de la segunda etapa. Así, su aplicación práctica respeta el seguimiento de cuatro componentes:

2.1. Tareas técnicas: Se señala el conjunto de prácticas o métodos a llevar a cabo como solución a la necesidad. Se indaga en torno al reconocimiento de patrones observables en la solución potencial. Posibilidad de inserción de problemáticas concatenadas o anidadas.

2.2. Riesgos probables: Se lista el conjunto de riesgos adosados al proceso de análisis, diseño, desarrollo e implementación del software y sus posibles planes de contingencia.

2.3. Recursos: Comprende la definición de los recursos técnicos, operativos, organizaciones y económicos del proyecto. Plantea la negociación final del proceso asumido entre el equipo de alto desempeño y el cliente.

2.4. Calendario: Determina la esquematización de actividades en función del tiempo. Inserta la entrega puntual de incrementos del proyecto, hasta llegar a su versión de despacho y despliegue.

3. Imagina el potencial: Comprendiendo una propiedad creativa, propone como producto esperado el diseño de la solución, así, la arquitectura funcional del software será objeto de atención durante el proceso perseguido en la etapa en curso, a razón de cuatro componentes:

3.1. Traducción de requerimientos: Refleja la asimilación de las necesidades del usuario con respecto al sistema. Selección de las mejores rutas para identificar las entidades con las que interactuará el sistema.

3.2. Notación UML: Se escoge el conjunto de diagramas de respaldo de la arquitectura funcional del software, con base en la notación internacional UML (Unified Modeling Language – Lenguaje de modelamiento unificado). Su producción representa el artefacto demandado por el diseño del sistema.

3.3. Base de datos: Se construye el modelo de datos y posteriormente es seleccionado el DBMS (Data Base Management System – Sistema manejador de base de datos).

3.4. Interfaces gráficas: Genera la comunicación gráfica y virtual entre el software y el usuario potencial. Se imprimen criterios de usabilidad, estética, gestión de contenido, así como impacto visual, con base en las características generales de la solución. El equipo de alto desempeño escoge el software indicado para la producción de las interfaces.

4. Desarrolla la solución: Refleja una propiedad innovadora, trayendo a su vez como producto esperado, el software construido, lo que implica un proceso completo de codificación y pruebas para la generación del producto informático. La estación admite los componentes:

4.1. Verificación UML: Con la arquitectura diseñada, el ingeniero de software procede a realizar una comprobación funcional del conjunto de artefactos proporcionados por la estación anterior, con el propósito de generar una ruta posible para la programación del sistema.

4.2. Metodología de programación: Se construye un esquema de codificación a seguir. Previamente, se ha de identificar el paradigma de programación, de acuerdo al proyecto específico, y posteriormente el lenguaje sobre el cual se construirá el software.

4.3. Codificación: Se ejecuta el proceso de programación con base en la metodología y la documentación. Se realizan pruebas parciales de ejecución funcional y conexiones determinadas hacia la base de datos.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

4.4. Validación funcional: Se realiza el conjunto de pruebas, indicando tanto las condiciones normales de ejecución, como las de riesgo, con el objeto de garantizar niveles apropiados de seguridad, confiabilidad, usabilidad y mantenimiento del software desarrollado.

5. Asegura la calidad: Con una propiedad efectiva, trae como resultado, el despliegue del software-producto, incluyendo su despacho, implementación y validación por parte de los usuarios, a través de los componentes:

5.1. Despacho: Comprende la preparación del software para su entrega, incluyendo aspectos de marketing, manual o guía de usuario, así como las condiciones mínimas requeridas para su implementación.

5.2. Implementación: Indica la preparación de las estaciones de trabajo para la inserción del sistema, así las mismas se adecuan con base en las condiciones mínimas para lograr una efectiva instalación del software. Adicionalmente, se permite el adiestramiento requerido por los usuarios finales en relación al conjunto de funciones que el sistema ofrece. En último término, se refleja un conjunto de casos de prueba, a fin de lograr una validación funcional del producto por los clientes.

5.3. Configuración de cambios: Ante la necesidad de modificar, ampliar o reducir algún componente del software, tras la ejecución de los casos de prueba. Es posible, la readecuación del producto en sus etapas de codificación y diseño, tras la inadecuada traducción de requerimientos o artefactos del sistema. El equipo, una vez realice las ediciones pertinentes, repite las etapas de despacho e implementación hasta tanto se hayan acordado las condiciones óptimas de entrega.

5.4. FeedBack: Se establece como resultado final del proceso de despacho e implementación. El cliente y equipo desarrollador fijan acuerdos de cierre, tras la satisfacción de cada requerimiento, respondiendo a la premisa constitutiva del modelo, la garantía de la calidad del proceso informático, transformando el software-producto en ingeniería.

Con la descripción de la última fase, se estableció la compresión global de InsideOut, atendiendo los diferentes escenarios suscitados dentro del desarrollo informático, respondiendo en agilidad y adaptabilidad en alineación con el proceso.

## Conclusiones

El análisis del sector de tecnologías de información en empresas desarrolladoras de sistemas informáticos reveló una notable carencia de parámetros de calidad que sirvan como garantía funcional de los productos generados. Mediante un diagnóstico basado en un enfoque investigativo, se identificó una aplicación limitada de patrones y métodos eficaces, apoyados en los principios fundamentales de la ingeniería de software, para asegurar la calidad de los procesos. Esta constatación impulsó la necesidad de realizar un análisis profundo para identificar los factores clave en la producción informática, lo que llevó a la definición de los patrones de calidad predominantes en la industria. Además, la conexión directa con el entorno de estudio ofreció una base empírica sólida para confrontar la teoría con la realidad práctica, reforzando la validez del análisis realizado.

La identificación de los parámetros de calidad en la producción informática permitió determinar las prácticas predominantes en la industria. Se evidenció la influencia de metodologías tradicionales utilizadas por los equipos desarrolladores, lo que reveló un perfil profesional marcado por desconfianza, imprecisiones en la evaluación de la calidad, cálculo de costos y dimensionamiento de resultados, junto con una escasa adherencia a los estándares y patrones tecnológicos establecidos. Estos hallazgos destacaron la necesidad de desarrollar un modelo que acompañe y direccione el entorno laboral, fundamentado en los principios esenciales de la ingeniería de software, para garantizar prácticas más precisas y consistentes.

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

Como resultado de este análisis, se concluyó que la evolución del oscurantismo en la práctica de la ingeniería de software ha experimentado un cambio significativo, impulsado por el fortalecimiento de nuevos esquemas de trabajo que abogan por la construcción de escenarios legítimos para la aplicación efectiva de la disciplina. En este contexto, la consolidación del trabajo del equipo interdisciplinario permitió desarrollar una perspectiva más clara sobre qué, cuándo y cómo deben llevarse a cabo los procesos de desarrollo de software.

Tomando en cuenta que, la definición del software no ha representado, para los diferentes especialistas de la industria, una sencilla labor, se hizo preciso reafirmar que el término ha comprendido, sin duda, un proceso productivo estructurado por instrucciones, procedimientos, programas, reglas y documentación. Siendo el producto caracterizado por los elementos anteriores, se requirió un nuevo enfoque organicista y cohesivo, respetando los patrones que dirigen la disciplina, e integrando la orientación ágil que demanda el proceso del software actual. De este modo, se reafirmó la funcionalidad, trazabilidad y usabilidad de la solución, considerándose una opción confiable para la industria en atención a la transformación del software en ingeniería.

De este modo, el modelo InsideOut se presentó como una solución integral y versátil para el aseguramiento de la calidad en el desarrollo informático, destacando por su enfoque sinérgico, práctico, especializado y orientado a la producción continua. Su capacidad para fomentar entornos colaborativos, integrar métodos prescriptivos y ágiles, y garantizar rutas óptimas para cada proyecto fortaleciendo la eficiencia y adaptabilidad de los equipos.
Al mismo tiempo, el modelo promovió altos estándares de calidad alineados con normativas internacionales como el ISO/IEC 25000:2014, asegurando productos confiables y funcionales. Asimismo, su estructura iterativa permitió organizar incrementos parciales durante el ciclo de vida del software, generando avances documentados y medibles.

En definitiva, InsideOut demostró un potencial de aplicabilidad en diversos sectores profesionales, educación, banca, banca, salud, manufactura y servicios, adaptándose a contextos cambiantes y necesidades específicas. Este enfoque flexible lo posiciona como un marco metodológico replicable y escalable, capaz de transformar y optimizar procesos informáticos en la industria, combinando especialización, calidad y practicidad, ofreciendo un modelo robusto que impulsa la mejora continua y establece un referente para la excelencia en el desarrollo de software.

## Referencias

Blandon, C., Jaramillo, J. (2023). Calidad del software y seguridad de aplicaciones a partir del proceso de desarrollo de software AGILISO y el estándar OWASP. V Congreso Internacional en Inteligencia Ambiental, Ingeniería de Software, Salud Electrónica y Móvil. Tecnológico de Costa Rica. Cartago, Costa Rica DOI: https://doi.org/10.18845/tm.v36i8.6923

Caicedo, F. (2023). Mejora de la calidad del software a través de la integración y entrega continua. ODIGOS. Volumen 4, No. 2, Págs. 45-55. DOI: https://doi.org/10.35290/ro.v4n2.2023.899

Carrizo, D., Alfaro, A. (2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. Revista chilena de ingeniería. Volumen 26, No. 1. DOI: http://dx.doi.org/10.4067/S0718-33052018000100114

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025

Díaz, R., Somaza, J. (2020). Diseño de un sistema informático para el diagnóstico y planes de tratamiento de armonización dento-facial. Trabajo de Grado de Odontología. Universidad José Antonio Páez (UJAP). San Diego, Venezuela.

Flores, J., Robles, E., Arizpe, V. (2018). Técnicas de Ingeniería de Software aplicadas a la acreditación de un programa educativo de Educación Superior. CONAIC. Volumen 5, No. 3, Págs. 6-16. DOI: https://doi.org/10.32671/terc.v5i3.56

Guevara, C., Pulla, C., Minchala, W., Moscoso, S. (2024). Propuesta de procesos misionales basados en transformación digital: impacto en la cadena de suministros académica. Gestión I+D. Volumen 9, No. 1. Disponible en: http://saber.ucv.ve/ojs/index.php/rev_GID/article/view/27635

Ibarra, S., Muñoz, M. (2018). Support tool for software quality assurance in software development. 7th International Conference On Software Process Improvement (CIMPS). Guadalajara, México. DOI: https://doi.org/10.1109/CIMPS.2018.8625617

ISO (2020). ISO/IEC 25000:2014. Documento en línea. Disponible en: https://www.iso.org/es/contents/data/standard/06/47/64764.html. Consulta: 22/11/2024.

Maquera, H., Ataucusi, O., Vega, J., Delgado, M. (2016). Métricas para el desarrollo de software para el control administrativo de proyectos en la oficina general de extensión universitaria y proyección social de la UNCP. Prospectiva universitaria en ingeniería y tecnología. Volumen 13, No. 1, Págs. 76-79. DOI: https://doi.org/10.26490/uncp.prospectivauniversitaria.2016.13.386

Muñoz, M. (2019). Disminuyendo la Brecha del Aseguramiento de la Calidad entre Gestor y Miembros del Equipo Mediante el Uso de Herramientas de Software. RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação. No. 31. DOI: https://doi.org/10.17013/risti.31.18-32

Peláez, L., Toro, A., Arias, J., Rodríguez, D. (2018). El aseguramiento de la calidad de los requisitos en la industria del software en el eje cafetero colombiano. INGE CUC. Volumen 15, No. 2, Págs. 110-122. DOI: http://doi.org/10.17981/ingecuc.15.2.2019.11

Pin, L., Toala, M., Ílava, J. (2020). La inteligencia artificial en la calidad del software: una revisión sistemática de la literatura. UNESUM-CIENCIAS Revista Científica Multidisciplinaria. Volumen 4, No. 1, Págs. 75-86. DOI: https://doi.org/10.47230/unesum-ciencias.v4.n1.2020.208

Pressman, R., Maxim, B. (2021). Ingeniería del software. Un enfoque práctico. 9na. Edición. Editorial McGraw Hill, México.

Serna, E. (2021). Métodos formales, ingeniería de requisitos y pruebas del software. 1ra. Edición. Editorial Instituto Antioqueño de Investigación, Colombia.

Sinisterra, M., Quimis, N., González, Y., Merino, J. (2021). Calidad del software aplicado a la evaluación de la seguridad en sistemas de información. UNESUM-Ciencias: Revista Científica Multidisciplinaria. Volumen 5, No. 4, Págs. 23-30. DOI: https://doi.org/10.47230/unesum-ciencias.v5.n4.2021.411

Depósito Legal: PPX200002ZU2142 / ISSN: 1856-4194. Volumen 24 Edición No 01– Año 2025