

**Tendencias investigativas sobre calidad de software en etapas tempranas: un análisis
bibliométrico**

Maria Isabel Ramirez Ramirez

Asesor

Jhon Fernando Sanchez Alvarez

Universidad Nacional Abierta y a Distancia UNAD
Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI
Ingeniería de Sistemas

2025

Nota de Aceptación

Nombre Director de Trabajo de Grado

Jurado

Jurado

Dedicatoria

Para mi papi, mi mami, mis hermanos, mis sobrinos, mi negrito y mi pecesuela, que siempre creyeron en mi mucho más de lo que yo lo hacía; porque ser fuerte y valiente es más fácil cuando están a mi lado.

Agradecimientos

A Dios, por ser fuente inagotable de amor, por ser luz en mi camino, por darme fuerza y hacerme un llamado cuando sentía desfallecer, por siempre poner en mi camino lo que necesito mucho más allá de lo que quiero; por sostener mis sueños cuando me sentía débil y por hacerme merecedora de cada triunfo y logro cumplido. Para ti padre amado, de la niña de tus ojos.

A mis padres Pedro Ramírez y Lucia Ramírez, nunca habrá palabras suficientes para expresar lo que siento en mi corazón; gracias por hacer de mi lo que soy hoy día, por sus constantes esfuerzos, sacrificios y amor incondicional, porque sin importar las dificultades y cuanto costara siempre hicieron suyos nuestros sueños, de niña me costaba mucho entender sus decisiones pero hoy lo valen todo, gracias por guiar mi camino, por mostrarme que puedo luchar y trabajar por mis sueños, que puedo con todo porque soy fuerte y valiente y porque esto que soy no es más que el reflejo de ustedes en mi. Espero que se sientan tan afortunados de ser mis padres como me siento yo de ser su hija.

A mis hermanos Angela, Claudia, Diana, Carlos y Caro (Nuestro Ángel German), por ser mi fuente de inspiración, de crecimiento personal y profesional día a día, por creer en mí, por sostenerme en mis fracasos y por aplaudirme y celebrar mis logros. Son mi mayor regalo y mi gran fortuna.

A Nana un agradecimiento especial por ayudarme a hacer este proyecto posible, por materializar este sueño con sus ideas y profesionalismo, gracias por acompañarme en cada etapa de mi estudio profesional, por ser mi maestra desde la escuela hasta mi fase universitaria, por levantarme cuando estaba cayendo y traerme hasta la meta.

A mis sobrinos Juan Pablo, Dahiana, Lorena, Laura, Samuel, Maria Camila, Maria Victoria y Emiliano, por ser el motor de mi vida y empujarme a ser mejor cada día para ustedes; nunca creí querer ser ejemplo para alguien hasta que llegaron a mi vida; quiero que mis sueños sean inspiración para lograr los suyos y que sepan que siempre voy a estar aquí para apoyar y celebrar cada uno de sus éxitos y que en los momentos difíciles siempre podrán contar conmigo.

A mi negrito y compañero de vida Julian Yotagri, por ser inspiración en mi vida, por tu paciencia, entrega y amor incondicional, gracias por ayudarme a reconstruir la mujer que soy, por motivarme a ser mejor cada día, por abrir mis alas, mi mente y mi ser para seguir soñando en grande. Por brindarme seguridad, apoyo y amor en este proceso y por ver con orgullo mis sueños futuros y los ya materializados. Este logro es Nuestro.

A mi pecesuela Sandra Avendaño, por mostrarme que no hay que llevar la misma sangre para sentir y valorar como propios los sueños de nuestras amigas; gracias por ser inspiración de resiliencia y fortaleza en mi vida. Por empujarme a ser mejor y creer en mis sueños.

A mi tutor Jhon Fernando Sanchez, por su invaluable apoyo y orientación a lo largo de este proceso, su experiencia, profesionalismo y dedicación fueron fundamentales para el desarrollo de esta tesis. Agradezco su motivación, críticas constructivas, las cuales me impulsaron a superarme y lograrlo.

A mis maestros, todos aquellos que pasaron por mi vida y dejaron una huella imborrable, aquellos que hoy cruzo en mi camino y me ven con ojos de orgullo y amor; a ellos gracias, porque sus consejos, correcciones y amor ayudaron a forjar mi ser; aquí una mención especial a Carlos Mario Molina, el respeto y admiración que siento por ti son mucho más de lo que podría describir, sin tu guía me hubiese tomado más tiempo y trabajo tomar las decisiones correctas. Tus consejos y amor hoy me tienen aquí.

A mis amigos y compañeros, gracias por ser parte de este camino, por sus consejos, amabilidad y apoyo constante.

Finalmente pero no menos importante, a mi misma, por el esfuerzo, dedicación y motivación que he tenido para enfrentar cada reto y desafío; por tener la convicción de lograrlo cuando sentía que constaba más, porque hoy después de 7 años de ejercer mi trabajo como profesional logré este sueño. Y porque esta es la muestra de que podré seguir logrando mis objetivos con determinación y amor por lo que hago.

Resumen

La calidad del software en las primeras etapas es crucial para mejorar la eficiencia y satisfacción en las organizaciones, es esencial que el software cumpla con criterios de calidad para garantizar su óptimo funcionamiento; aunque los estudios sobre este tema son limitados, la importancia de invertir en investigaciones para mejorar la calidad del software es evidente, especialmente ante el constante aumento de fallas. El objetivo de esta investigación es explorar las tendencias investigativas alrededor de la calidad del software en una etapa temprana. La metodología abarca un análisis bibliométrico regido por la declaración PRISMA-2020 en las bases de datos Scopus y Web Of Science. Los resultados revelaron un crecimiento polinomial cuadrático en las publicaciones de 99,12% con los años 2018 y 2022 como los más relevantes. En los referentes investigativos se tienen los autores Malhotra y Smidts; la revista Journal Of Information Processing Systems y los países India y Estados Unidos. La evolución temática paso de investigar sobre Software Product Testing a Decision Analytics y Reliability Performance. Los clústeres temáticos destacan conceptos como Software metric, fuzzy logic, machine learning y software Testing. Las principales palabras clave emergentes son Project Management, software Engineering y el concepto creciente de Fuzzy logic.

Palabras clave: Pruebas, confiabilidad, software, gestión, calidad.

Abstract

The quality of the software in the early stages is crucial to improve efficiency and satisfaction in organizations, it is essential that the software meets quality criteria to guarantee its optimal functioning; Although studies on this topic are limited, the importance of investing in research to improve software quality is evident, especially in the face of the constant increase in failures. The aim of this research is to explore research trends around software quality at an early stage. The methodology includes a bibliometric analysis governed by the PRISMA-2020 declaration in the Scopus and Web Of Science databases. The results revealed a quadratic polynomial growth in publications of 99.12% with the years 2018 and 2022 as the most relevant. The authors Malhotra and Smidts are among the research references; the Journal Of Information Processing Systems and the countries of India and the United States. The thematic evolution went from researching Software Product Testing to Decision Analytics and Reliability Performance. The thematic clusters highlight concepts such as Software metric, fuzzy logic, machine learning and software testing. The main emerging keywords are Project Management, Software Engineering and the growing concept of Fuzzy logic.

Keywords: Testing, reliability, software, management, quality.

Tabla de Contenido

Introducción	12
Justificación	19
Objetivos.....	21
Objetivo General.....	21
Objetivos Específicos	21
Marco Conceptual.....	22
Metodología	24
Criterios de Elegibilidad.....	24
Fuente de Información.....	25
Estrategia de Búsqueda.....	26
Gestión de Datos.....	28
Proceso de Selección	28
Resultados.....	31
Discusión.....	58
Análisis de Resultados.....	58
Comparación Con Otros Estudios	62
Implicaciones Teóricas	68
Implicaciones Prácticas	69
Impacto Industrial de las Tendencias Emergentes.....	72
Desafíos y Oportunidad Emergentes	74
Limitaciones	76
Futuras Investigaciones	78

Conclusiones	81
Referencias Bibliográficas	84

Lista de Figuras

Figura 1 <i>Diagrama de flujo PRISMA</i>	30
Figura 2 <i>Publicaciones por año</i>	32
Figura 3 <i>Principales autores</i>	35
Figura 4 <i>Principales revistas</i>	38
Figura 5 <i>Principales países</i>	41
Figura 6 <i>Evolución temática</i>	44
Figura 7 <i>Red de coocurrencia de palabras clave</i>	48
Figura 8 <i>Vigencia y frecuencia de las palabras clave</i>	52

Introducción

La calidad del software en etapas tempranas es una de las herramientas más útiles en la optimización del software en las organizaciones, con la finalidad de contar y ofrecer optimización, eficiencia y satisfacción de necesidades, es por ello que el software debe contar con criterios que garanticen su calidad (Saleh, H AL_Bayati & Hadi Thanoon 2020). El término calidad del software indica el grado de rendimiento de las características originales a las que debe responder un sistema informático durante su ciclo de vida, estas características deben garantizar un sistema fiable para el cliente, que aumente su satisfacción con la eficacia y la funcionalidad del sistema (Nakahara, Monden & Yücel,2021).

La evaluación arquitectónica del software resulta eficiente, puesto que posibilita la detección y corrección de errores de diseño en etapas tempranas del ciclo de desarrollo. De acuerdo con la norma ISO 9126-1, la calidad del software se clasifica en factores como funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad, cada uno con características específicas que influyen en su desempeño y adaptabilidad (Pargaonkar, 2021; Dayanandan & Kalimuthu, 2018).

En el contexto contemporáneo, el software se erige como un componente de suma importancia en la vida real, y su calidad se revela como un factor determinante para su durabilidad y fiabilidad (Sahu & Srivastava, 2018). No obstante, numerosos desarrolladores optan por desviarse de los procesos establecidos sin la debida supervisión, lo que repercute negativamente en los resultados del proyecto (Nakahara, Monden & Yücel, 2021). La evaluación de la calidad en etapas tempranas resulta esencial para la reducción de costes y el incremento de la eficacia del desarrollo. Métodos como la lógica difusa permiten gestionar la incertidumbre y

mejorar la evaluación de errores y la confiabilidad, con aplicaciones que trascienden el ámbito del software, como en procesos industriales (Gupta & Gupta, 2024; Ángeles Medina, 2022).

La lógica difusa se erige como un complemento de las métricas tradicionales de calidad, otorgando prioridad a las acciones clave en las fases iniciales del desarrollo, particularmente en contextos de incertidumbre. Gupta y Gupta (2024) demostraron que, al combinarse con algoritmos evolutivos, optimiza la asignación de errores y mejora la eficiencia del proceso. Estas herramientas emergentes fortalecen la confiabilidad de las evaluaciones y amplían las posibilidades en el aseguramiento de calidad desde las etapas iniciales.

El modelo de proceso de software ejerce una influencia en la calidad del sistema, dado que la detección tardía de un defecto conlleva una mayor dificultad para su corrección. La capacidad de predecir la calidad en etapas iniciales facilita la implementación de estrategias de mantenimiento y la asignación eficiente de recursos (Reddivari & Raman, 2019). El presente estudio tiene como objetivo analizar cómo el éxito de un software depende de su calidad (Cowlessur, Pattnaik & Pattanayak, 2020), y cómo la detección temprana de defectos ahorra tiempo y costes, aumentando su confiabilidad (Rai et al., 2017).

Los gerentes de proyectos se enfrentan al desafío de desarrollar software de calidad bajo restricciones de tiempo y presupuesto. El modelado predictivo, por su parte, emerge como una herramienta de gran utilidad al permitir la estimación de atributos como defectos, mantenibilidad y esfuerzo, mediante el uso de métricas como acoplamiento, cohesión, herencia y polimorfismo (Malhotra, 2017). La confiabilidad del software se evalúa mediante la observación de fallas en las pruebas, aunque los modelos basados en el tiempo no son adecuados para entornos dinámicos (Sudharson & Prabha, 2019).

La confiabilidad del software se define como la probabilidad de que el sistema no experimente fallos en un tiempo determinado. Los modelos de crecimiento de confiabilidad del software (SRGM, por sus siglas en inglés) predicen su comportamiento en función de fallos previos y contribuyen a determinar el momento óptimo de lanzamiento y el esfuerzo de prueba requerido (Jabeen & Afzal, 2019). Desde la década de 1970, se han desarrollado numerosos SRGM para estimar la confiabilidad a lo largo del proceso de prueba (Asraful Haque & Ahmad, 2021).

La comprensión de los atributos de calidad se erige como un componente esencial para la optimización de la confiabilidad del software. La evaluación de métricas en etapas tempranas permite la identificación oportuna de diseños deficientes y su posterior corrección, lo que se traduce en una reducción de costes y una optimización del producto final (Jain, 2012). Durante el proceso de desarrollo, las organizaciones deben tomar decisiones estratégicas sobre la asignación de recursos para la realización de pruebas y la fecha de lanzamiento del software. Los SRGM contribuyen a la optimización de estas decisiones al prever con exactitud defectos y la calidad esperada (Rana et al., 2014).

Los SRGM implementan la ingeniería de confiabilidad en el ámbito del software, mediante la utilización de herramientas matemáticas para la modelización de defectos y la determinación del momento oportuno para la finalización de las pruebas de software, garantizando así su confiabilidad (Dalal & Mallows, 1988; Rana et al., 2014). No obstante, garantizar la confiabilidad en sistemas cada vez más complejos constituye un desafío, puesto que la mayoría de los modelos asumen que las fallas detectadas se corrigen de inmediato, lo cual no siempre resulta realista (Wang, Hu & Liu, 2016).

La predicción temprana de la confiabilidad permite evaluar la viabilidad de los requisitos propuestos y fundamentar las decisiones de diseño. Sin embargo, los métodos actuales se fundamentan en estimaciones analíticas que no siempre proporcionan cifras exactas (Singh et al., 2016). En este sentido, la relevancia de la calidad del software se incrementa progresivamente, si bien muchas de las técnicas empleadas se centran en la corrección de defectos tardíos en lugar de prevenirlos desde las primeras etapas del desarrollo. Las pruebas finales se implementan en etapas avanzadas del desarrollo, lo que limita su capacidad para influir en la mayoría de los atributos de calidad (Shepperd, 1990).

El incremento en el número de usuarios con conocimientos especializados en tecnología ha propiciado un aumento en la demanda de software dotado de funciones avanzadas. Esta circunstancia ha obligado a las empresas a asegurar su confiabilidad incluso después de su lanzamiento (Shrivastava, Sharma & Pham, 2023). La confiabilidad del software constituye un atributo de calidad fundamental, no obstante, su predicción tiende a posponerse a etapas avanzadas del desarrollo, cuando la implementación de mejoras resulta más onerosa (Rizvi, Singh & Khan, 2016). A pesar de que entre el 70 % y el 80 % de las fallas surgen en la fase de requisitos, la mayoría de los modelos existentes evalúan la confiabilidad solo en etapas avanzadas (Mohanta, Vinod & Mall, 2011).

Para mejorar la confiabilidad del software, se requiere el desarrollo de métodos que automaticen la predicción desde las primeras fases del diseño, lo que reduciría costes y esfuerzo (D'Ambrogio, Iazeolla & Mirandola, 2002). La fase de diseño, que resulta crucial para determinar la calidad, presenta aún vacíos en la literatura sobre la influencia de las interdependencias y las métricas de diseño (Nair & Selvarani, 2012). A pesar de las propuestas de modelos de predicción temprana en la década de 1990, aún persisten interrogantes sin

resolver, lo que subraya la necesidad de un enfoque más exhaustivo (Smidts, Stutzke & Stoddard, 1998).

Un desafío adicional radica en la suposición de que las fallas provienen exclusivamente del software, sin considerar la interacción con el hardware, lo que puede generar estimaciones inexactas (Park et al., 2012; Lee, Chang & Pham, 2020). Métodos como la Ingeniería de Software Basada en Componentes y la computación blanda han mostrado potencial para mejorar la confiabilidad en entornos inciertos (Diwaker et al., 2018). No obstante, la mayoría de los estudios recientes se han centrado en la predicción de la calidad mediante el aprendizaje automático (Cowlessur, Pattnaik & Pattanayak, 2020), mientras que el creciente número de fallas ha impulsado una mayor inversión en la mejora de la fiabilidad (Yang, 2022).

La ausencia de metodologías eficaces para evaluar la calidad en las etapas iniciales del desarrollo conlleva consecuencias significativas. Según Standish Group (2020), el 66 % de los proyectos de software experimentan fracaso parcial o total debido a la detección tardía de defectos. La corrección de errores en etapas avanzadas puede llegar a ser hasta 100 veces más costosa que en fases tempranas (Boehm & Basili, 2001). En sectores críticos como la salud y la banca, estas deficiencias pueden acarrear consecuencias graves, tanto económicas como de seguridad.

Desde una perspectiva académica, si bien se han desarrollado métricas y modelos predictivos, aún no existe un consenso unánime sobre las metodologías más efectivas en entornos industriales específicos. Además, se ha observado un número limitado de estudios centrados en pruebas posteriores al lanzamiento con garantía, lo que representa una brecha en la literatura (Shrivastava, Sharma & Pham, 2023). En este sentido, el objetivo de la presente investigación es abordar dicha laguna de conocimiento mediante un análisis bibliométrico que

permita identificar tendencias, clústeres temáticos y enfoques emergentes en el ámbito de la evaluación temprana de la calidad del software. Este análisis contribuirá al desarrollo de mejores prácticas en este campo.

En consecuencia a los vacíos identificados el objetivo de esta investigación es explorar las tendencias investigativas alrededor de la calidad del software en una etapa temprana; la contribución de esta investigación radica en la relevancia y actualidad del tema abordado, la calidad del software en las primeras etapas de desarrollo es crucial para garantizar la confiabilidad y eficiencia de los productos finales, y permite ampliar el conocimiento existente en este campo de gran importancia para la industria de la tecnología. Este estudio proporciona un análisis cuantitativo que facilita la identificación de tendencias, autores clave y áreas de investigación emergentes en este ámbito, lo cual será de gran utilidad para la comunidad académica y profesional interesada en este tema. Para ello, se plantean las siguientes preguntas de investigación:

¿Cuáles son los años donde más interés se ha presentado sobre la calidad del software en etapas tempranas?

¿Qué tipo de crecimiento presenta la cantidad de artículos científicos sobre la calidad del software en etapas tempranas?

¿Cuáles son los principales referentes investigativos sobre la calidad del software en etapas tempranas?

¿Cuál es la evolución temática derivada de la producción científica sobre la calidad del software en etapas tempranas?

¿Cuáles son los principales clústeres temáticos sobre la calidad del software en etapas tempranas?

¿Cuáles son las palabras clave crecientes y emergentes en el campo de investigación de la calidad del software en etapas tempranas?

Este artículo se estructura en seis secciones que comprenden el resumen, la introducción, la metodología, los resultados, la discusión y las conclusiones. En el resumen, se presenta una visión general del estudio, destacando el objetivo principal, la metodología empleada, los hallazgos más relevantes y las conclusiones fundamentales. La introducción contextualiza la investigación, proporcionando el trasfondo teórico y la motivación del estudio, además de definir el problema de investigación y los objetivos específicos. La metodología detalla el diseño del estudio, los criterios de selección de datos, las bases de datos consultadas y los procedimientos analíticos aplicados. En los resultados exponen los hallazgos obtenidos del análisis de los datos, presentando estadísticas descriptivas, gráficas y tendencias identificadas. En la discusión, se realiza una comparación con otros artículos relacionados y se contemplan las implicaciones prácticas, teóricas y posibles limitaciones. Finalmente, en las conclusiones se resumen los principales hallazgos, discuten su relevancia y sugieren posibles direcciones para futuras investigaciones.

Justificación

Relevancia del tema: La calidad del software en las etapas tempranas del desarrollo es crucial para garantizar el éxito de un proyecto a lo largo de su ciclo de vida. Las decisiones tomadas en estas fases iniciales tienen un impacto significativo en el rendimiento, la fiabilidad y la sostenibilidad del software. Detectar y corregir errores en estas etapas no solo reduce los costos de desarrollo, sino que también mejora la eficiencia y la satisfacción del usuario final. Este enfoque proactivo asegura que el software cumpla con los estándares de calidad desde el inicio, evitando problemas costosos y complicados de corregir en etapas posteriores.

Beneficios del análisis bibliométrico: El análisis bibliométrico es una metodología poderosa para examinar la evolución de la investigación en torno a la calidad del software en etapas tempranas. Este tipo de análisis permite identificar tendencias, vacíos en la literatura y áreas emergentes que requieren atención. Al aplicar técnicas bibliométricas, es posible mapear el desarrollo del campo, identificar los principales actores y contribuir a la creación de un cuerpo de conocimiento más coherente y estructurado. Este enfoque ofrece una visión cuantitativa y objetiva de la investigación, lo que lo convierte en una herramienta valiosa para cualquier estudio académico riguroso.

Impacto en la comunidad académica y en la industria: Los resultados de esta investigación tienen el potencial de beneficiar tanto a la comunidad académica como a la industria del software. Para los investigadores, el estudio proporciona una base sólida para futuras investigaciones, destacando las áreas que requieren mayor exploración. Para los profesionales de la industria, los hallazgos pueden servir como guía para mejorar las prácticas de desarrollo de software, optimizando la asignación de recursos y mejorando la calidad del producto desde las fases más tempranas del ciclo de vida. Esto no solo incrementa la

confiabilidad del software, sino que también contribuye a una mejor planificación y ejecución de proyectos.

Innovación y contribución al campo: La originalidad de esta investigación radica en su enfoque en la calidad del software durante las fases iniciales de desarrollo, un aspecto que, aunque reconocido como crucial, ha recibido menos atención en la literatura. Al combinar un análisis bibliométrico con un enfoque en la predicción temprana de la calidad, esta monografía no solo contribuye al entendimiento actual del campo, sino que también propone nuevas perspectivas y enfoques para mejorar la calidad del software desde el principio. Esta investigación no solo llena un vacío existente, sino que también establece una base para futuros estudios que podrían llevar a avances significativos en la ingeniería de software

Objetivos

Objetivo General

Explorar las tendencias investigativas alrededor de la calidad del software en una etapa temprana de desarrollo.

Objetivos Específicos

Identificar los años de mayor interés sobre la calidad del software en etapas tempranas.

Identificar el tipo de crecimiento de la cantidad de artículos científicos sobre la calidad del software en etapas tempranas.

Identificar los principales referentes investigativos sobre la calidad del software en etapas tempranas.

Identificar la evolución temática derivada de la producción científica sobre la calidad del software en etapas tempranas.

Identificar los principales clústeres temáticos sobre la calidad del software en etapas tempranas.

Identificar las palabras clave crecientes y emergentes en el campo de investigación de la calidad del software.

Marco Conceptual

Las métricas de calidad del software permiten evaluar atributos clave como la mantenibilidad, la confiabilidad y la eficiencia. Estas métricas se clasifican en tres categorías principales: estructurales, funcionales y de proceso. Dentro de las métricas estructurales, la complejidad ciclomática, propuesta por McCabe (1976), constituye un indicador de la dificultad del código y su repercusión en la mantenibilidad. Las métricas funcionales, tales como los puntos de función, estiman el esfuerzo de desarrollo según los requisitos del usuario (Albrecht, 1979). Por último, las métricas de proceso, tales como la densidad de defectos, permiten evaluar la calidad del software a lo largo de su ciclo de vida (Kan, 2002). Su aplicación en fases tempranas ayuda a detectar problemas, reducir costes y mejorar la fiabilidad (Fenton & Pfleeger, 1997).

Se ha observado una evolución progresiva en los modelos predictivos de calidad, los cuales han experimentado una transición desde enfoques basados en estadísticas tradicionales y regresiones lineales (Briand et al., 1992) hasta modelos de aprendizaje automático y redes neuronales, que han demostrado una capacidad superior para predecir defectos y evaluar el rendimiento del software (Nam & Kim, 2015). Los modelos híbridos, que combinan técnicas estadísticas con inteligencia artificial, han demostrado una mayor eficacia en la identificación temprana de errores (Radjenović et al., 2013). Asimismo, el empleo de vastos conjuntos de datos y metodologías como el procesamiento de lenguaje natural ha optimizado la gestión de la calidad en las fases iniciales del desarrollo (Malhotra, 2015).

La lógica difusa ha experimentado una notable evolución en el ámbito de la evaluación de la calidad del software, permitiendo la gestión de incertidumbre y subjetividad en el proceso de toma de decisiones (Zadeh, 1965). En contraste con los modelos convencionales, facilita la evaluación preliminar cuando las métricas son incompletas o inexactas (Dubois & Prade, 1998).

Además, su aplicación permite la asignación de valores graduales a atributos como la confiabilidad y la mantenibilidad, lo que resulta en evaluaciones más flexibles y adaptativas (Bansal & Sharma, 2012). Además, su implementación en herramientas de aseguramiento de calidad ha reducido los errores no detectados en etapas iniciales, optimizando así las pruebas y validaciones (Mkaouer et al., 2016).

Metodología

Para dar cumplimiento al objetivo y a las preguntas de investigación planteadas, se propone un análisis bibliométrico el cual permite explorar, detectar y mostrar información relevante y significativa en gran volumen de documentos, cuyos resultados constituyen una fuente de información útil para evaluar la actividad científica (Solano et al, 2009). De esta manera aumentar las causas de minucia y repetibilidad del diseño metodológico el cual se realiza partiendo de los criterios descritos en la declaración PRISMA 2020 para revisiones de literatura por medio de la cual se definen criterios de elegibilidad, fuente de información, estrategia de búsqueda y gestión de los datos obtenidos de la ejecución de esta estrategia en la fuente de datos seleccionada (Page et al, 2021).

Criterios de Elegibilidad

Los criterios de inclusión se basaron en la evaluación de los principales metadatos, específicamente los títulos y palabras clave, debido a su relevancia y disponibilidad en bases de datos académicas. Para garantizar una amplia cobertura del tema, se empleó una combinación de términos clave como “calidad de software” y “etapas tempranas”, incluyendo sinónimos y términos relacionados en distintos contextos académicos. Se incluyeron artículos publicados en revistas indexadas en Scopus y Web of Science, asegurando que los documentos analizados estuvieran revisados por pares y cumplieran con estándares científicos rigurosos. Asimismo, se consideraron estudios sobre métricas de calidad, fiabilidad del software, modelos predictivos y lógica difusa.

En cuanto a los criterios de exclusión, se descartaron documentos que no estuvieran escritos en inglés o español, ya que estos idiomas concentran la mayor parte de la producción científica relevante en la disciplina. También se eliminaron estudios de conferencias con baja

visibilidad, tesis de grado y documentos técnicos de instituciones no académicas, dado que su rigor metodológico puede no ser equiparable al de las revistas indexadas. Finalmente, se excluyeron investigaciones que, aunque mencionaban la calidad del software, no proporcionaban un enfoque centrado en las etapas tempranas del desarrollo, asegurando así la pertinencia de los datos analizados.

El proceso de selección constó de tres fases de exclusión. En la primera fase, se eliminaron registros con indexación errónea para garantizar la precisión de los datos. En la segunda fase, se descartaron documentos sin acceso al texto completo, aunque esta restricción solo se aplicó a Revisiones Sistemáticas de Literatura. En la tercera fase, se excluyeron artículos con indexación incompleta, asegurando la integridad de la información analizada.

Fuente de Información

Se tomó la decisión de seleccionar las bases de datos Scopus y Web of Science debido a su reconocimiento como las fuentes más prestigiosas para estudios bibliométricos. Ambas bases de datos poseen criterios rigurosos de indexación, garantizando que las publicaciones incluidas han sido sometidas a procesos estrictos de revisión por pares, asegurando así su calidad y relevancia científica (Asubiaro & Onaolapo, 2023).

En el contexto de la investigación sobre la calidad del software en etapas tempranas, estas bases de datos ofrecen una cobertura multidisciplinar y actualizada, lo que permite identificar tendencias, autores influyentes y redes de colaboración en el área (Singh et al., 2021). Scopus es reconocida por su amplio alcance en literatura científica y técnica, mientras que Web of Science proporciona herramientas avanzadas de análisis de impacto y citación, lo que facilita una evaluación más precisa de la producción científica en el campo.

La elección de Scopus y Web of Science también se fundamenta en su compatibilidad con metodologías bibliométricas establecidas, tales como el análisis de coocurrencia de palabras clave y la identificación de clústeres temáticos. Además, ambas plataformas han sido utilizadas en estudios previos sobre la calidad del software, lo que permite la comparación de resultados y la replicabilidad del estudio.

Sin embargo, es preciso señalar que esta selección conlleva ciertas limitaciones. Al excluir bases de datos especializadas como IEEE Xplore o ACM Digital Library, se corre el riesgo de omitir estudios relevantes publicados en conferencias o revistas técnicas específicas del área. Además, dado que Scopus y Web of Science otorgan prioridad a las publicaciones de alto impacto, la literatura gris, como los informes técnicos y los documentos de trabajo, podría excluirse del análisis. A pesar de estas limitaciones, la robustez y el reconocimiento de estas bases de datos justifican su uso como fuente principal para el presente estudio.

Estrategia de Búsqueda

Para materializar la búsqueda en las bases de datos seleccionadas, se diseñaron ecuaciones de búsqueda especializadas con el objetivo de garantizar una recuperación de información relevante y exhaustiva. La formulación de estas ecuaciones se fundamentó en una combinación de términos clave y sinónimos estratégicos, alineados con los criterios de inclusión previamente definidos. Se consideraron términos que reflejan los conceptos fundamentales de la investigación, como software quality, quality assurance y software development, asegurando la inclusión de estudios relacionados con la evaluación de la calidad del software.

Dado que el objeto de estudio se centra en las etapas iniciales del desarrollo del software, se han incorporado términos específicos como early stages, early phases, early software quality,

early software development y early software testing. Esta selección responde a la necesidad de identificar literatura que aborde la evaluación de la calidad desde las primeras fases del ciclo de vida del software, aspecto crucial para la identificación temprana de defectos y la optimización de los procesos de aseguramiento de calidad.

Se implementaron operadores booleanos (OR y AND) para garantizar una combinación eficaz de los términos de búsqueda. El operador booleano OR amplía la cobertura de la consulta al incluir sinónimos y términos relacionados, mientras que el operador AND restringe la búsqueda a documentos que aborden simultáneamente los dos ejes temáticos principales: calidad del software y su evaluación en etapas tempranas.

En el ámbito de la investigación científica, se implementó un proceso de ajuste de las ecuaciones a las particularidades de cada base de datos, lo que resultó en una optimización de la precisión y la confiabilidad de los resultados. En el caso de Scopus, se empleó el campo Title para recuperar estudios en los que los términos clave aparecieran en el título, garantizando así una alta relevancia en los resultados obtenidos. En Web of Science, se empleó el campo TS (Topic Search), que permite buscar en títulos, resúmenes y palabras clave, ampliando la cobertura de la consulta. Esta adaptación metodológica optimiza la precisión y exhaustividad de la búsqueda en cada plataforma.

Para la base de datos Scopus: Title= ("software quality" OR "quality assurance" OR "software development" OR "early stages" OR "early phases") AND ("early software quality" OR "early software development" OR "early software testing")

Para la base de datos Web of Science: TS=("software quality" OR "quality assurance" OR "software development" OR "early stages" OR "early phases") AND TS=("early software quality" OR "early software development" OR "early software testing")

Gestión de Datos

Para el análisis bibliométrico, se utilizaron VOSviewer y Microsoft Excel con el fin de extraer, visualizar y procesar los datos de los artículos seleccionados. VOSviewer, desarrollado por Van Eck y Waltman (2010), fue empleado para construir mapas de redes de coocurrencia de palabras clave y colaboración entre autores, utilizando el algoritmo de agrupamiento basado en fuerza de atracción (force-directed layout). Se establecieron parámetros como el número mínimo de ocurrencias de una palabra clave (threshold) para garantizar la relevancia de los términos analizados, así como la normalización de asociación para mejorar la interpretación de los vínculos temáticos.

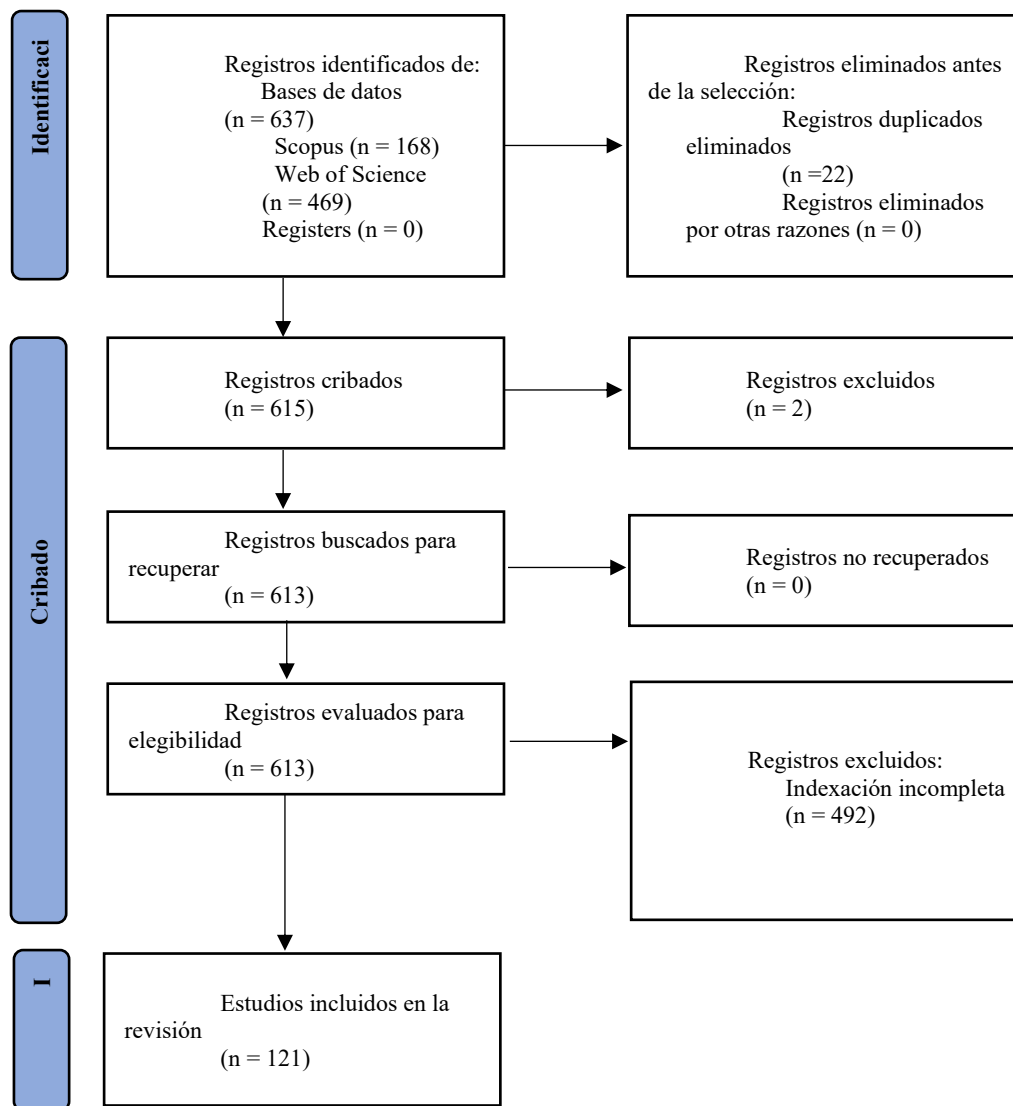
Por otro lado, Microsoft Excel fue utilizado para la extracción, almacenamiento y tratamiento de los datos obtenidos de cada base de datos, facilitando la organización y gestión eficiente de la información. Además, esta herramienta permitió la generación de gráficos descriptivos sobre la evolución temporal de las publicaciones y la distribución de artículos por país, proporcionando una representación clara de los indicadores bibliométricos. La combinación de VOSviewer y Microsoft Excel aseguró una plataforma robusta para el análisis y visualización de tendencias, mejorando la precisión en la interpretación de los resultados (Yusri, Edriati & Yuhendri, 2020).

Proceso de Selección

De acuerdo con la declaración PRISMA 2020, es fundamental mencionar si se utilizó un clasificador automático derivado internamente para asistir en el proceso de selección de estudios. Además, es importante describir la validación interna o externa de este clasificador para comprender el riesgo de estudios perdidos o clasificaciones incorrectas. En el presente estudio, se emplearon herramientas de automatización de Microsoft Excel® como soporte interno para

aplicar de forma independiente los criterios de inclusión y exclusión. Esto permitió reducir el riesgo de estudios perdidos o clasificaciones incorrectas gracias a la convergencia de los resultados obtenidos, asegurando una mayor precisión y consistencia en la selección de estudios relevantes para el análisis bibliométrico (Page et al., 2021).

Posteriormente, el análisis bibliométrico se llevó a cabo utilizando herramientas especializadas como VOSviewer, donde se procesaron los datos extraídos para generar mapas de redes de co-ocurrencia de palabras clave. Estos mapas fueron construidos mediante algoritmos de agrupamiento, como el método de clustering basado en fuerza de atracción (force-directed layout), que permitió identificar clústeres temáticos representativos. Las métricas bibliométricas, tales como la frecuencia de aparición de palabras clave, el grado de centralidad y la densidad de los nodos, fueron analizadas para interpretar la estructura temática y las relaciones entre conceptos clave. Este enfoque detallado permitió identificar patrones de investigación emergentes y consolidar los principales focos de atención en el campo, ofreciendo un análisis riguroso y fundamentado de las redes de co-ocurrencia.

Figura 1*Diagrama de flujo PRISMA*

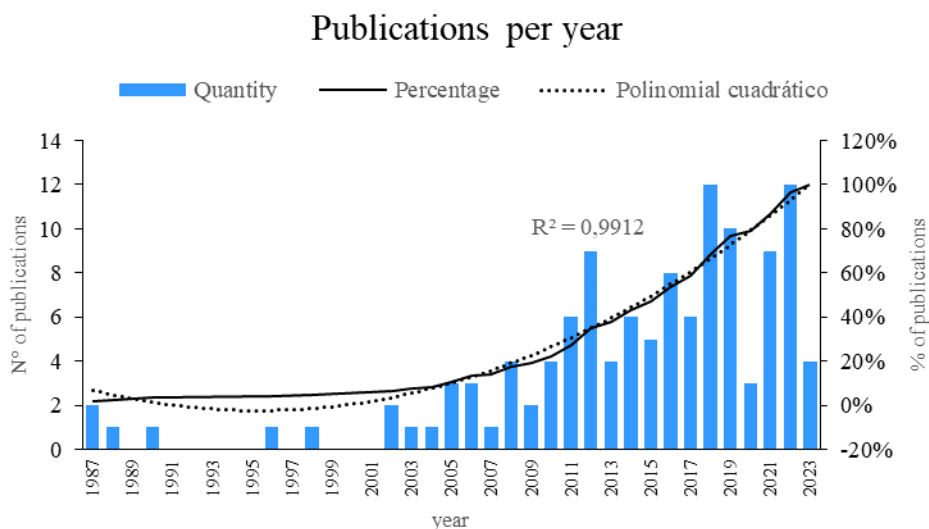
Nota. *Elaboración propia a partir de Scopus y Web of Science*

La parte inicial del estudio consistió en la identificación de registros, producto de la estrategia de búsqueda en cada fuente de información seleccionada, seguida de la eliminación de registros duplicados; posteriormente, se aplicaron las tres fases de exclusión previamente mencionadas; como resultado de este proceso, se seleccionaron finalmente 121 artículos que fueron incluidos en la presente bibliometría.

Resultados

En esta sección se exponen los resultados encontrados acorde con el objetivo propuesto, respondiendo a las preguntas de investigación, se puntualizan los aspectos bibliométricos del presente análisis relativos a la calidad del software en etapas tempranas de desarrollo, donde se examinan los componentes asociados a las tendencias investigativas considerando el tipo de crecimiento que presenta la cantidad de artículos científicos relacionados con la temática, los años donde se ha presentado más interés sobre la calidad del software en etapas tempranas y los principales referentes investigativos; más adelante se analizan la evolución temática, la red de coocurrencia y las palabras clave crecientes y emergentes en el campo de estudio.

El primer apartado a considerar son las publicaciones por año en el cual se puede observar a través de la figura 2, que las primeras publicaciones se dieron en el año 1987 y en una de ellas se habla de considerar cuidadosamente los paradigmas de la ingeniería de software y los efectos potenciales de su uso frente a la calidad del software (Perkins & Gorzela, 1987).

Figura 2*Publicaciones por año*

Nota. *Elaboración propia a partir de Scopus y Web of Science*

En el ámbito de la investigación científica, se ha observado un incremento notable en la producción académica relacionada con la calidad del software, particularmente en sus etapas iniciales. Este crecimiento, que presenta una tendencia polinomial cuadrática, alcanzó un 99,12 % entre los años 2022 y 2019. Además, los años 2022, 2018 y 2019 se distinguieron por su alta productividad, con investigaciones que han ejercido un impacto significativo en la industria del software al proporcionar modelos y enfoques innovadores para mejorar la fiabilidad y estabilidad del desarrollo de software.

En 2022, la investigación se centró en la mejora de la fiabilidad del software mediante modelos avanzados. Un estudio fundamental de este año desarrolló un modelo híbrido de regresión de soporte vectorial basado en la desviación cuartílica para conjuntos de datos de fiabilidad del software, lo que representó una contribución significativa al mejorar la precisión en la predicción de la fiabilidad (Reddy & Prasanth, 2022). Esta mejora tiene aplicaciones directas

en la industria del software, ya que permite a las empresas optimizar el mantenimiento predictivo y reducir los costes asociados a fallos en la producción. Además, se exploró el uso de modelos de métricas de estabilidad para evaluar la calidad del software, proporcionando herramientas clave para mejorar la resiliencia del software en entornos de producción, donde la estabilidad es crucial para garantizar una experiencia de usuario óptima y evitar interrupciones en servicios críticos (Dimri, Purohit & Dumka, 2022).

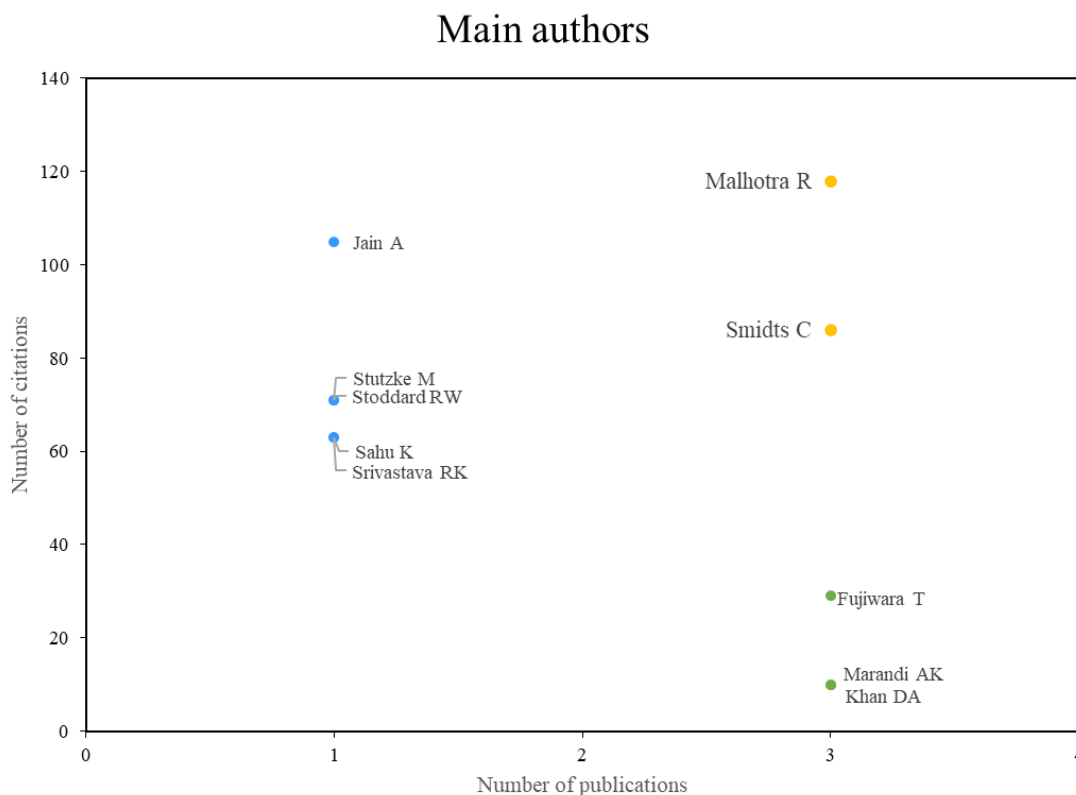
En el año 2018, los estudios se centraron en la teoría y la aplicación práctica de modelos de fiabilidad del software. Uno de los avances más notables fue el análisis de los dominios de no propagación y sus transformaciones, lo que permitió comprender cómo ciertas modificaciones en el código pueden afectar la fiabilidad del software (Mutha & Smidts, 2018). Esta línea de investigación resulta de particular relevancia para la industria del software, ya que facilita el desarrollo de estrategias de prueba más efectivas para la identificación de deficiencias en etapas tempranas del ciclo de vida del software. Además, se implementó el modelo de Ley de Potencia para evaluar la fiabilidad del software, demostrando su aplicabilidad en escenarios prácticos y su capacidad para proporcionar estimaciones precisas de fiabilidad en distintos entornos de desarrollo (Renu Deepti, Srivani & Sandhya, 2018). Este modelo permite a las empresas predecir el rendimiento del software antes de su lanzamiento.

Asimismo, el año 2019 marcó un hito significativo con la introducción de enfoques innovadores en el análisis de la fiabilidad del software. Un avance significativo consistió en la aplicación del algoritmo de optimización de enjambre de partículas no lineal (NPSO) a un modelo de proceso no homogéneo de Poisson (NHPP), lo cual resultó en una mejora de la precisión en la predicción de la fiabilidad del software (Rani & Mahapatra, 2019). En el ámbito industrial, la implementación de estos algoritmos posibilita la optimización de los procesos de

mantenimiento y la reducción del riesgo de fallos críticos en software de misión crítica, tales como aplicaciones financieras o sistemas de control industrial. Además, se ha empleado la utilización de algoritmos genéticos para evaluar la fiabilidad del software, lo que ha permitido resaltar el papel que desempeñan los métodos evolutivos en la identificación de patrones de error y en la optimización de estrategias de aseguramiento de calidad en proyectos de desarrollo de gran escala (Jain & Sharma, 2019).

Estos avances han permitido fortalecer la fiabilidad del software desde etapas tempranas del desarrollo, ofreciendo a la industria herramientas más sofisticadas para la detección temprana de defectos, la optimización del mantenimiento predictivo y la mejora en la estabilidad de productos digitales. En un contexto en el que la calidad del software constituye un factor determinante para la competitividad empresarial, estos hallazgos representan una contribución significativa para la mejora continua en la ingeniería de software y la reducción de costes asociados a errores en producción.

Se tiene la figura 3, donde se identifican en color amarillo los autores que se han destacado tanto en productividad como impacto; en color azul se distinguen aquellos autores que son referentes en cuanto a impacto académico medido por el número de citas recibidas en sus publicaciones; por último, en color verde se encuentra un tercer grupo que se destaca en productividad debido al número de sus publicaciones. Se encontró que los principales autores en términos de productividad e impacto fueron Malhotra R y Smidts C, mientras que Jain A y Stutzke M con Stoddard RW figuraron por impacto y Fujiwara T destacó en productividad.

Figura 3*Principales autores*

Nota. Elaboración propia a partir de Scopus y Web of Science

Los hallazgos de diversos autores han tenido un impacto significativo en la industria del software, especialmente en el desarrollo de estrategias para mejorar la fiabilidad y calidad del software en etapas tempranas. Entre los autores mencionados, Malhotra ha contribuido de manera notable al campo de la predicción de fallos en software mediante el uso de métodos estadísticos y de aprendizaje automático. En su estudio de 2012, Malhotra y Jain (2012) abordaron la mejora de la calidad del software mediante el uso de técnicas de programación genética, lo que ha permitido a las empresas de software reducir los defectos antes de la fase de implementación. Además, su desarrollo de un modelo basado en programación de expresión genética (Singh, Kaur & Malhotra, 2009) ha aportado un enfoque innovador para predecir la

calidad del software, facilitando la automatización en la evaluación del rendimiento de sistemas complejos en la industria.

Por otro lado, Smidts ha realizado contribuciones fundamentales al modelado de la fiabilidad del software en etapas tempranas del desarrollo, aspecto crucial para las empresas que buscan minimizar fallos en la producción. Su investigación en 1998 introdujo un marco metodológico para la predicción temprana de la fiabilidad (Smidts, Stutzke & Stoddard, 1998), lo que ha servido de base para las herramientas modernas de análisis de calidad en la industria. Además, su estudio de 2017 presentó un enfoque basado en métricas cuantitativas (Shi et al., 2017), lo que permitió a las empresas evaluar de manera más objetiva la fiabilidad del software y tomar decisiones basadas en datos durante el ciclo de desarrollo.

En el campo de la aplicación de aprendizaje automático en la industria del software, Jain, en colaboración con Malhotra, ha desempeñado un papel crucial en la implementación de modelos predictivos para la detección de fallos (Malhotra & Jain, 2012). Este tipo de enfoques han sido adoptados por empresas de desarrollo de software para mejorar sus estrategias de aseguramiento de calidad, reduciendo los costes asociados a la corrección de errores en etapas avanzadas del desarrollo. Asimismo, Stutzke y Stoddard, en colaboración con Smidts, aportaron uno de los primeros enfoques comprensivos para el modelado de fiabilidad del software (Smidts, Stutzke & Stoddard, 1998), lo que resultó fundamental para la implementación de estándares en pruebas de software y evaluación de calidad en la industria.

En lo que respecta a la productividad y a las contribuciones a la industria, autores como Fujii y Dohi presentaron en 2011 métodos innovadores para la evaluación de la fiabilidad en procesos de desarrollo incremental (Fujii et al., 2011). Los hallazgos de estos autores han permitido a las empresas que implementan metodologías ágiles mejorar la predicción de defectos

y optimizar los procesos de prueba continua, asegurando entregas más estables en ciclos de desarrollo iterativo.

En última instancia, los aportes de Marandi y Khan han resultado de gran relevancia para la optimización de costos en el ámbito de la industria del software. En 2015, estos investigadores examinaron el impacto de los modelos de regresión lineal en la mejora de la calidad del software y la estimación de costes (Marandi & Khan, 2015), lo que ha permitido a las empresas adoptar modelos predictivos para gestionar eficazmente sus recursos. En 2017, Marandi y Khan continuaron explorando métodos estadísticos para mejorar la calidad y reducir los costes, proporcionando herramientas útiles para que las organizaciones maximicen la eficiencia en sus procesos de desarrollo sin comprometer la calidad del software (Marandi & Khan, 2017).

En términos de principales revistas, se identificaron dos grupos distintivos, como se observa en la Figura 4. El primer grupo, en color azul, comprende revistas que se destacan por su impacto, a pesar de tener un bajo índice de productividad científica, esta calificación de impacto se recibe por obtener un gran número de citas en sus artículos publicados; entre las que se incluyen *Journal of Information Processing Systems*, *IEEE Transactions on Reliability* e *ICIC Express Letters*. En contraste, el segundo grupo de revistas científicas referentes, identificado en color verde, se caracteriza por su alta productividad científica, aunque no necesariamente por su número de citas, en este último grupo, se destacan principalmente *Communications in Computer and Information Science* y *Journal of Theoretical and Applied Information Technology*.

Figura 4*Principales revistas*

Nota. *Elaboración propia a partir de Scopus y Web of Science*

Las principales revistas científicas en términos de impacto, como el Journal of Information Processing Systems, IEEE Transactions on Reliability e ICIC Express Letters, han desempeñado un papel clave en la consolidación del conocimiento sobre la calidad del software en sus etapas iniciales, proporcionando herramientas y enfoques que han sido adoptados en la industria para mejorar la fiabilidad y el rendimiento del software.

En el Journal of Information Processing Systems, Malhotra y Jain (2012) presentaron un estudio fundamental sobre la predicción de fallos en el software mediante métodos estadísticos y aprendizaje automático. Esta investigación ha sido de vital importancia para la industria del software, ya que ha permitido a las empresas implementar modelos predictivos que reducen los

defectos antes de la implementación, optimizan los procesos de aseguramiento de calidad y disminuyen los costes asociados a correcciones tardías.

Por otra parte, IEEE Transactions on Reliability es reconocida por su impacto en la modelización de la fiabilidad del software. En este sentido, el estudio de Smidts, Stutzke y Stoddard (1998) sobre la predicción temprana de la fiabilidad ha permitido a los desarrolladores y a las empresas de software prever problemas desde las fases iniciales del desarrollo, facilitando la toma de decisiones estratégicas para la implementación de medidas correctivas oportunas (Smidts, Stutzke y Stoddard, 1998). Estos enfoques han sido ampliamente utilizados en la industria para mejorar la estabilidad de productos de software críticos, como los sistemas embebidos y las plataformas financieras.

Además, ICIC Express Letters ha contribuido significativamente mediante estudios como el de Sahu y Srivastava (2018), quienes aplicaron enfoques de soft computing para la predicción de la fiabilidad del software. Este tipo de metodologías han sido adoptadas por la industria para mejorar la calidad del software en sectores altamente regulados, como la aviación y la salud, donde la predicción precisa de fallos es fundamental para garantizar la seguridad y el cumplimiento normativo.

En lo que respecta a la productividad, se ha observado que revistas como Communications in Computer and Information Science y Journal of Theoretical and Applied Information Technology han desempeñado un papel crucial en la difusión de investigaciones con un impacto significativo en el ámbito de la industria del software. Communications in Computer and Information Science ha sido un vehículo para la promoción de estudios relacionados con la armonización de enfoques orientados a procesos y productos en la calidad del software, como lo evidencia el trabajo de García-Mireles et al. (2012). Estos hallazgos han sido implementados en

empresas para mejorar sus modelos de desarrollo de software, asegurando una integración eficiente de metodologías ágiles y de ingeniería de software tradicional (Alipour & Isazadeh, 2008).

Otro estudio publicado en esta misma revista examinó la predicción de la fiabilidad del software basada en especificaciones formales de requisitos (Alipour & Isazadeh, 2008), destacando la importancia de una documentación precisa y estructurada para la reducción de errores en proyectos a gran escala. Este enfoque ha sido adoptado por la industria en el desarrollo de software de misión crítica, como sistemas de control industrial y aplicaciones financieras.

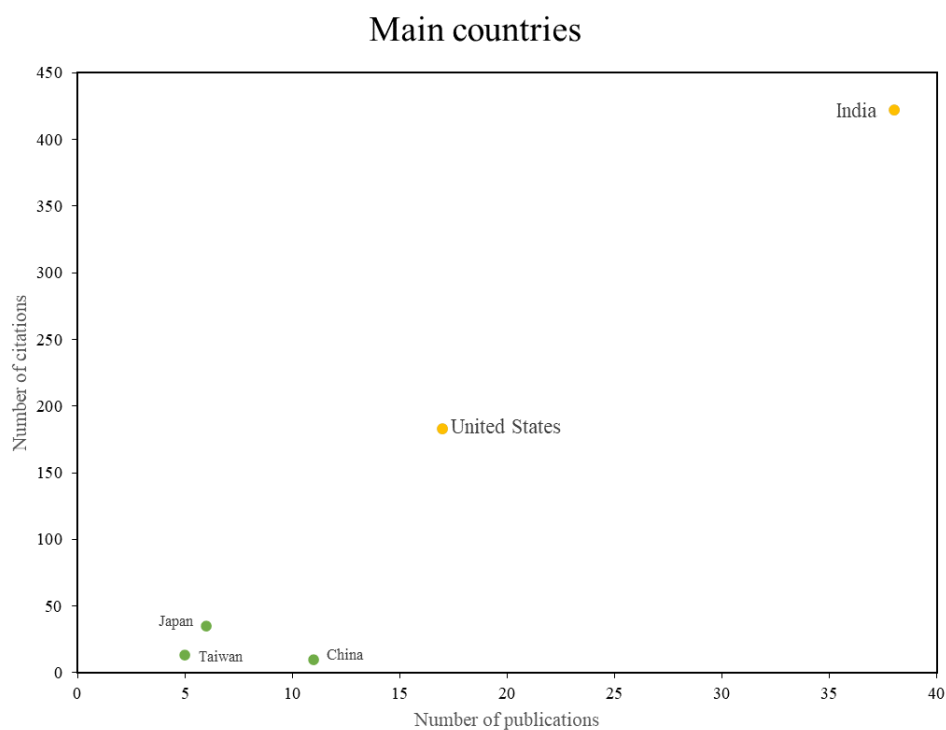
Finalmente, el *Journal of Theoretical and Applied Information Technology* ha sido fundamental en el desarrollo de modelos predictivos aplicables a diversas fases del desarrollo de software. En este sentido, Ibraigheeth y Fadzli (2018) llevaron a cabo un exhaustivo análisis de estrategias de predicción de fiabilidad, mientras que Park (2022) realizó un análisis comparativo sobre atributos de fiabilidad en modelos de software con características de distribución exponencial. Estos estudios han ejercido una influencia significativa en la industria al proporcionar fundamentos teóricos para la implementación de métricas avanzadas en la evaluación de la calidad del software, lo que ha resultado en una mejora en la toma de decisiones en proyectos de desarrollo a gran escala.

En términos de principales países, se identificaron dos grupos distintivos, como se muestra en la Figura 5, el grupo de color amarillo corresponde a los países más relevantes en cuanto a impacto y número de publicaciones, el impacto medido por la cantidad de citas que reciben los artículos publicados en cada país; mientras que el grupo de color verde es el más productivo por la cantidad de artículos que se han publicado. El primer grupo incluye países que se destacan tanto en productividad como en impacto, siendo India y Estados Unidos los más

prominentes en esta categoría, estos países no solo producen una gran cantidad de investigaciones, sino que también reciben un alto número de citas, indicando la influencia y relevancia de sus trabajos en la comunidad científica. El segundo grupo está compuesto por países que se distinguen principalmente por su alta productividad científica, aunque no necesariamente por su número de citas, en este grupo sobresalen Japón, China y Taiwán, que aportan significativamente al volumen de publicaciones sobre la calidad del software en etapas tempranas, contribuyendo de manera sustancial al avance del conocimiento en este campo, aunque con menor impacto en términos de citas.

Figura 5

Principales países



Nota. Elaboración propia a partir de Scopus y Web of Science

India ha destacado como un referente significativo en el ámbito de la industria del software, particularmente en lo que respecta a la evaluación temprana de la fiabilidad y el diseño

del software. Tripathi y Mall (2005) sentaron las bases para la mejora continua del software en sus etapas iniciales, lo que ha permitido a las empresas tecnológicas optimizar sus ciclos de desarrollo y reducir los costes asociados a errores tardíos. Asimismo, Kumar y Yadav (2013) introdujeron modelos de calidad basados en métricas de los primeros artefactos del desarrollo, facilitando la detección temprana de problemas y la adopción de medidas correctivas a tiempo. Estos enfoques han sido fundamentales en sectores como el desarrollo de aplicaciones empresariales y la industria de servicios de TI, donde India se erige como un líder mundial en externalización de software.

Por su parte, Estados Unidos ha realizado contribuciones fundamentales que han transformado la gestión y el desarrollo de software en la industria. En este sentido, Boehm (1987) propuso métodos para mejorar la productividad del software, integrando aspectos técnicos y de gestión, lo que ha sido clave en empresas de tecnología para optimizar el desarrollo ágil y reducir costos operativos. Asimismo, el estudio de Yin, Peterson y Arellano (2004) sobre la complejidad del software en la evaluación de la fiabilidad ha proporcionado herramientas esenciales para la gestión del riesgo en proyectos a gran escala. Estas investigaciones han sido adoptadas por compañías líderes en tecnología, como Microsoft, Google y Oracle, para mejorar la calidad y la confiabilidad de sus productos.

En el contexto japonés, la industria del software ha experimentado un progreso significativo mediante el desarrollo de modelos de crecimiento fiables, empleando técnicas avanzadas de aprendizaje profundo. Investigaciones como las de San et al. (2019, 2021) han mejorado la precisión de las predicciones de fallos en proyectos de software, lo que ha beneficiado a empresas de sectores como la robótica, la manufactura inteligente y los sistemas

embebidos. Estas innovaciones han facilitado la implementación de soluciones más confiables en áreas críticas, como la automoción y la electrónica de consumo.

En contraste, China ha promovido enfoques innovadores para la predicción de la calidad del software y la recopilación de requisitos de fiabilidad, como se evidencia en estudios como los de Peng et al. (2009) y Pan et al. (2018). Estas investigaciones han sido adoptadas por la industria del software para mejorar las prácticas de desarrollo y asegurar la calidad desde las primeras etapas del ciclo de vida del software. Empresas tecnológicas chinas de renombre, tales como Huawei y Alibaba, han adoptado estos enfoques para garantizar la fiabilidad en productos de alto impacto, tales como infraestructuras en la nube y plataformas de comercio electrónico.

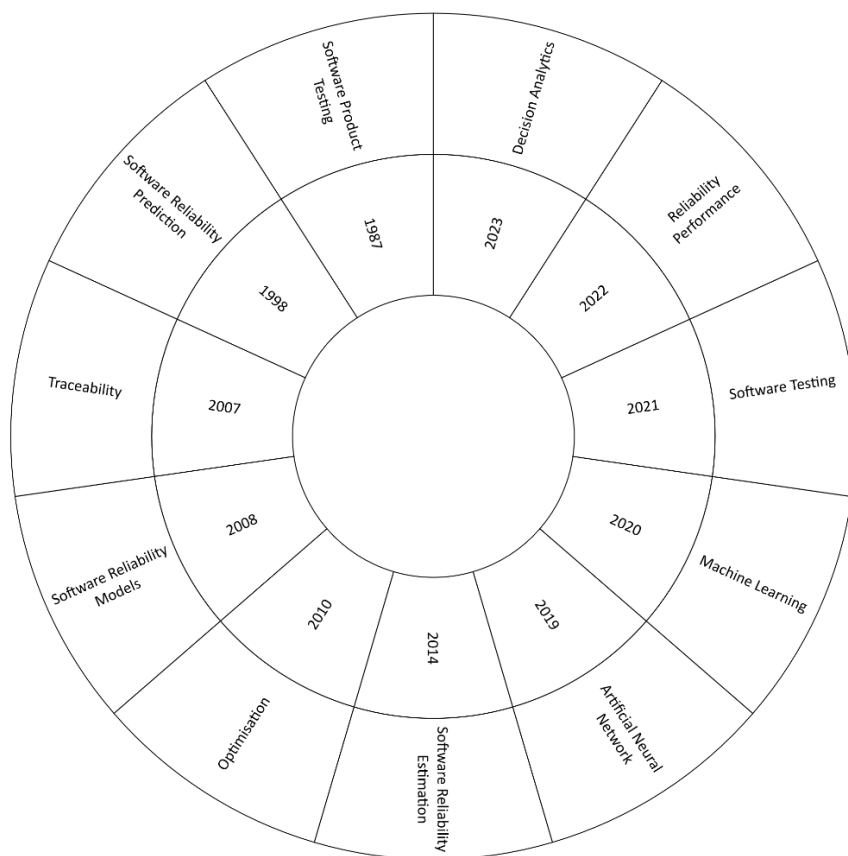
En Taiwán, se han desarrollado métodos avanzados para gestionar la calidad del software de manera eficiente. Hsu y Huang (2008) integraron pruebas de camino con estimaciones de fiabilidad, mientras que Lo (2010) aplicó algoritmos de máquina de soporte vectorial y algoritmos genéticos para la predicción de fiabilidad del software. Estas metodologías han sido adoptadas por la industria para mejorar la precisión en la detección de fallos y optimizar los procesos de aseguramiento de calidad, lo que redundará en beneficio de las empresas de semiconductores y fabricantes de software embebido en hardware, como MediaTek y ASUS.

La presente investigación, como se observa en la Figura 6, exploró la evolución temática en la literatura, analizando la palabra clave más utilizada en cada año de investigación entre 1987 y 2023, esta figura presenta los conceptos clave más relevantes y el año en que más se estudió sobre ese tema. Se observó que en 1987, como primer año, destacó la aparición de conceptos como Software Product Testing. Posteriormente, en 1998, se enfocó en Software Reliability Prediction, seguido de Traceability en 2007 y Software Reliability Estimation en 2014. Además, en los últimos años se evidencia una preponderancia de temas como Decision Analytics,

Reliability Performance, Software Testing, y Machine Learning, lo que refleja las tendencias actuales de investigación en el campo de la calidad del software en etapas tempranas.

Figura 6

Evolución temática



Nota. *Elaboración propia a partir de Scopus y Web of Science*

El concepto de prueba de software, estudiado en sus primeras etapas por Boehm (1987), desempeñó un papel crucial en el desarrollo de estándares y metodologías que aún hoy son fundamentales en la industria del software. La incorporación de las pruebas de software en las fases iniciales del ciclo de vida resultó en la reducción de defectos y el consiguiente ahorro de costes asociados a la corrección de problemas tardíos, lo cual benefició notablemente a sectores como el desarrollo de software empresarial, las aplicaciones bancarias y los sistemas críticos

para la salud y la seguridad. Este enfoque inicial sentó las bases para investigaciones posteriores sobre la predicción y estimación de la fiabilidad del software, influyendo en la evolución de metodologías de aseguramiento de calidad.

En 1998 se produjo un punto de inflexión con el enfoque en la predicción de la fiabilidad del software, lo que reflejó la necesidad de anticipar y prevenir fallos en los sistemas de software (Smidts, Stutzke y Stoddard, 1998). Este avance permitió a la industria del software desarrollar modelos predictivos utilizados en sectores como el financiero y el aeroespacial, donde la confiabilidad del software es crítica (Smidts, Stutzke y Stoddard, 1998). Las empresas de tecnología adoptaron estos enfoques para mejorar la resiliencia de sus productos, asegurando una mayor disponibilidad y menor tiempo de inactividad en sistemas como redes de telecomunicaciones y plataformas de comercio electrónico (Smidts, Stutzke, & Stoddard, 1998).

En 2007, la investigación sobre trazabilidad puso de manifiesto la relevancia del rastreo y la gestión de los artefactos de software a lo largo de todo el ciclo de vida del desarrollo (Van Den Berg, Conejero & Hernández, 2007). Este enfoque resultó fundamental para sectores regulados como la industria farmacéutica y la automotriz, donde la trazabilidad del software constituye un requisito normativo. En consecuencia, las empresas empezaron a adoptar herramientas de gestión de requisitos y trazabilidad con el fin de garantizar el cumplimiento normativo y mejorar la calidad de sus productos.

En este sentido, el concepto de estimación de fiabilidad del software adquirió una relevancia significativa en 2014, permitiendo a desarrolladores e investigadores evaluar y mejorar la confiabilidad de los sistemas de software (Garg y Lai, 2014). Esta evolución resultó fundamental para industrias que dependen de sistemas de misión crítica, tales como la aviación y la infraestructura de TI, donde las fallas pueden ocasionar pérdidas millonarias. En respuesta a

esta situación, las empresas líderes en el sector han comenzado a integrar modelos de estimación de fiabilidad en sus procesos de desarrollo, con el objetivo de ofrecer soluciones más robustas y confiables.

El año 2020 marcó un punto de inflexión en el ámbito del aseguramiento de la calidad del software, con la irrupción del Machine Learning. Esta tendencia, como se evidencia en el estudio de Goyal y Bhatia (2020), ha permitido el desarrollo de técnicas de aprendizaje automático para la predicción de la calidad del software. Esta tendencia ha sido aprovechada por la industria del software para mejorar la detección temprana de defectos mediante algoritmos avanzados, optimizando los procesos de prueba y reduciendo los tiempos de desarrollo (Goyal y Bhatia, 2020). En consonancia con lo anterior, Nakamichi et al. (2020) formularon un enfoque basado en requisitos para definir métricas de calidad en software de aprendizaje automático, lo que ha impulsado la adopción de estrategias de calidad en sectores que utilizan la IA, como la conducción autónoma y la salud digital.

En 2021, la investigación en pruebas de software se centró en la estandarización de procesos y en la automatización de pruebas (Zhao, Hu y Gong, 2021; Esnaashari y Damia, 2021). La implementación de herramientas automatizadas de generación de datos de prueba ha optimizado la calidad del software en industrias con ciclos de desarrollo acelerados, como las aplicaciones móviles y el desarrollo ágil en startups tecnológicas (Esnaashari y Damia, 2021). La implementación de la automatización ha permitido a las empresas reducir costes y mejorar la eficiencia en las pruebas de regresión y validación de software.

En el año 2022, la investigación sobre Reliability Performance puso de manifiesto la relevancia de evaluar la confiabilidad del software mediante métodos estadísticos (Hou, Huang y Fang, 2022; Yang, 2022). Estos avances han sido fundamentales para industrias que requieren

software con altos estándares de rendimiento, como la banca digital y las plataformas de nube, donde la disponibilidad y la confiabilidad del software impactan directamente en la experiencia del usuario y en la estabilidad de los servicios.

En el año 2023, la investigación en Decision Analytics ha adquirido una relevancia significativa, con enfoques innovadores como la técnica híbrida fuzzy AHP-TOPSIS aplicada a la selección de parámetros en pruebas de software (Singh y Kumar, 2023). Este tipo de analítica avanzada ha sido adoptada en la industria para optimizar la toma de decisiones en la gestión de la calidad del software. Además, Attaran y Celik (2023) han explorado los beneficios de los Digital Twins en la ingeniería de software, lo que ha abierto nuevas oportunidades para la simulación y validación de software en entornos controlados antes de su implementación en producción.

La presente bibliometría presenta la principal red de coocurrencia de palabras clave a través de un total de ocho clústeres temáticos, como se observa en la Figura 7, esta red presenta los conceptos más relevantes relacionados con el tema de investigación y cómo estos se relacionan entre sí. El clúster de color rojo, compuesto por términos como Software Metrics, Artificial Neural Network, Software Reliability Growth Model, Open Source Software y Design Metrics, es el más destacado. Le sigue el clúster de color azul, que incluye términos como Machine Learning, Software Defect Prediction, Statistical Methods, Metrics y Empirical Validation, asimismo, se identifican otros clústeres de colores azul claro, morado, verde, amarillo y naranja, que reflejan otros elementos de afinidad conceptual en el campo de la calidad del software en etapas tempranas. Por ejemplo, el clúster verde resalta el concepto de Fuzzy Logic, mientras que en el naranja el término más relevante es Reliability Prediction, además el clúster de color azul claro resalta el software Engineering en afinidad con otros términos como Software Architecture, Multi-Criteria Decision Making y Software Reliability Evaluation.

software de gran escala. Además, Lee y Baik (2011) subrayan la importancia de las mediciones del ciclo de vida temprano para la adopción de software de código abierto, mientras que Gokhale (2005) analiza expresiones de varianza en modelos de crecimiento de fiabilidad, proporcionando herramientas fundamentales para la planificación de estrategias de mantenimiento y mejora continua.

El segundo conglomerado de mayor relevancia, representado por el color azul, agrupa términos como «aprendizaje automático», «predicción de defectos de software», «métodos estadísticos», «métricas» y «validación empírica», lo que evidencia el enfoque predominante en la industria hacia la predicción de defectos de software mediante técnicas estadísticas y de aprendizaje automático. La implementación de estos métodos ha sido extensiva en el ámbito industrial, con el propósito de optimizar la detección temprana de errores y potenciar la confiabilidad de los productos. Kumar, Nayak y Behera (2022) presentan un modelo basado en el aprendizaje conjunto para la predicción de defectos a partir de características de calidad del software, lo que permite a las empresas reducir costes en pruebas y depuración.

Por otro lado, Okumoto (1985) describe un método estadístico para el control de calidad del software, influyendo en la implementación de estándares de calidad en la industria. Además, Fenton y Neil (1999) critican los modelos tradicionales de predicción de defectos, resaltando la necesidad de enfoques empíricos más sólidos que respalden las decisiones en la gestión de proyectos de software. Investigaciones como las de Voas (1998) y Olague et al. (2007) han contribuido significativamente al desarrollo de métricas más precisas para evaluar la calidad en ciclos iterativos y metodologías ágiles, fortaleciendo la fiabilidad de los sistemas en entornos empresariales (Olague et al., 2007).

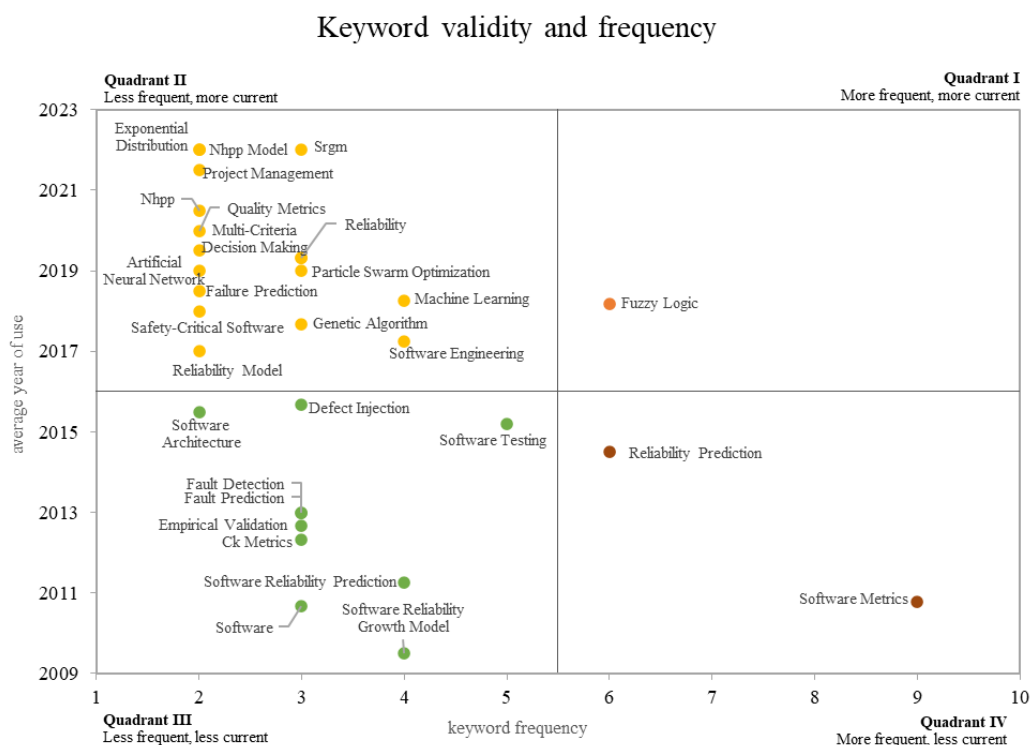
En lo que respecta a otros clústeres de relevancia, el clúster de color verde, centrado en Fuzzy Logic, ha propulsado el desarrollo de modelos híbridos neuro-difusos aplicados en la evaluación de la fiabilidad del software (Gandhi et al., 2022). La implementación de estos modelos en sectores como la banca y la automatización industrial ha contribuido a la optimización de la toma de decisiones en sistemas críticos. Por su parte, el clúster naranja, centrado en Reliability Prediction, destaca la relevancia de la predicción temprana de la fiabilidad del software, con estudios como el de Abu Talib (2016), que proveen marcos metodológicos aplicables a la ingeniería de software en sectores como la aeronáutica y la salud.

Por último, el clúster azul claro, que engloba términos como Ingeniería del Software, Arquitectura del Software, Decisión Multicriterio y Evaluación de la Fiabilidad del Software, resalta la evaluación de la calidad en revisiones sistemáticas y enfoques avanzados de estimación de la fiabilidad del software. Yang et al. (2021) y Bei-Yang (2013) presentan métodos innovadores para evaluar la calidad del software en revisiones sistemáticas, lo que ha favorecido el diseño de arquitecturas más robustas en la industria. Zhao et al. (2005) y Neha et al. (2021) abordan la estimación y asignación de fiabilidad en las primeras etapas de desarrollo utilizando redes de recompensa estocásticas y teoría difusa de Pitágoras, metodologías que han sido adoptadas por empresas tecnológicas para mitigar riesgos en proyectos de software de alta complejidad (Yang et al., 2021).

Se propone un plano cartesiano que mide la frecuencia de uso de palabras clave en el eje X y la vigencia de uso en el eje Y, permitiendo observar cuatro cuadrantes diferentes, como se muestra en la Figura 8. Para ello se dividieron los resultados en cuatro cuadrantes, siendo el I donde se localizan las palabras con mayor novedad y periodicidad; el II por su parte, alberga aquellos conceptos que de igual manera son novedosos, manteniendo su vigencia pero con una

frecuencia más baja, lo que los posiciona como conceptos emergentes en el campo; el III se tiene conceptos de poca frecuencia y poca vigencia, por último, los pertenecientes al cuadrante IV se caracterizan por tener poca vigencia pero son muy frecuentes, por ello se consideran decrecientes. En el cuarto cuadrante, se encuentran los conceptos decrecientes, tales como Reliability Prediction y Software Metrics, estos términos, aunque fueron prominentes en el pasado, han experimentado una disminución en su frecuencia de uso, lo que indica una posible saturación o una evolución hacia términos más específicos o avanzados. El tercer cuadrante alberga conceptos que no han logrado consolidarse completamente en la literatura, como Defect Injection, Software Testing y Software, estos términos, aunque relevantes, no han alcanzado la prevalencia necesaria para establecerse firmemente en la investigación actual.

En contraste, el segundo cuadrante muestra palabras clave que, aunque son poco frecuentes, tienen una alta vigencia y se consideran emergentes, ejemplos de estos términos incluyen Srgm, Project Management, Genetic Algorithm, Failure Prediction y Quality Metrics, estas palabras clave reflejan áreas de investigación novedosas y prometedoras que están ganando relevancia en la comunidad científica. Por último, el primer cuadrante está reservado para conceptos consolidados y crecientes, como Fuzzy Logic, este término no solo es frecuentemente utilizados, sino que también mantienen una alta vigencia, lo que indica su importancia y relevancia continua en el campo de la calidad del software en etapas tempranas.

Figura 8*Vigencia y frecuencia de las palabras clave*

Nota. *Elaboración propia a partir de Scopus y Web of Science*

En el plano cartesiano, el cuarto cuadrante agrupa los conceptos que han experimentado una disminución en su frecuencia de uso en la literatura reciente, reflejando su menor presencia en la actualidad en comparación con periodos anteriores. Un ejemplo notable es Reliability Prediction, que en su momento de mayor relevancia se erigió como una herramienta indispensable en el ámbito de la industria del software, permitiendo anticipar y optimizar la confiabilidad de los sistemas. Este enfoque predictivo permitía estimar la probabilidad de fallos antes del despliegue del producto, lo que ayudaba a las empresas a reducir los riesgos y los costes asociados con los fallos post-producción. Sahu et al. (2021) pusieron de manifiesto la relevancia de estas técnicas en la evaluación de la fiabilidad del software, subrayando su impacto

en la optimización de los procesos de desarrollo en sectores como la banca, la salud y la automatización industrial. No obstante, en la actualidad, el avance de técnicas más sofisticadas basadas en Machine Learning y Artificial Intelligence ha desplazado en gran medida este enfoque tradicional, dando paso a metodologías de predicción más avanzadas.

Otro concepto que ha experimentado un declive es el de Software Metrics, que en el pasado fue fundamental para la evaluación cuantitativa de diversos aspectos del desarrollo y rendimiento del software. En el ámbito industrial, dichas métricas ofrecían un marco objetivo para la toma de decisiones, posibilitando la evaluación de aspectos tales como la complejidad del código, la mantenibilidad y la fiabilidad del software. En este sentido, investigaciones previas, como las llevadas a cabo por Wohlin, Wesslén y Runeson en 1996, exploraron la aplicabilidad de las métricas para la predicción de la confiabilidad del software mediante el análisis del uso de especificaciones y diseños, facilitando la detección temprana de errores. No obstante, la implementación de enfoques ágiles y el desarrollo iterativo han ocasionado una disminución en la dependencia hacia métricas tradicionales, favoreciendo el uso de indicadores dinámicos que se ajustan a metodologías de entrega continua y DevOps.

En el tercer cuadrante, los conceptos, si bien no se han consolidado completamente, siguen siendo relevantes en el ámbito de la industria del software. El término Defect Injection hace referencia a la práctica de introducir defectos de forma deliberada en el software con el propósito de evaluar la eficacia de los procesos destinados a la eliminación de errores. Esta técnica resulta fundamental para mejorar la resiliencia de los sistemas en entornos donde la confiabilidad es crítica, como en el software de control de infraestructuras o sistemas financieros. Marandi y Khan (2015) resaltan la importancia de esta metodología para fortalecer los

mecanismos de detección y corrección de errores, lo que ha llevado a su adopción en pruebas automatizadas avanzadas.

En el ámbito del testing de software, Villegas et al. (2011) enfatizan la evaluación rigurosa de sistemas de software adaptativos y autogestionados. En el sector industrial, esta relevancia se ha acentuado con el auge de sistemas basados en Internet of Things (IoT) y Edge Computing, donde el software debe responder de manera dinámica a cambios en el entorno. La necesidad de frameworks de evaluación continua ha impulsado la automatización de pruebas y la integración de herramientas de validación en entornos DevSecOps.

En última instancia, el concepto general de software abarca múltiples aspectos del desarrollo y gestión de sistemas, y su percepción de calidad varía en función de la experiencia del usuario. Cordeiro y Freitas (2012) exploraron la importancia de alinear las expectativas de los usuarios con las prácticas de desarrollo, destacando cómo la priorización de requisitos impacta en la satisfacción del cliente. En el ámbito industrial, esta tendencia ha propiciado un enfoque creciente en metodologías centradas en la experiencia del usuario (User Experience o UX), donde la integración de la retroalimentación en todas las etapas del proceso de desarrollo resulta imperativa para asegurar la calidad del producto final. Estos hallazgos subrayan la evolución de la industria del software hacia enfoques más dinámicos, donde la adaptabilidad y la automatización desempeñan un papel central en la mejora continua de los sistemas.

En el análisis del cuadrante 2 del plano cartesiano, se identificaron conceptos emergentes en el ámbito de la calidad del software, caracterizados por su escasa presencia en la literatura científica y su notable relevancia en el contexto de la investigación y la industria. Estas tendencias reflejan avances significativos en el ámbito de la ingeniería de software y tienen un

impacto directo en la mejora de la confiabilidad y eficiencia de los sistemas empleados en sectores críticos, tales como la banca, la salud y la industria manufacturera.

Uno de los conceptos más notables es el denominado Software Reliability Growth Models (SRGM), modelos concebidos para predecir y potenciar la confiabilidad del software a lo largo de su ciclo de vida. Estos modelos permiten a los ingenieros de software identificar y corregir fallos de manera proactiva, lo que es crucial en sistemas donde la fiabilidad es un factor determinante, como en software de control de infraestructuras o aplicaciones médicas. En este sentido, Oveisi et al. (2023) hacen hincapié en la importancia de implementar estos modelos en la gestión del desarrollo de software, lo que permite a las empresas optimizar sus estrategias de mantenimiento y minimizar los costes asociados a fallos inesperados.

En el ámbito del software, la implementación de metodologías de gestión de proyectos se ha convertido en un aspecto de suma relevancia, particularmente en el contexto de la complejidad creciente que caracterizan estos desarrollos. Haque y Ahmad (2022) ponen de manifiesto la relevancia de modelos de confiabilidad fundamentados en la eficiencia de la eliminación de fallos, que posibilitan una gestión más efectiva de los proyectos de software. En el ámbito industrial, esta tendencia se manifiesta en la adopción de metodologías ágiles y marcos de trabajo como Scrum y DevOps, que persiguen la optimización de la entrega continua de productos con altos estándares de calidad.

El uso de algoritmos genéticos en la optimización de la calidad del software representa otra tendencia emergente con aplicaciones directas en la industria. Estos algoritmos, inspirados en la selección natural, han sido empleados para resolver problemas de optimización en el desarrollo de software, mejorando la asignación de recursos, la identificación de fallos y la optimización del rendimiento. Suresh (2015) evidenció la eficacia de estos algoritmos en la

optimización del software orientado a objetos, lo que resultó en su adopción en aplicaciones de inteligencia artificial, análisis de big data y optimización de redes de comunicación.

En el ámbito de la industria del software, la Failure Prediction emerge como un concepto en auge. La capacidad de anticipar fallos antes de su ocurrencia permite a las empresas minimizar el tiempo de inactividad y reducir los costes de mantenimiento. En este sentido, Nagappan, Murphy y Basili (2008) llevaron a cabo un análisis que exploró cómo la estructura organizacional influye en la calidad del software y en la capacidad de anticipar problemas. En este sentido, en sectores como el financiero y el de telecomunicaciones, la predicción de fallos mediante el uso de modelos avanzados de análisis de datos y aprendizaje automático se erige como un elemento fundamental para garantizar la estabilidad de los sistemas.

Las Quality Metrics se erigen como un componente esencial para la evaluación de la calidad del software desde una perspectiva cuantitativa. En este sentido, Khalid, Rasheed y Abbas (2021) propusieron un marco impulsado por modelos para estandarizar la elicitación de requisitos, asegurando la calidad desde las primeras fases del desarrollo. En el ámbito industrial, la implementación de métricas de calidad facilita la toma de decisiones basadas en datos para la mejora de productos, garantizando la conformidad con estándares internacionales como ISO/IEC 25010.

En el marco del estudio, se identificó que la metodología Fuzzy Logic (Lógica Fuzzy) es un componente esencial en la evaluación y optimización de la calidad del software. La lógica difusa se utiliza en la industria para gestionar la incertidumbre en los procesos de desarrollo y evaluación de software. Masood y Khan (2018) demostraron que su aplicación en sistemas de inferencia mejora significativamente la predicción temprana de la calidad del software, facilitando la toma de decisiones en entornos con información incompleta. El empleo de la lógica

difusa se ha expandido progresivamente en campos como la inteligencia artificial, el procesamiento de datos y los sistemas expertos, permitiendo la evaluación de la confiabilidad del software en sectores como la robótica y la automatización industrial.

Yaremchuk y Kharchenko (2022) examinaron la implementación de modelos de lógica difusa fundamentados en la similitud para la evaluación de la confiabilidad en software crítico, enfatizando su relevancia en contextos donde los datos pueden presentar ambigüedad. A medida que la demanda de software de alta calidad y confiabilidad continúa en aumento, se anticipa que la lógica difusa mantendrá su rol fundamental en la industria, particularmente cuando se fusiona con técnicas avanzadas como Machine Learning y Deep Learning. La implementación de la lógica difusa permitirá a las empresas desarrollar software más robusto, adaptable y alineado con los desafíos actuales del mercado.

Discusión

Análisis de Resultados

El análisis bibliométrico reveló que algunos autores destacan tanto en productividad como en impacto, mientras que otros se especializan en una de estas dimensiones. Esta distribución puede estar influenciada por diversos factores, como la calidad metodológica de los estudios, la aplicabilidad de sus hallazgos en entornos industriales y la colaboración con redes de investigación consolidadas. En muchos casos, la citación de un artículo no solo responde a su rigor académico, sino también a su utilidad práctica en la mejora de procesos de calidad del software. Autores con alta productividad pueden enfocarse en la exploración de múltiples enfoques y metodologías, mientras que aquellos con mayor impacto suelen producir estudios de referencia que establecen marcos conceptuales o modelos aplicables a largo plazo.

Además, el impacto de una publicación puede estar condicionado por la accesibilidad del estudio, su visibilidad en conferencias influyentes y la adopción de sus conceptos en normativas de calidad del software. En este sentido, futuras investigaciones podrían analizar el grado de aplicación de estos estudios en estándares de calidad, así como su presencia en libros de texto y documentos técnicos del sector.

El análisis de la producción científica no solo permite identificar a los autores más influyentes, sino también evaluar el crecimiento de la investigación en esta área. Los resultados revelan un crecimiento polinomial cuadrático del 99,12 % en la publicación de artículos científicos relacionados con la calidad del software en etapas tempranas, destacando los años 2018 y 2022 como periodos particularmente productivos. Este incremento refleja un interés creciente en esta temática, lo que puede atribuirse a la demanda de sistemas más confiables y eficientes en industrias tecnológicamente avanzadas. Si bien estudios previos, como el de

Smidts, Stutzke y Stoddard (1998), han resaltado la importancia de evaluar la fiabilidad en etapas tempranas, es crucial analizar si el enfoque actual de la investigación mantiene un equilibrio entre la teoría y la aplicación práctica en entornos industriales, considerando que gran parte de los modelos desarrollados carecen de validación en contextos complejos.

A pesar del liderazgo de India y Estados Unidos en la producción científica sobre métricas de calidad y fiabilidad, se observa una brecha significativa en la contribución de países emergentes y regiones en desarrollo. Esto podría explicarse por factores como la inversión en investigación y desarrollo tecnológico (Malhotra & Jain, 2012), así como por barreras económicas y de infraestructura que limitan la adopción de enfoques innovadores como los modelos híbridos neuro-difusos (Gandhi et al., 2022). En este sentido, resulta esencial fomentar colaboraciones internacionales que faciliten la transferencia de conocimientos y tecnologías hacia estos contextos menos representados.

La evolución temática observada en la investigación sobre calidad del software evidencia una transición desde enfoques tradicionales, como Software Product Testing, hacia metodologías más avanzadas, como Decision Analytics y Reliability Performance. Este cambio refleja una mayor sofisticación en las herramientas utilizadas, impulsada por la integración de técnicas como lógica difusa y aprendizaje automático para optimizar la evaluación de calidad (Gupta & Gupta, 2024).

Uno de los principales factores que ha motivado esta transformación es el aumento en la escala y heterogeneidad de los sistemas de software, lo que ha hecho que las metodologías tradicionales de prueba resulten insuficientes para garantizar confiabilidad en todas las fases del ciclo de vida del software. La incorporación de técnicas como Machine Learning y Fuzzy Logic

permite manejar mejor la incertidumbre en la evaluación de calidad, optimizando la toma de decisiones en etapas tempranas del desarrollo.

El análisis bibliométrico revela un creciente interés en técnicas como Fuzzy Logic y Machine Learning en las etapas iniciales del desarrollo de software. Sin embargo, es imperativo examinar estudios que documenten su aplicabilidad práctica. En el ámbito de la lógica difusa, Gandhi et al. (2022) implementaron un modelo neuro-difuso para evaluar la confiabilidad del software en sistemas industriales de control automático, logrando una reducción significativa de defectos antes de la fase de pruebas. Este caso ilustra la factibilidad de integrar lógica difusa en entornos productivos que demandan alta disponibilidad y limitada tolerancia a fallos.

En el ámbito del aprendizaje automático, Goyal y Bhatia (2020) implementaron algoritmos de clasificación para anticipar deficiencias en aplicaciones móviles durante la fase de diseño. El enfoque adoptado por la empresa de desarrollo ágil supuso una optimización en la asignación de pruebas y una reducción de los tiempos de retrabajo del 25 %. Los hallazgos corroboran la eficacia de los modelos predictivos como instrumentos pragmáticos para la adopción de decisiones anticipadas en ciclos iterativos de desarrollo.

Además, la creciente presión del mercado por reducir tiempos de entrega sin comprometer la calidad ha llevado a la adopción de enfoques basados en Decision Analytics, donde modelos predictivos y análisis de datos permiten estimar la confiabilidad y el rendimiento del software antes de su implementación. Este enfoque no solo minimiza los costos asociados a fallos en producción, sino que también mejora la capacidad de adaptación a nuevos estándares y regulaciones en la industria del software.

Sin embargo, aunque esta evolución representa un avance significativo, los resultados sugieren que términos clave como Reliability Prediction han disminuido en vigencia, lo que

podría indicar un desplazamiento hacia áreas más especializadas o una posible saturación en este campo. Esto plantea interrogantes sobre la sostenibilidad de estas nuevas líneas de investigación y su impacto a largo plazo en la práctica industrial, lo que resalta la importancia de seguir explorando la aplicabilidad real de estos enfoques en entornos productivos y su integración con marcos normativos existentes.

El análisis de redes de coocurrencia identificó ocho clústeres temáticos en la investigación sobre calidad del software en etapas tempranas. Sin embargo, estos clústeres no son estáticos, sino que han evolucionado con el tiempo, reflejando la integración de nuevos conceptos y enfoques interdisciplinarios.

Uno de los hallazgos clave es la interrelación entre Software Metrics y Fuzzy Logic, lo que destaca la importancia de las métricas tradicionales como base para desarrollar enfoques más avanzados. El clúster de Software Metrics, tradicionalmente asociado a mediciones estructurales del código, ha convergido con estudios sobre Machine Learning y Software Defect Prediction, lo que sugiere un interés creciente en la automatización del aseguramiento de calidad. De manera similar, la relación entre Fuzzy Logic y Reliability Prediction indica una tendencia a abordar la evaluación de calidad con modelos capaces de manejar imprecisión e incertidumbre en los datos.

Otra intersección relevante se observa entre los clústeres de Project Management y Software Engineering, reflejando la creciente importancia de enfoques integrales que consideran tanto la calidad técnica del software como la gestión eficiente de recursos y tiempos. La combinación de estos enfoques sugiere que la investigación en calidad del software está evolucionando hacia un paradigma donde los aspectos técnicos y organizacionales deben analizarse de manera conjunta para optimizar el desarrollo de software.

Sin embargo, a pesar de la evolución en estas líneas de investigación, surge la preocupación de que enfoques como Fuzzy Logic aún son limitados en su aplicabilidad práctica. Aunque herramientas como VOSviewer permiten visualizar estas relaciones conceptuales, la falta de análisis empírico en escenarios reales podría restringir la adopción de estos enfoques en proyectos de desarrollo de software. Por ello, es fundamental que futuras investigaciones combinen métodos cuantitativos y cualitativos para validar su utilidad y facilitar su implementación en entornos industriales.

Por último, la identificación de palabras clave emergentes como Project Management y Failure Prediction sugiere áreas prometedoras para el avance del campo. Estas tendencias subrayan la necesidad de integrar enfoques holísticos que no solo consideren atributos técnicos, sino también aspectos organizacionales y de gestión en el desarrollo de software (Reddy & Prasanth, 2022). No obstante, el desafío radica en garantizar que estos avances sean accesibles y aplicables en diferentes contextos industriales, especialmente aquellos con recursos limitados, brechas y maximizar el impacto de las innovaciones en calidad del software.

Comparación Con Otros Estudios

Los resultados de esta investigación muestran coherencia con estudios previos en cuanto a la relevancia de los modelos predictivos, las métricas de calidad y los enfoques sistemáticos para evaluar y mejorar la calidad del software desde etapas tempranas. Comparando estos hallazgos con estudios anteriores, se observa una convergencia en la necesidad de herramientas automatizadas, modelos predictivos precisos y consideraciones contextuales para asegurar la calidad del software, lo que demuestra la evolución continua del campo hacia enfoques más integrados y avanzados.

Investigaciones recientes han destacado la incorporación de modelos de aprendizaje automático, como Machine Learning y Deep Learning, para mejorar la predicción de defectos en software y optimizar la evaluación de calidad en etapas tempranas (Malhotra, 2015; Nam & Kim, 2015). Estos enfoques han permitido mejorar la precisión en la identificación de fallos, reduciendo costos en pruebas y mantenimiento. Sin embargo, la literatura sugiere que la falta de interpretabilidad de estos modelos puede ser un desafío para su implementación en entornos industriales, ya que los desarrolladores y tomadores de decisiones necesitan explicaciones más claras sobre los factores que influyen en las predicciones (Radjenović et al., 2013).

El trabajo de Wei et al. (2012) propone un modelo de razonamiento automatizado basado en árboles de objetivos para los requisitos de calidad del software, enfatizando la necesidad de automatizar y sistematizar la evaluación de calidad desde las etapas iniciales. Esto coincide con los hallazgos del presente estudio, que resaltan la importancia de las métricas y los modelos de crecimiento de la confiabilidad del software. Ambos estudios subrayan la necesidad de abordar la calidad del software desde el inicio del desarrollo mediante modelos predictivos y herramientas de evaluación automatizada.

El estudio de Boukhelif, Hanine & Kharmoum (2023) sobre pruebas de software inteligentes refuerza esta tendencia, destacando el crecimiento en el uso de técnicas avanzadas y automatizadas en la evaluación de la calidad del software. La inteligencia artificial y el aprendizaje automático han sido identificados como factores clave en esta transformación, complementando los hallazgos de esta bibliometría sobre la evolución hacia enfoques más sofisticados en la evaluación de la calidad del software.

Por otro lado, investigaciones anteriores han utilizado métodos estadísticos tradicionales, como la regresión logística y el análisis de varianza (ANOVA), para evaluar métricas de calidad

del software. Estos enfoques han sido ampliamente empleados debido a su solidez teórica y facilidad de aplicación, aunque presentan limitaciones en términos de adaptabilidad a datos no estructurados y sistemas dinámicos (Briand, Basili & Thomas, 2001). Además, estos métodos suelen basarse en conjuntos de datos históricos y requieren una selección manual de variables clave, lo que puede generar sesgos en la evaluación de la calidad (Fenton & Pfleeger, 1997).

Los trabajos sobre modelos de Poisson compuestos y modelos bayesianos empíricos paramétricos para predecir el crecimiento de la confiabilidad del software también encuentran eco en los resultados actuales. Aunque el concepto de Reliability Prediction ha sido identificado como decreciente en este análisis, sigue siendo un tema central en estos estudios. Barraza (2013, 2015) destaca la importancia de la precisión en la estimación de la confiabilidad a lo largo del ciclo de vida del software, lo cual es crucial para mejorar la calidad final del producto.

Asimismo, Hu, Xie & Ng (2006) y Besseris (2010) exploraron métodos para mejorar la predicción de la confiabilidad del software incorporando información previa y utilizando métodos Taguchi, respectivamente. Estos enfoques reflejan la necesidad de integrar diversas técnicas para lograr una evaluación más precisa y robusta, lo que subraya la complejidad del desafío de asegurar la calidad del software en etapas tempranas.

Además de los modelos predictivos y los métodos estadísticos, otras investigaciones han explorado enfoques complementarios que coinciden con los hallazgos del presente estudio. Karkalas y Gutierrez-Santos (2014) investigaron el uso de herramientas de calidad de código y sistemas basados en reglas para mejorar el aprendizaje de JavaScript, destacando la importancia de la calidad del código desde una perspectiva educativa. Esto complementa el análisis actual, donde se encontraron conceptos como Software Testing y Defect Injection, resaltando la necesidad de integrar prácticas de calidad desde la formación de los desarrolladores.

Por otro lado, Bharathi y Selvarani (2015) presentan un marco para la estimación de la confiabilidad del software orientado a objetos utilizando métricas de complejidad de diseño. Su estudio se alinea con las observaciones de esta investigación sobre la importancia de métricas y modelos predictivos, destacando que la utilización de métricas de diseño permite prever la confiabilidad del software e identificar problemas potenciales en etapas tempranas.

El estudio de Fischer y Senft (2016) sobre ingeniería de software centrada en el ser humano resalta la importancia de considerar factores contextuales en la evaluación de calidad, lo que complementa el análisis del cuadrante 1 del plano cartesiano propuesto en esta bibliometría. En este contexto, el papel de la lógica difusa y su capacidad de manejar la incertidumbre en los procesos de desarrollo cobra especial relevancia.

Finalmente, el trabajo de Pargaonkar (2023) sobre la intersección de la calidad del código y el análisis dinámico en el desarrollo de software refuerza la importancia de mantener altos estándares de calidad en cada fase del ciclo de vida del software. Este enfoque se refleja especialmente en el cuadrante 2 del plano cartesiano propuesto, donde la gestión de proyectos y las métricas de calidad emergen como áreas clave.

Los resultados de esta investigación coinciden con estudios previos en la necesidad de evaluar la calidad del software desde etapas tempranas mediante enfoques sistemáticos. Sin embargo, mientras que estudios como el de Al Obisat et al. (2018) abordan la calidad del software desde una perspectiva general, este análisis bibliométrico profundiza en la evolución y coocurrencia de términos y técnicas a lo largo del tiempo, proporcionando una visión más detallada de las tendencias actuales y emergentes en el campo.

Una de las divergencias más notables es la disminución en la vigencia del concepto Reliability Prediction, lo que sugiere un cambio en las prioridades de investigación hacia

enfoques más automatizados y flexibles. No obstante, algunos estudios continúan destacando la relevancia de la predicción de confiabilidad, lo que indica que sigue siendo un área importante, aunque en evolución.

El presente estudio también aporta un análisis detallado sobre la intersección entre Software Metrics y Fuzzy Logic, un aspecto poco explorado en la literatura previa (Briand, Basili & Thomas, 2001). Además, destaca la necesidad de integrar metodologías cuantitativas y cualitativas para validar la utilidad de estos enfoques en entornos industriales, una brecha que aún persiste en la investigación sobre calidad del software.

Con el propósito de complementar el análisis bibliométrico cuantitativo, se llevó a cabo una revisión cualitativa de estudios representativos que han ejercido un impacto significativo en el ámbito de la calidad del software en etapas iniciales. Los artículos en cuestión se distinguen por su elevada cita, su notable relevancia metodológica y sus innovadoras propuestas en lo que respecta a técnicas como el aprendizaje automático, la lógica difusa y los modelos de fiabilidad.

Uno de los estudios clásicos en esta línea es el de Smidts, Stutzke y Stoddard (1998), quienes desarrollaron un marco para la predicción de la confiabilidad en etapas tempranas del desarrollo de software. La metodología propuesta por el autor de este estudio combina métricas tradicionales con escenarios de prueba, lo que permite anticipar defectos antes de la fase de pruebas funcionales.

En el presente estudio se aborda la problemática concerniente a la necesidad de establecer un marco normativo que regule las actividades de las empresas de servicios temporales. Este enfoque ha sido la base de múltiples investigaciones que buscan integrar la predicción temprana en metodologías ágiles y ciclos iterativos.

Por su parte, Wei et al. (2012) propusieron un modelo de razonamiento automatizado basado en árboles de objetivos, aplicable a los requisitos de calidad del software. Este modelo ha sido ampliamente reconocido como un referente sobresaliente en el ámbito de la automatización de la toma de decisiones desde las primeras fases del desarrollo, especialmente en entornos industriales que requieren un alto grado de trazabilidad.

En el presente estudio se aborda la problemática concerniente a la metodología empleada en la investigación cualitativa. Para ello, se ha realizado una exhaustiva revisión bibliográfica que ha permitido constatar la existencia de diversas posturas al respecto. En este sentido, se ha observado que la metodología empleada en la investigación cualitativa se caracteriza por una serie de aspectos que, en conjunto, conforman un paradigma. En la literatura científica, este paradigma se ha denominado «metodología cualitativa».

En el ámbito de las técnicas emergentes, Gandhi et al. (2022) implementaron modelos híbridos neuro-difusos para predecir la confiabilidad en sistemas de control industrial. El estudio realizado evidenció una mejora en la precisión de las predicciones y una reducción de los errores no detectados. Estos hallazgos demuestran la aplicabilidad de la lógica difusa en contextos con alta incertidumbre y datos incompletos.

En el ámbito de la investigación en inteligencia artificial, Nam y Kim (2015) se distinguieron por su trabajo en la integración de algoritmos de aprendizaje automático para la predicción de defectos en el código fuente. Este estudio demostró que estos modelos superan a los enfoques estadísticos tradicionales en términos de precisión, lo que representa un avance significativo en el campo. Como se ha puesto de manifiesto en investigaciones previas, esta línea de investigación ha experimentado una notable expansión en lo que respecta a su desarrollo posterior, especialmente en lo que atañe a aplicaciones móviles y entornos ágiles.

En el presente estudio se aborda la problemática concerniente a la necesidad de establecer un marco normativo que regule las actividades de las empresas de servicios temporales.

En última instancia, Boukhelif, Hanine y Kharmoum (2023) proporcionaron una perspectiva contemporánea al explorar el empleo de pruebas de software inteligentes, integrando la inteligencia artificial en la evaluación automatizada de la calidad. El presente estudio aborda la convergencia entre la inteligencia artificial y el aseguramiento de la calidad, identificando esta última como una de las tendencias emergentes más relevantes de los últimos años. En el presente estudio se aborda la problemática concerniente a la necesidad de establecer un marco normativo que regule las actividades de las empresas de servicios temporales.

Implicaciones Teóricas

El análisis bibliométrico demuestra que la calidad del software en etapas tempranas está cobrando mayor relevancia dentro de la comunidad científica, reflejando una evolución temática que pasa de enfoques tradicionales como Reliability Prediction hacia metodologías más integrales, como Decision Analytics y Reliability Performance. Esto evidencia una transición hacia el uso de técnicas avanzadas, como la lógica difusa y el aprendizaje automático, que permiten manejar la complejidad inherente a las primeras fases del ciclo de vida del software (Gupta & Gupta, 2024). Este cambio teórico ofrece una base para investigaciones futuras que profundicen en el desarrollo de marcos híbridos que combinen estas herramientas con enfoques tradicionales para maximizar su impacto práctico.

Además, la identificación de clústeres temáticos y palabras clave emergentes proporciona una estructura conceptual que orienta a los investigadores hacia temas prioritarios y enfoques prometedores. Conceptos como Project Management y Failure Prediction destacan como áreas críticas para mejorar no solo la fiabilidad técnica, sino también la gestión de recursos y la toma

de decisiones en proyectos de desarrollo de software. Estos hallazgos subrayan la necesidad de explorar cómo integrar estos elementos en modelos predictivos más sofisticados, generando oportunidades para investigaciones interdisciplinarias que involucren ciencias de datos, gestión de proyectos e ingeniería de software.

No obstante, persiste el desafío de validar los modelos y herramientas emergentes en contextos reales, especialmente en industrias con restricciones económicas o técnicas. Este vacío teórico destaca la necesidad de estudios empíricos que evalúen la eficacia de estas metodologías en entornos diversos, promoviendo una mayor generalización y aplicabilidad de los resultados.

Implicaciones Prácticas

Los resultados de esta investigación resaltan la transición hacia herramientas avanzadas, como Fuzzy Logic y modelos híbridos de predicción, lo que representa una oportunidad para que las empresas de software optimicen sus procesos desde las primeras etapas del desarrollo. Estas técnicas permiten identificar defectos y predecir la fiabilidad del software con mayor precisión, reduciendo costos y aumentando la confiabilidad del producto final (Ángeles Medina, 2022). Sin embargo, su adopción plantea desafíos como la capacitación técnica y la adquisición de recursos adecuados, especialmente en contextos donde la implementación de tecnologías avanzadas puede ser costosa o compleja.

La predominancia de países como India y Estados Unidos en la producción científica evidencia una concentración de recursos e investigaciones en regiones con ecosistemas tecnológicos desarrollados. Este hallazgo resalta la necesidad de iniciativas globales que fomenten la transferencia de conocimiento hacia países emergentes. La colaboración internacional y los programas de financiamiento dirigidos podrían cerrar las brechas en el acceso

a tecnologías emergentes, permitiendo una aplicación más equitativa de las mejores prácticas en calidad del software.

El surgimiento de conceptos como Quality Metrics y Software Reliability Growth Models (SRGM) indica un enfoque creciente hacia la cuantificación precisa de atributos clave del software, lo que tiene implicaciones directas en la toma de decisiones organizacionales. La industria puede beneficiarse de estas tendencias integrando estas métricas en sus procesos de desarrollo y aseguramiento de calidad, optimizando la asignación de recursos y mejorando la experiencia del usuario final.

Una de las aplicaciones más relevantes es la incorporación de modelos de predicción de defectos basados en Machine Learning en los ciclos de desarrollo de software. Empresas como Microsoft y Google ya utilizan técnicas de aprendizaje automático para detectar anomalías en el código antes de su implementación, reduciendo costos de mantenimiento y mejorando la estabilidad de los sistemas. Los hallazgos de este estudio pueden servir como referencia para que pequeñas y medianas empresas adopten herramientas similares, integrando modelos predictivos en sus entornos de desarrollo para anticipar problemas de calidad.

En cuanto a la implementación de Fuzzy Logic, su creciente adopción en la literatura sugiere que podría mejorar la evaluación de métricas de calidad en entornos dinámicos e inciertos. Una posible aplicación en la industria es su integración en sistemas de evaluación de calidad automatizados, donde se requiera una clasificación flexible de defectos sin depender de umbrales rígidos. Sectores como la aviación, la salud y la ciberseguridad podrían beneficiarse particularmente de este enfoque, ya que permite analizar incertidumbres en los datos sin comprometer la precisión de la evaluación.

Para facilitar la adopción de estas metodologías, se recomienda que las empresas implementen pruebas piloto en proyectos de software en fase de desarrollo. Por ejemplo, podrían aplicar Fuzzy Logic en la clasificación de defectos reportados en sistemas de gestión de incidencias como JIRA o Bugzilla, evaluando su efectividad en la toma de decisiones sobre correcciones de código. Asimismo, la combinación de Machine Learning y Fuzzy Logic podría generar modelos híbridos capaces de identificar problemas de calidad con mayor precisión y adaptabilidad a distintos entornos de desarrollo.

Si bien este estudio ha identificado tendencias clave en la evaluación de la calidad del software, aún existen áreas que requieren exploración adicional. Una de las principales limitaciones detectadas es la falta de validaciones empíricas en la aplicación de modelos predictivos emergentes en entornos industriales. Futuras investigaciones podrían centrarse en la implementación de estos enfoques en casos de estudio específicos, midiendo su impacto en términos de reducción de errores y optimización del tiempo de desarrollo.

Además, el análisis bibliométrico realizado se basa en publicaciones indexadas en Scopus y Web of Science, lo que garantiza la calidad de los estudios analizados, pero también puede generar sesgos al excluir literatura gris o investigaciones en bases especializadas como IEEE Xplore o ACM Digital Library. Futuros estudios podrían ampliar la cobertura de fuentes para obtener una visión más integral del estado del arte en calidad del software.

Otra línea de investigación prometedora es la exploración de modelos de inteligencia artificial explicable (XAI) en la evaluación de calidad del software. Una de las principales limitaciones de los modelos de aprendizaje automático es su falta de interpretabilidad, por lo que estudios futuros podrían enfocarse en desarrollar técnicas que permitan explicar las decisiones

tomadas por estos modelos en términos comprensibles para los desarrolladores y gerentes de calidad.

Finalmente, se recomienda continuar explorando la relación entre Software Metrics y Fuzzy Logic, ya que la bibliometría ha identificado un aumento en la intersección de estos conceptos. La creación de nuevas métricas híbridas que combinen criterios cuantitativos tradicionales con evaluaciones basadas en lógica difusa podría ser una contribución clave para la mejora de los procesos de aseguramiento de calidad en la industria del software.

Impacto Industrial de las Tendencias Emergentes

Las tendencias emergentes identificadas en el presente estudio, tales como Fuzzy Logic, Machine Learning, Software Reliability Growth Models (SRGM), Project Management aplicado a calidad, Genetic Algorithms, Failure Prediction y Quality Metrics desde requisitos, están transformando los procesos de desarrollo y aseguramiento de calidad del software, especialmente en sectores donde la confiabilidad y eficiencia son cruciales.

En el ámbito financiero, el aprendizaje automático ha introducido una transformación significativa en la predicción de fallos en plataformas digitales, lo que ha resultado en una reducción de los riesgos de interrupciones y en una optimización del mantenimiento. Diversos estudios han evidenciado que los modelos en cuestión exhiben un rendimiento superior en términos de precisión en comparación con los enfoques estadísticos convencionales. Este hallazgo ha motivado su adopción por parte de empresas tecnológicas caracterizadas por su agilidad operativa.

En el ámbito de la industria de la salud, se ha implementado el sistema de Fuzzy Logic para gestionar la incertidumbre inherente al desarrollo de software embebido en dispositivos médicos. Esta integración ha demostrado mejorar la capacidad de predicción de defectos y, por

consiguiente, reducir los riesgos clínicos asociados. En el ámbito de la manufactura y la automatización industrial, los SRGM han desempeñado un papel crucial en la identificación temprana de defectos y la optimización del mantenimiento en sectores como la robótica y el control de procesos.

La implementación de prácticas de gestión de proyectos en el ámbito de la calidad ha adquirido una relevancia significativa en entornos caracterizados por una alta presión temporal y la utilización de marcos de trabajo como Scrum y DevOps, los cuales se encuentran alineados con métricas de calidad desde las etapas iniciales del proceso de desarrollo. En el ámbito de la inteligencia artificial y el análisis de grandes volúmenes de datos, los algoritmos genéticos han adquirido una notable relevancia como herramientas de optimización. Su implementación ha demostrado mejorar la eficiencia de los sistemas y facilitar la detección temprana de errores.

En el ámbito de las telecomunicaciones y los sistemas embebidos, la predicción de fallos se ha erigido como una prioridad imperante. Esta estrategia se implementa con el propósito de mitigar fallos en la producción y optimizar la experiencia del usuario. En última instancia, la implementación de métricas de calidad a partir de los requisitos ha facilitado un mayor control y alineación en proyectos de software de gran envergadura, garantizando el cumplimiento de las especificaciones funcionales y los atributos de calidad.

Las tendencias actuales en la calidad del software, como Machine Learning, Fuzzy Logic y los Software Reliability Growth Models (SRGM), pueden beneficiarse de una integración más profunda con los estándares internacionales establecidos por ISO/IEC 25010, que define un marco de referencia para evaluar la calidad del software. Estos enfoques emergentes, al centrarse en la mejora de la fiabilidad, la eficiencia y la mantenibilidad, se alinean estrechamente con los atributos clave de este estándar. Por ejemplo, el uso de Machine Learning en la predicción de

fallos podría optimizar la fiabilidad del software, un atributo crítico en ISO/IEC 25010, al permitir la detección temprana de defectos y la asignación proactiva de recursos.

Igualmente, la Lógica Difusa (Fuzzy Logic), al manejar incertidumbre en el procesamiento de datos, puede mejorar la usabilidad y la seguridad de sistemas en entornos complejos, alineándose con la necesidad de software fácil de usar y seguro. Por último, los modelos de crecimiento de la confiabilidad del software (SRGM) son fundamentales para fortalecer la mantenibilidad y la portabilidad, atributos esenciales en la mejora continua de software en proyectos que requieren ciclos de vida prolongados. En este sentido, integrar estas tendencias emergentes con los estándares ISO/IEC 25010 garantiza una evaluación holística y coherente de la calidad del software, permitiendo su alineación con las mejores prácticas internacionales.

Desafíos y Oportunidad Emergentes

Un desafío clave identificado es la falta de validación empírica de herramientas avanzadas como la lógica difusa en entornos reales. Si bien estas metodologías han demostrado ser prometedoras en estudios académicos, su implementación práctica requiere de ensayos adicionales para asegurar su eficacia en diversos contextos industriales. Por lo tanto, se considera esencial el desarrollo de proyectos piloto que integren estas tecnologías en entornos controlados y reales, como en los sectores de la salud y la automoción, donde la gestión de la incertidumbre resulta crucial. En el ámbito de la investigación científica y tecnológica, la implementación de proyectos específicos se erige como una estrategia fundamental para avanzar en el conocimiento y la innovación. En este sentido, la evaluación del rendimiento de la lógica difusa en la predicción de fallos o en la mejora de la fiabilidad de los sistemas embebidos emerge como un campo de interés. Este análisis, acompañado de un riguroso seguimiento, permite no solo

comprender el impacto de la lógica difusa en la eficiencia operativa, sino también identificar oportunidades para reducir los errores en los sistemas.

Por otro lado, la aparición de conceptos innovadores como Genetic Algorithms y Failure Prediction están generando nuevas oportunidades para optimizar la precisión en la predicción de defectos, particularmente cuando se integran con técnicas de aprendizaje automático (Gandhi et al., 2022). Para implementar de manera efectiva estas herramientas, se recomienda la creación de marcos de trabajo específicos para su integración en los procesos de desarrollo de software, especialmente en sectores críticos como la tecnología financiera y la seguridad, donde las deficiencias pueden acarrear consecuencias graves. La implementación de un plan concreto podría incluir la capacitación de equipos en el uso de algoritmos específicos, junto con la creación de prototipos de sistemas de predicción de fallos que utilicen datos históricos para entrenar modelos predictivos más robustos.

Los resultados obtenidos en el presente estudio posicionan a la presente investigación en el contexto de las tendencias globales en ingeniería de software, puesto que ponen de manifiesto la importancia de la innovación continua en la evaluación de la calidad del software. La transición hacia enfoques más integrales refleja un esfuerzo por alinearse con las demandas actuales de sistemas altamente confiables y adaptativos, que resultan esenciales en sectores como la salud, la seguridad y la tecnología financiera. Este posicionamiento no solo refuerza la relevancia de los hallazgos, sino que también establece un marco para futuras investigaciones que aborden las brechas identificadas y fortalezcan la conexión entre la teoría y la práctica en este campo en evolución. Para lograr este objetivo, resulta imperativo fomentar la colaboración entre los sectores académico y industrial, mediante la creación de espacios de innovación

conjunta que posibiliten la evaluación de la viabilidad de tales herramientas en entornos industriales reales y el avance hacia su adopción a gran escala.

Limitaciones

Este estudio presenta ciertas limitaciones que pueden influir en la interpretación de los resultados. Una de las principales restricciones es la dependencia de Scopus y Web of Science, lo que excluye investigaciones relevantes de bases especializadas como IEEE Xplore y ACM Digital Library, así como literatura gris proveniente de informes técnicos y estudios industriales. Esta limitación podría generar un sesgo en la cobertura, restringiendo la identificación de enfoques emergentes que no han sido ampliamente citados en la literatura académica.

En particular, las áreas emergentes, tales como el aprendizaje automático aplicado a las pruebas de software o el uso de lógica difusa en entornos industriales, tienden a ser publicadas en congresos especializados (por ejemplo, ICSE, FSE, ASE), cuyos artículos pueden no estar completamente representados en las bases de datos seleccionadas. Esta exclusión tiene el potencial de limitar la identificación de aplicaciones prácticas recientes y el rastreo de prototipos o soluciones técnicas en etapas experimentales.

No obstante, se considera que el análisis realizado conserva su validez académica y su relevancia científica, al centrarse en publicaciones con alto nivel de revisión por pares y visibilidad internacional. Se plantea la posibilidad de que investigaciones futuras puedan complementar el presente estudio mediante la realización de revisiones sistemáticas que integren literatura gris o bases de datos más especializadas, ampliando así el espectro de análisis y permitiendo una visión más completa de las aplicaciones emergentes en calidad de software en etapas tempranas.

Además, el análisis se centró en publicaciones en inglés, lo que puede excluir estudios valiosos publicados en otros idiomas. Esta restricción introduce un posible sesgo geográfico, ya que investigaciones provenientes de regiones donde la calidad del software es un área emergente pueden no haber sido consideradas. Como resultado, la visión obtenida puede estar dominada por tendencias investigativas de países con mayor producción científica en este campo.

Desde el punto de vista metodológico, el estudio se basa en un enfoque bibliométrico cuantitativo, el cual permite identificar tendencias generales, pero no profundiza en el contenido cualitativo de los artículos analizados. Asimismo, el uso de herramientas como Microsoft Excel y VOSviewer para el análisis de datos puede presentar limitaciones técnicas y sesgos de interpretación, lo que podría influir en la categorización y representación de las tendencias emergentes en la literatura. Aunque estas herramientas son útiles para la visualización de redes de coocurrencia y la identificación de patrones, no se ha profundizado en los sesgos que podrían surgir de los parámetros de búsqueda definidos, como los términos clave, la base de datos seleccionada o el período temporal abarcado.

Además, la herramienta VOSviewer, al ser un software automatizado para el análisis de coocurrencias, puede no captar ciertos matices del contexto o la relevancia semántica de los términos, lo que podría generar distorsiones en la interpretación de los resultados. La falta de un enfoque más crítico hacia estos elementos podría limitar la objetividad de los hallazgos y la exhaustividad del análisis.

Para abordar estas limitaciones, futuras investigaciones podrían ampliar la cobertura incluyendo bases de datos adicionales y explorando repositorios de literatura gris, informes técnicos y estudios de caso empresariales. Esto permitiría una visión más completa de la

evolución en la evaluación de la calidad del software, incorporando tanto la perspectiva académica como la experiencia del sector industrial.

Otro enfoque prometedor es la aplicación de minería de textos en artículos completos, en lugar de limitarse a metadatos y palabras clave. Esto facilitaría la identificación de relaciones conceptuales más profundas entre modelos predictivos, métricas de calidad y enfoques de aseguramiento del software, proporcionando un análisis más detallado de la evolución temática en esta área.

Finalmente, se recomienda complementar el análisis bibliométrico con un enfoque cualitativo, mediante entrevistas con expertos y estudios de validación en entornos industriales. Así mismo, la evaluación empírica de metodologías emergentes como Fuzzy Logic y modelos híbridos de aprendizaje automático en sectores críticos como ciberseguridad y software de misión crítica permitiría determinar su impacto real en la optimización del desarrollo y la confiabilidad del software.

Futuras Investigaciones

Como se ha puesto de manifiesto en investigaciones previas, el campo de la calidad del software en etapas tempranas de desarrollo sigue siendo un área de investigación en constante evolución, con múltiples oportunidades para estudios futuros. A pesar de la exploración exhaustiva de la eficacia de diversas herramientas y enfoques, tales como los modelos de crecimiento de la confiabilidad del software (SRGM), persiste la necesidad de un análisis más exhaustivo en el ámbito de las metodologías ágiles y DevOps. En dichas metodologías, la velocidad de iteración y los ciclos de desarrollo continuo representan factores determinantes. El objeto de futuros estudios podría enfocarse en los procedimientos a seguir para adaptar dichos

modelos a entornos de desarrollo más dinámicos, manteniendo su capacidad predictiva y mejorando la eficiencia en la entrega del software.

El presente estudio aborda la relevancia de la investigación en el ámbito del Project Management y la calidad del software, con un enfoque particular en proyectos que implementan metodologías híbridas. La integración de modelos de confiabilidad con enfoques de gestión de proyectos podría mejorar significativamente la eficiencia, optimizando tanto el tiempo de desarrollo como los recursos empleados. La inteligencia artificial (IA) emerge como un campo prometedor para la automatización de la toma de decisiones en la gestión de la calidad del software, un aspecto que podría ser objeto de investigaciones futuras.

La optimización de la calidad del software mediante algoritmos genéticos y otros métodos evolutivos es otra área que requiere mayor profundización. A pesar de que estos algoritmos han demostrado su eficacia en la resolución de problemas complejos, se requiere evidencia empírica que compare su rendimiento con el de otras técnicas, como las redes neuronales. En el ámbito de la investigación, se plantea la posibilidad de explorar la combinación de enfoques específicos con el propósito de mejorar la confiabilidad del software en sectores críticos tales como el sector de la salud y el sector aeronáutico. En estos sectores, la precisión y la confiabilidad se erigen como imperativos críticos.

En este sentido, la predicción de fallos (Failure Prediction) emerge como una tendencia que merece una mayor atención. La integración de métodos de aprendizaje profundo y el análisis de datos históricos podría optimizar la precisión y la capacidad de detección de errores en el software. Además, se considera pertinente examinar la incidencia de los factores humanos, tales como las prácticas de codificación y la experiencia del equipo, en la predicción de fallos y en la calidad general del software.

En lo que respecta a las métricas de calidad, se recomienda avanzar en la concepción de nuevas métricas que no solo evalúen aspectos técnicos, sino que también consideren la experiencia del usuario y la sostenibilidad a largo plazo del software. Para facilitar su adopción práctica y adaptarlas a las necesidades cambiantes del mercado, es necesario que estas métricas sean validadas en entornos industriales.

En última instancia, la Lógica Difusa (Fuzzy Logic) exhibe una notable capacidad para la evaluación de la calidad del software, particularmente en contextos donde la incertidumbre y la imprecisión son habituales. Se recomienda investigar la integración de esta herramienta con enfoques adicionales de inteligencia artificial para mejorar la toma de decisiones en la calidad del software y su aplicabilidad en sistemas complejos. Además, el análisis de su impacto en la confiabilidad del software en entornos de misión crítica proporcionaría valiosos aportes para sectores como la aeronáutica y la medicina.

Conclusiones

Los resultados de este análisis bibliométrico evidencian un crecimiento polinomial cuadrático del 99,12 % en las publicaciones sobre calidad del software en etapas tempranas, destacando los años 2012, 2018, 2019, 2021 y 2022 como los periodos más productivos. Este crecimiento refleja no solo el interés sostenido de la comunidad científica, sino también la creciente importancia del tema en el desarrollo tecnológico global, especialmente en un contexto donde la ingeniería de software demanda enfoques más sofisticados para garantizar la calidad desde las primeras fases del ciclo de vida del software.

En términos de producción científica, se identificaron como referentes a Malhotra R. y Smidts C., quienes destacan tanto por su productividad como por el impacto de sus contribuciones en el campo. Asimismo, autores como Jain A. y el equipo conformado por Stutzke M. y Stoddard R.W. han demostrado una alta influencia académica. Revistas de prestigio como *Journal of Information Processing Systems*, *IEEE Transactions on Reliability* e *ICIC Express Letters* se consolidan como las principales fuentes de publicación en esta área. A nivel geográfico, India y Estados Unidos lideran la producción científica, lo que pone en evidencia la concentración de recursos y esfuerzos investigativos en estas regiones. Este hallazgo subraya la necesidad de fomentar colaboraciones internacionales que permitan la diversificación del conocimiento y su aplicación en diversos contextos.

La evolución temática observada en la literatura refleja una transición desde enfoques tradicionales, como *Software Product Testing* y *Software Reliability Prediction*, hacia metodologías más avanzadas, como *Decision Analytics*, *Reliability Performance*, *Software Testing* y *Machine Learning*. Este cambio demuestra la adopción de herramientas como lógica difusa y aprendizaje automático, que permiten abordar de manera más integral los desafíos

asociados a la evaluación de calidad en las primeras etapas del desarrollo. Sin embargo, la disminución en la vigencia de conceptos como Reliability Prediction sugiere un desplazamiento hacia enfoques más especializados, lo que plantea interrogantes sobre la madurez y la evolución de ciertas metodologías en la literatura científica.

El análisis de clústeres temáticos permitió identificar áreas predominantes en la investigación actual, destacando conceptos clave como Software Metrics, Fuzzy Logic y Artificial Neural Networks. Estas líneas de estudio subrayan la importancia de las métricas tradicionales como base para nuevas metodologías, al tiempo que resaltan el potencial de herramientas emergentes para optimizar la evaluación de calidad desde etapas tempranas. Además, la identificación de palabras clave emergentes, como Project Management, Failure Prediction y Genetic Algorithms, indica una expansión del campo hacia enfoques más holísticos, que integran tanto aspectos técnicos como de gestión.

Desde una perspectiva práctica, estos resultados pueden orientar la implementación de herramientas predictivas en la industria del software, optimizando la detección de defectos y reduciendo costos de mantenimiento. Tecnologías como Machine Learning y Fuzzy Logic pueden ser integradas en sistemas de gestión de calidad, mejorando la toma de decisiones en sectores altamente exigentes como la salud y la ciberseguridad. Asimismo, la incorporación de modelos híbridos en procesos de prueba y aseguramiento de calidad podría permitir una evaluación más precisa y adaptativa a contextos dinámicos.

A nivel académico, esta investigación proporciona un marco de referencia sólido para estudios futuros sobre la evolución de los enfoques de calidad del software. Se recomienda ampliar la investigación hacia la validación empírica de modelos predictivos en entornos industriales, así como explorar el uso de minería de textos y análisis cualitativos para obtener una

comprensión más profunda de las tendencias emergentes. Estas líneas de investigación permitirán fortalecer la conexión entre los avances teóricos y su aplicabilidad en escenarios reales, asegurando que las nuevas metodologías sean efectivas en la mejora de la calidad del software en un entorno tecnológico cada vez más competitivo.

Referencias Bibliográficas

- Abu Talib, M. (2016). Towards early software reliability prediction for computer forensic tools (case study). *SpringerPlus*, 5, 1-12. <https://doi.org/10.1186/s40064-016-2539-0>
- Al Obisat, F. M., Alhalhouli, Z. T., Alrawashdeh, T. I., & Alshabatat, T. E. (2018). Review of Literature on Software Quality. *World of Computer Science & Information Technology Journal*, 8(5).
- Albrecht, A. J. (1979). Measuring application development productivity. Proceedings of the Joint SHARE/GUIDE/IBM Application Development Symposium, 10, 83-92.
- Alipour, H., & Isazadeh, A. (2008, March). Software reliability prediction based on a formal requirements specification. In *Computer Society of Iran Computer Conference* (pp. 816-820). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-89985-3_110
- Angles Medina, G. C. (2022). Desarrollo de un sistema para el diagnóstico del reencauche de neumáticos aplicando lógica difusa.
- Asraful Haque, M., & Ahmad, N. (2021). A logistic growth model for software reliability estimation considering uncertain factors. *International Journal of Reliability, Quality and Safety Engineering*, 28(05), 2150032.
- Asubiaro, T. V., & Onalapo, S. (2023). A comparative study of the coverage of African journals in Web of Science, Scopus, and CrossRef. *Journal of the Association for Information Science and Technology*, 74(7), 745-758. <https://doi.org/10.1002/asi.24758>
- Attaran, M., & Celik, B. G. (2023). Digital Twin: Benefits, use cases, challenges, and opportunities. *Decision Analytics Journal*, 6, 100165. <https://doi.org/10.1016/j.dajour.2023.100165>
- Bansal, A., & Sharma, A. (2012). Quality evaluation framework based on fuzzy logic inference system for component-based software. *International Journal of Computer Applications*, 59(5), 10-16.
- Barraza, N. R. (2013). Parameter estimation for the compound poisson software reliability model. *International Journal of Software Engineering and its Applications*, 7(1), 137-148.
- Barraza, N. R. (2015). A parametric empirical bayes model to predict software reliability growth. *Procedia Computer Science*, 62, 360-369. <https://doi.org/10.1016/j.procs.2015.08.416>

- Bei-Yang, W. (2013). Software Reliability Evaluation Approaches using Weighted Software Network. *Information Technology Journal*, 12(21), 6288.
- Besseris, G. J. (2010). Taguchi methods in software quality testing. *International Journal of Quality Engineering and Technology*, 1(3), 339-372. <https://doi.org/10.1504/IJQET.2010.034615>
- Bharathi, R., & Selvarani, R. (2015, December). A framework for the estimation of OO software reliability using design complexity metrics. In *2015 International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15)* (pp. 1-7). IEEE. doi: 10.1109/ITACT.2015.7492648.
- Boehm, B. W. (1987). Improving software productivity. *Computer*, 20(09), 43-57. <https://doi.ieeecomputersociety.org/10.1109/MC.1987.1663694>
- Boehm, B. W., & Basili, V. R. (2001). Software defect reduction top 10 list. *Computer*, 34(1), 135-137.
- Boukhelif, M., Hanine, M., & Kharmoum, N. (2023). A decade of intelligent software testing research: a bibliometric analysis. *Electronics*, 12(9), 2109. <https://doi.org/10.3390/electronics12092109>
- Briand, L. C., Basili, V. R., & Thomas, W. M. (1992). A pattern recognition approach for software engineering data analysis. *IEEE Transactions on Software Engineering*, 18(11), 931-942.
- Briand, L. C., Basili, V. R., & Thomas, W. M. (2001). A pattern recognition approach for software engineering data analysis. *IEEE Transactions on Software Engineering*, 18(11), 931-942.
- Cordeiro, A., & Freitas, A. L. (2012). Priorização de requisitos e avaliação da qualidade de software segundo a percepção dos usuários. *Ciência da Informação*, 40(2), 160-179.
- Oveisi, S., Moeini, A., Mirzaei, S., & Farsi, M. A. (2023). Software reliability prediction: A survey. *Quality and Reliability Engineering International*, 39(1), 412-453.
- Cowlessur, S. K., Pattnaik, S., & Pattanayak, B. K. (2020). A review of machine learning techniques for software quality prediction. *Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2018, Volume 2*, 537-549.
- D'Ambrogio, A., Iazeolla, G., & Mirandola, R. (2002, November). A method for the prediction of software reliability. In *Proceedings of the 6-th IASTED software engineering and applications conference (SEA2002)*, Cambridge, MA.

- Dayanandan, U., & Kalimuthu, V. (2018). A fuzzy analytical hierarchy process (FAHP) based software quality assessment model: maintainability analysis. *Int J Intell Eng Syst*, 11(4), 88-96.
- Dimri, S. C., Purohit, K. C., & Dumka, A. (2022, November). Analysis of Software Quality Enhancement through Stability Metrics Models. In *2022 International Conference on Fourth Industrial Revolution Based Technology and Practices (ICFIRTP)* (pp. 44-48). IEEE. doi: 10.1109/ICFIRTP56122.2022.10059411.
- Diwaker, C., Tomar, P., Poonia, R. C., & Singh, V. (2018). Prediction of software reliability using bio inspired soft computing techniques. *Journal of medical systems*, 42, 1-16.
<https://doi.org/10.1007/s10916-018-0952-3>
- Dubois, D., & Prade, H. (1998). Possibility theory: Qualitative and quantitative aspects. In Gabbay, D. & Smets, P. (Eds.), *Handbook of Defeasible Reasoning and Uncertainty Management Systems* (Vol. 1, pp. 31-70). Springer.
- Esnaashari, M., & Damia, A. H. (2021). Automation of software test data generation using genetic algorithm and reinforcement learning. *Expert Systems with Applications*, 183, 115446.
<https://doi.org/10.1016/j.eswa.2021.115446>
- Fenton, N. E., & Neil, M. (1999). A critique of software defect prediction models. *IEEE Transactions on software engineering*, 25(5), 675-689. doi: 10.1109/32.815326.
- Fenton, N., & Pfleeger, S. L. (1997). *Software metrics: A rigorous and practical approach* (2nd ed.). PWS Publishing.
- Fischer, H., & Senft, B. (2016). Human-Centered software engineering as a chance to ensure software quality within the digitization of human workflows. In *Human-Centered and Error-Resilient Systems Development: IFIP WG 13.2/13.5 Joint Working Conference, 6th International Conference on Human-Centered Software Engineering, HCSE 2016, and 8th International Conference on Human Error, Safety, and System Development, HESSD 2016, Stockholm, Sweden, August 29-31, 2016, Proceedings 8* (pp. 30-41). Springer International Publishing.
https://doi.org/10.1007/978-3-319-44902-9_3

- Fujii, T., Dohi, T., & Fujiwara, T. (2011, May). Towards quantitative software reliability assessment in incremental development processes. In *Proceedings of the 33rd International Conference on Software Engineering* (pp. 41-50). <https://doi.org/10.1145/1985793.1985800>
- Gandhi, P., Khan, M. Z., Sharma, R. K., Alhazmi, O. H., Bhatia, S., & Chakraborty, C. (2022). Software Reliability Assessment Using Hybrid Neuro-Fuzzy Model. *Computer Systems Science & Engineering*, 41(3). <https://doi.org/10.32604/csse.2022.019943>
- García-Mireles, G. A., Moraga, M. Á., García, F., & Piattini, M. (2012). Towards the harmonization of process and product oriented software quality approaches. In *Systems, Software and Services Process Improvement: 19th European Conference, EuroSPI 2012, Vienna, Austria, June 25-27, 2012. Proceedings 19* (pp. 133-144). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31199-4_12
- Gokhale, S. S. (2005, January). Variance expressions for software reliability growth models. In *Annual Reliability and Maintainability Symposium, 2005. Proceedings.* (pp. 628-633). IEEE. doi: 10.1109/RAMS.2005.1408434.
- Goyal, S., & Bhatia, P. K. (2020). Comparison of machine learning techniques for software quality prediction. *International Journal of Knowledge and Systems Science (IJKSS)*, 11(2), 20-40. DOI: 10.4018/IJKSS.2020040102
- Gupta, C., & Gupta, V. (2024). Enhancing bug allocation in software development: a multi-criteria approach using fuzzy logic and evolutionary algorithms. *PeerJ Computer Science*, 10, e2111. <https://doi.org/10.7717/peerj-cs.2111>
- Haque, M. A., & Ahmad, N. (2022). A software reliability model using fault removal efficiency. *Journal of Reliability and Statistical Studies*, 15(2), 459-472. doi: 10.13052/jrss0974-8024.1523
- Hou, Y. F., Huang, C. Y., & Fang, C. C. (2022). Using the methods of statistical data analysis to improve the trustworthiness of software reliability modeling. *IEEE Access*, 10, 25358-25375. doi: 10.1109/ACCESS.2022.3154103.
- Hsu, C. J., & Huang, C. Y. (2008, September). Integrating path testing with software reliability estimation using control flow graph. In *2008 4th IEEE International Conference on Management of Innovation and Technology* (pp. 1234-1239). IEEE. doi: 10.1109/ICMIT.2008.4654546.

- Hu, Q. P., Xie, M., & Ng, S. H. (2006). Software reliability prediction improvement with prior information incorporated. In *12th ISSAT International Conference on Reliability and Quality in Design* (pp. 303-307).
- Ibraigheeth, M., & Fadzli, S. A. (2018). Software reliability prediction in various software development stages. *Journal of theoretical and applied information technology*, 96(7).
- Jabeen, G., Luo, P., & Afzal, W. (2019). An improved software reliability prediction model by using high precision error iterative analysis method. *Software Testing, Verification and Reliability*, 29(6-7), e1710.
- Jain, A. (2012). FAULT PREDICTION USING STATISTICAL AND MACHINE LEARNING METHODS FOR IMPROVING SOFTWARE QUALITY (Doctoral dissertation).
- Jain, R., & Sharma, A. (2019). Assessing software reliability using genetic algorithms. *The journal of engineering research [tjer]*, 16(1), 11-17. <https://doi.org/10.24200/tjer.vol16iss1pp11-17>
- Kan, S. H. (2002). Metrics and models in software quality engineering (2nd ed.). Addison-Wesley.
- Karkalas, S., & Gutierrez-Santos, S. (2014, July). Enhanced javascript learning using code quality tools and a rule-based system in the flip exploratory learning environment. In *2014 IEEE 14th International Conference on Advanced Learning Technologies* (pp. 84-88). IEEE. doi: 10.1109/ICALT.2014.35.
- Khalid, S., Rasheed, U., & Abbas, M. (2021, November). A Model Driven Framework for Standardizing Requirement Elicitation by Quantifying Software Quality Factor. In *2021 International Conference on Innovative Computing (ICIC)* (pp. 1-6). IEEE. doi: 10.1109/ICIC53490.2021.9693054.
- Kumar, C., & Yadav, D. K. (2013). Software quality modeling using metrics of early artifacts. <https://doi.org/10.1049/cp.2013.2285>
- Kumar, P. S., Nayak, J., & Behera, H. S. (2022). Model-based Software Defect Prediction from Software Quality Characterized Code Features by using Stacking Ensemble Learning. *Journal of Engineering Science & Technology Review*, 15(2). doi:10.25103/jestr.152.17
- Lee, D. H., Chang, I. H., & Pham, H. (2020). Software reliability model with dependent failures and SPRT. *Mathematics*, 8(8), 1366.

- Lee, W., Lee, J. K., & Baik, J. (2011, July). Software reliability prediction for open source software adoption systems based on early lifecycle measurements. In *2011 IEEE 35th Annual Computer Software and Applications Conference* (pp. 366-371). IEEE. doi: 10.1109/COMPSAC.2011.55.
- Lin, C. T., Tang, K. W., Chang, J. R., & Huang, C. Y. (2010, December). An investigation into whether the NHPP framework is suitable for software reliability prediction and estimation. In *2010 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 626-630). IEEE. doi: 10.1109/IEEM.2010.5674524.
- Lo, J. H. (2010, August). A study of applying support vector machine and Genetic Algorithm to software reliability forecasting. In *2010 International Conference on Electronics and Information Engineering* (Vol. 1, pp. V1-324). IEEE. doi: 10.1109/ICEIE.2010.5559867.
- Malhotra, R. (2015). A systematic review of machine learning techniques for software fault prediction. *Applied Soft Computing*, 27, 504-518.
- Malhotra, R. (2017, February). Software quality predictive modeling: An effective assessment of experimental data. In *Proceedings of the 10th Innovations in Software Engineering Conference* (pp. 215-216).
- Malhotra, R., & Jain, A. (2012). Fault prediction using statistical and machine learning methods for improving software quality. *Journal of Information Processing Systems*, 8(2), 241-262.
<http://dx.doi.org/10.3745/JIPS.2012.8.2.241>
- Marandi, A. K., & Khan, D. A. (2015). An impact of linear regression models for improving the software quality with estimated cost. *Procedia Computer Science*, 54, 335-342.
<https://doi.org/10.1016/j.procs.2015.06.039>
- Marandi, A. K., & Khan, D. A. (2015, March). Statistical analysis of defect removal effectiveness to improve the software quality and reducing the estimated cost. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 509-513). IEEE.
- Marandi, A. K., & Khan, D. A. (2017). An approach of statistical methods for improve software quality and cost minimization. *International Journal of Applied Engineering Research*, 12(6), 1054-1061.
- Masood, M. H., & Khan, M. J. (2018). Early software quality prediction based on software requirements specification using fuzzy inference system. In *Intelligent Computing Methodologies: 14th*

- International Conference, ICIC 2018, Wuhan, China, August 15-18, 2018, Proceedings, Part III* 14 (pp. 722-733). Springer International Publishing. https://doi.org/10.1007/978-3-319-95957-3_75
- McCabe, T. J. (1976). A complexity measure. *IEEE Transactions on Software Engineering*, SE-2(4), 308-320. doi: 10.1109/TSE.1976.233837
- Mkaouer, M. W., Kessentini, M., Ouni, A., McMillan, C., & Deb, K. (2016). On the use of many-objective evolutionary optimization for software refactoring. *IEEE Transactions on Software Engineering*, 42(4), 300-317.
- Mohanta, S., Vinod, G., & Mall, R. (2011). A technique for early prediction of software reliability based on design metrics. *International Journal of System Assurance Engineering and Management*, 2, 261-281.
- Mutha, C., & Smidts, C. (2018). Basis for non-propagation domains, their transformations and their impact on software reliability. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 232(6), 535-558. <https://doi.org/10.1177/1748006X17744380>
- Nagappan, N., Murphy, B., & Basili, V. (2008, May). The influence of organizational structure on software quality: an empirical case study. In *Proceedings of the 30th international conference on Software engineering* (pp. 521-530). <https://doi.org/10.1145/1368088.1368160>
- Nair, T. G., & Selvarani, R. (2012). Defect proneness estimation and feedback approach for software design quality improvement. *Information and software technology*, 54(3), 274-285.
- Nakahara, H., Monden, A., & Yücel, Z. (2021, November). A simulation model of software quality assurance in the software lifecycle. In *2021 IEEE/ACIS 22nd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* (pp. 236-241). IEEE.
- Nakamichi, K., Ohashi, K., Namba, I., Yamamoto, R., Aoyama, M., Joeckel, L., ... & Heidrich, J. (2020, August). Requirements-driven method to determine quality characteristics and measurements for machine learning software and its evaluation. In *2020 IEEE 28th International Requirements Engineering Conference (RE)* (pp. 260-270). IEEE. doi: 10.1109/RE48521.2020.00036.

- Nam, J., & Kim, S. (2015). CLAMI: Defect prediction on unlabeled datasets. Proceedings of the 30th IEEE/ACM International Conference on Automated Software Engineering (ASE'15), 452-463.
- Neha, Verma, V., Tandon, A., & Aggarwal, A. G. (2021). Software reliability allocation incorporating Pythagorean fuzzy theory and AHP. *International Journal of Industrial and Systems Engineering*, 38(1), 19-34. <https://doi.org/10.1504/IJISE.2021.115128>
- Okumoto, K. (1985). A statistical method for software quality control. *IEEE Transactions on Software Engineering*, (12), 1424-1430. doi: 10.1109/TSE.1985.232178.
- Olague, H. M., Etkorn, L. H., Gholston, S., & Quattlebaum, S. (2007). Empirical validation of three software metrics suites to predict fault-proneness of object-oriented classes developed using highly iterative or agile software development processes. *IEEE Transactions on software Engineering*, 33(6), 402-419. doi: 10.1109/TSE.2007.1015.
- Pan, C., Lu, M., Zhang, H., & Xu, B. (2018, July). Qualitative Software Reliability Requirements: Concept, Classification and Practical Elicitation Methods. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 164-171). IEEE. doi: 10.1109/QRS-C.2018.00040.
- Pargaonkar, S. (2021). Quality and Metrics in Software Quality Engineering. *Journal of Science & Technology*, 2(1), 62-69.
- Pargaonkar, S. (2023). Cultivating Software Excellence: The Intersection of Code Quality and Dynamic Analysis in Contemporary Software Development within the Field of Software Quality Engineering. *International Journal of Science and Research (IJSR)*, 12(9), 10-13. DOI: 10.21275/SR23829092346
- Park, J., Kim, H. J., Shin, J. H., & Baik, J. (2012, June). An embedded software reliability model with consideration of hardware related software failures. In *2012 IEEE Sixth International Conference on Software Security and Reliability* (pp. 207-214). IEEE.
- Park, S. K. (2022). Comparative Analysis on the Reliability Attributes of Finite Failure NHPP Software Reliability Model with Exponential Distribution Characteristics. *Journal of Theoretical and Applied Information Technology*, 100(18), 5220-5231.

- Peng, W., Yao, L., & Miao, Q. (2009, July). An approach of software quality prediction based on relationship analysis and prediction model. In *2009 8th International Conference on Reliability, Maintainability and Safety* (pp. 713-717). IEEE. doi: 10.1109/ICRMS.2009.5270097.
- Perkins, J. A., & Gorzela, R. S. (1987, March). Programming paradigms involving exceptions: a software quality approach. In *Proceedings of the Joint Ada conference fifth national conference on Ada technology and fourth Washington Ada Symposium* (pp. 142-150).
- Radjenović, D., Heričko, M., Torkar, R., & Živkovič, A. (2013). Software fault prediction metrics: A systematic literature review. *Information and Software Technology*, 55(8), 1397-1418.
- Rai, A., Choudhury, T., Sharma, S., & Ting, K. C. (2017, December). An efficient method to predict software quality using soft computing techniques. In *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)* (pp. 347-353). IEEE.
- Rana, R., Staron, M., Berger, C., Hansson, J., Nilsson, M., Törner, F., ... & Höglund, C. (2014). Selecting software reliability growth models and improving their predictive accuracy using historical projects data. *Journal of Systems and Software*, 98, 59-78.
- Rani, P., & Mahapatra, G. S. (2019). A novel approach of NPSO on dynamic weighted NHPP model for software reliability analysis with additional fault introduction parameter. *Heliyon*, 5(7).
<https://doi.org/10.1016/j.heliyon.2019.e02082>
- Reddivari, S., & Raman, J. (2019, July). Software quality prediction: an investigation based on machine learning. In *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)* (pp. 115-122). IEEE.
- Reddy, Y. G., & Prasanth, Y. (2022). A Hybrid Quartile deviation-based Support vector regression model for software reliability datasets. *International Journal of Advanced Computer Science and Applications*, 13(6). DOI:10.14569/IJACSA.2022.0130694
- Renu Deepti, S., Srivani, B., & Sandhya, N. (2018). Implementation of Power Law Model for Assessing Software Reliability. In *Proceedings of the Second International Conference on Computational Intelligence and Informatics: ICCII 2017* (pp. 599-608). Springer Singapore.
https://doi.org/10.1007/978-981-10-8228-3_55

- Rizvi, S. W. A., Singh, V. K., & Khan, R. A. (2016). Fuzzy logic based software reliability quantification framework: early stage perspective (FLSRQF). *Procedia Computer Science*, 89, 359-368.
- Ronchieri, E., Canaparo, M., & Salomoni, D. (2014). A software quality model by using discriminant analysis predictive technique. *Journal of Integrated Design and Process Science*, 18(4), 25-29.
DOI: [10.3233/jid-2014-0016](https://doi.org/10.3233/jid-2014-0016)
- Sahu, K., & Srivastava, R. K. (2018). Soft computing approach for prediction of software reliability. *Neural Netw*, 17, 19.
- Sahu, K., Alzahrani, F. A., Srivastava, R. K., & Kumar, R. (2021). Evaluating the impact of prediction techniques: Software reliability perspective. *Computers, Materials & Continua*, 67(2).
<http://dx.doi.org/10.32604/cmc.2021.014868>
- Saleh, I. A., H AL_Bayati, A., & Hadi Thanoon, K. (2020). Measure the software quality based on grasshopper optimization algorithm. *International Journal of Computing and Digital Systems*, 10, 2-8.
- San, K. K., Washizaki, H., Fukazawa, Y., Honda, K., Taga, M., & Matsuzaki, A. (2019, October). Dc-srgm: Deep cross-project software reliability growth model. In *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 61-66). IEEE. doi: 10.1109/ISSREW.2019.00044.
- San, K. K., Washizaki, H., Fukazawa, Y., Honda, K., Taga, M., & Matsuzaki, A. (2021). Deep cross-project software reliability growth model using project similarity-based clustering. *Mathematics*, 9(22), 2945. <https://doi.org/10.3390/math9222945>
- Shepperd, M. (1990). Early life-cycle metrics and software quality models. *Information and Software Technology*, 32(4), 311-316.
- Shi, Y., Li, M., Arndt, S., & Smidts, C. (2017). Metric-based software reliability prediction approach and its application. *Empirical Software Engineering*, 22, 1579-1633. <https://doi.org/10.1007/s10664-016-9425-9>
- Shrivastava, A. K., Sharma, R., & Pham, H. (2023). Software reliability and cost models with warranty and life cycle. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 237(1), 166-179.

- Singh, L., Rajput, H., Vinod, G., & Tripathi, A. K. (2016). Computing transition probability in Markov chain for early prediction of software reliability. *Quality and Reliability Engineering International*, 32(3), 1253-1263.
- Singh, V., Kumar, V., & Singh, V. B. (2023). A hybrid novel fuzzy AHP-Topsis technique for selecting parameter-influencing testing in software development. *Decision Analytics Journal*, 6, 100159. <https://doi.org/10.1016/j.dajour.2022.100159>
- Singh, Y., Kaur, A., & Malhotra, R. (2009). Prediction of software quality model using gene expression programming. In *Product-Focused Software Process Improvement: 10th International Conference, PROFES 2009, Oulu, Finland, June 15-17, 2009. Proceedings 10* (pp. 43-58). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02152-7_5
- Smidts, C., Stutzke, M., & Stoddard, R. W. (1998). Software reliability modeling: an approach to early reliability prediction. *IEEE Transactions on Reliability*, 47(3), 268-278. doi: 10.1109/24.740500.
- Standish Group. (2020). CHAOS Report: Beyond Infinity. Standish Group International.
- Sudharson, D., & Prabha, D. (2019). A novel machine learning approach for software reliability growth modelling with pareto distribution function. *Soft Computing*, 23(18), 8379-8387.
- Suresh, Y. (2015, October). Software quality assurance for object-oriented systems using meta-heuristic search techniques. In *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)* (pp. 441-448). IEEE. doi: 10.1109/ICATCCT.2015.7456924.
- Tripathi, R., & Mall, R. (2005, December). Early stage software reliability and design assessment. In *12th Asia-Pacific Software Engineering Conference (APSEC'05)* (pp. 8-pp). IEEE. doi: 10.1109/APSEC.2005.63.
- Van Eck, N., & Waltman, L. (2010). Software survey: VOSviewer, a computer program for bibliometric mapping. *scientometrics*, 84(2), 523-538. <https://doi.org/10.1007/s11192-009-0146-3>
- Villegas, N. M., Müller, H. A., Tamura, G., Duchien, L., & Casallas, R. (2011, May). A framework for evaluating quality-driven self-adaptive software systems. In *Proceedings of the 6th international symposium on Software engineering for adaptive and self-managing systems* (pp. 80-89). <https://doi.org/10.1145/1988008.1988020>

- Voas, J. M. (1998). The challenges of using COTS software in component-based development. *Computer*, 31(06), 44-45. [10.1109/MC.1998.683006](https://doi.org/10.1109/MC.1998.683006)
- Wang, L., Hu, Q., & Liu, J. (2016). Software reliability growth modeling and analysis with dual fault detection and correction processes. *IIE transactions*, 48(4), 359-370.
- Wei, B., Jin, Z., Zowghi, D., & Yin, B. (2012, July). Automated reasoning with goal tree models for software quality requirements. In *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops* (pp. 373-378). IEEE. doi: 10.1109/COMPSACW.2012.73.
- Wohlin, C., Wesslén, A., & Runeson, P. (1996). Software reliability estimations through usage analysis of specifications and designs. *International Journal of Reliability, Quality and Safety Engineering*, 3(02), 101-117. <https://doi.org/10.1142/S0218539396000089>
- Yang, L., Zhang, H., Shen, H., Huang, X., Zhou, X., Rong, G., & Shao, D. (2021). Quality assessment in systematic literature reviews: A software engineering perspective. *Information and Software Technology*, 130, 106397. <https://doi.org/10.1016/j.infsof.2020.106397>
- YANG, T. (2022). Performance Analysis on the Reliability Attributes of NHPP Software Reliability Model Applying Exponential and Inverse-Exponential Lifetime Distribution. *Journal of Theoretical and Applied Information Technology*, 100(22), 6645-6656.
- Yaremchuk, S., & Kharchenko, V. (2022). Similarity Based Fuzzy Logic Modeling for the Software Reliability Assessment. In *COLINS* (pp. 1557-1568).
- Yin, M. L., Peterson, J., & Arellano, R. R. (2004, January). Software complexity factor in software reliability assessment. In *Annual Symposium Reliability and Maintainability, 2004-RAMS* (pp. 190-194). IEEE. doi: 10.1109/RAMS.2004.1285446.
- Yusri, R., Edriati, S., & Yuhendri, R. (2020). Pelatihan Microsoft Office Excel Sebagai Upaya Peningkatan Kemampuan Mahasiswa Dalam Mengolah Data. *Rangkiang: Jurnal Pengabdian Pada Masyarakat*, 2(1), 32-37. <https://doi.org/10.22202/JR.2020.V1i2.3914>
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338-353.
- Zhao, Y., Hu, Y., & Gong, J. (2021, October). Research on international standardization of software quality and software testing. In *2021 IEEE/ACIS 20th International Fall Conference on Computer and Information Science (ICIS Fall)* (pp. 56-62). IEEE. doi: 10.1109/ICISFall51598.2021.9627426

赵靖, 刘宏伟, 崔刚, & 杨孝宗. (2005). Early stage software reliability estimation with stochastic reward nets. *东华大学学报: 英文版*, 22(3), 33-36.