

# EL PARADIGMA DE LA INGENIERÍA DE CALIDAD DE SOFTWARE ASISTIDA POR IA GENERATIVA: EFICIENCIA, AUTOMATIZACIÓN Y TRANSFORMACIÓN SECTORIAL

**Ing. Kevin Fernandez**

QA mid & Black belt six sigma  
 Kevin.fernandeze@cpic.cr  
 Cartago, COSTA RICA

## RESUMEN

La transición de la programación tradicional a la programación asistida por Inteligencia Artificial Generativa (GenAI) ha forzado una evolución paralela en el aseguramiento de calidad (QA). Este artículo analiza el impacto de la IA en la validación de software, comparando los métodos manuales frente a los asistidos por algoritmos, y explora aplicaciones críticas en los sectores de banca, finanzas y comercio. Finalmente, se abordan los límites éticos y el alcance real de estas herramientas en la mejora de procesos.

## ABSTRACT

The transition from traditional programming to Generative Artificial Intelligence (GenAI) assisted programming has forced a parallel evolution in Software Quality Assurance (QA). This article analyzes the impact of AI on software validation, comparing manual methods against algorithm-assisted approaches, and explores critical applications within the banking, finance, and retail sectors. Finally, it addresses the ethical boundaries and the actual scope of these tools in process improvement.

*Palabras clave: Inteligencia Artificial Generativa (GenAI), Aseguramiento de Calidad (QA), automatización cognitiva, SDLC, FinTech, pruebas de software, auto-curación, ética de la IA, CI/CD.*

*Keywords: Generative AI, Quality Assurance (QA), Cognitive Automation, SDLC, FinTech, Software Testing, Self-healing, AI Ethics, CI/CD.*

## I. INTRODUCCIÓN: DE LA CODIFICACIÓN A LA VALIDACIÓN

Como se expone en la literatura reciente sobre programación asistida [1], el uso de herramientas como GitHub Copilot o ChatGPT no solo acelera la escritura de código, sino que redefine cómo entendemos la lógica del programa. En la ingeniería de calidad, esto se traduce en una capacidad sin precedentes para generar casos de prueba, scripts de automatización y detección temprana de vulnerabilidades. La IA ya no es solo un oráculo de consulta, sino un colaborador activo en el ciclo de vida de desarrollo de software (SDLC).



Fig. 1. El ciclo de vida del defecto con Inteligencia Artificial Generativa (GenAI): Optimización del SDLC con Inteligencia Artificial (IA)

## II. IMPACTO: CALIDAD MANUAL VS CALIDAD ASISTIDA POR IA

La diferencia entre el enfoque tradicional y el asistido por IA marca un punto de inflexión en el Time-to-Market.

### A. QA Manual: El Factor humano y sus límites

El QA manual es de suma importancia para la experiencia de usuario (UX), pero presenta deficiencias en:

**Escalabilidad:** Imposible de ejecutar en entornos de integración continua (CI/CD) de alta frecuencia.

**Error humano:** Fatiga en pruebas de regresión repetitivas.

**Costo:** Alto consumo de horas-hombre para tareas de bajo valor intelectual.

Tabla 1. Tabla comparativa: QA Manual vs. QA Asistido por IA

Característica	QA Manual Tradicional	QA Asistido por GenAI
Velocidad de ejecución	Limitada por la jornada humana	Síncrona con pipelines de CI/CD
Mantenimiento de scripts	Manual y propenso a errores	Self-healing (Auto-curación)
Datos de prueba	Datos estáticos o limitados	Generación sintética masiva
Cobertura de seguridad	Muestreo periódico	Análisis continuo de vulnerabilidades
Costo a largo plazo	Escala lineal con el personal	Escala marginal tras la implementación

### B. QA Asistido por IA: La automatización cognitiva

A diferencia de la automatización basada en reglas (scripts rígidos), la IA ofrece:

**Pruebas de auto-curación (self-healing):** Si un elemento de la interfaz cambia (ej. un botón cambia de ID), la IA identifica el cambio y ajusta el script automáticamente, reduciendo el mantenimiento en un 70 %.

**Generación sintética de datos:** Creación de millones de registros de prueba que imitan el comportamiento real sin comprometer datos sensibles.

**Detección de anomalías:** Identificación de patrones de error que pasarían desapercibidos para el ojo humano en grandes volúmenes de logs.



Fig. 2. Impacto en el Time-to-Market antes y después de la autocuración de scripts.

## III. APLICACIONES REALES: SECTORES ESTRATÉGICOS

### A. Sector banca y finanzas

En un entorno donde el error cuesta millones y la seguridad es innegociable, la IA actúa como un filtro de alta precisión.

**Pruebas de seguridad:** La IA analiza el código en busca de patrones de inyección SQL o vulnerabilidades de día cero antes del despliegue.

**Validación de transacciones:** Simulación de picos de carga masivos para probar la resiliencia de los núcleos bancarios ante eventos como el "Viernes Negro" o cierres fiscales.

**Cumplimiento regulatorio:** Verificación automatizada de que cada flujo de pago cumple con normativas internacionales (ISO 20022).

### B. Sector comercial y retail

La calidad aquí se mide en la fluidez del "Customer Journey".

**Pruebas de regresión multicanal:** La IA valida que una promoción se aplique correctamente de forma simultánea en la web, aplicación móvil y puntos de venta físicos.

**Pruebas A/B automatizadas:** Algoritmos que prueban miles de variaciones de una interfaz para determinar cuál genera menos errores de conversión y mejor respuesta del sistema.



Fig. 3. Aplicaciones en sectores estratégicos banca y retail

## IV. MEJORA DE PROCESOS MEDIANTE IA GENERATIVA

Siguiendo los principios de optimización tecnológica, la integración de GenAI en las pruebas permite:

**Refactorización de pruebas legado:** Convertir scripts antiguos en lenguajes modernos de forma automatizada.

**Documentación de calidad:** Generación automática de reportes de errores técnicos explicados en lenguaje natural para los stakeholders financieros, lo que facilita la toma de decisiones basada en riesgos.

## V. CONCLUSIONES

**Alcance de la IA:** La IA no reemplaza al ingeniero de calidad; lo eleva a un rol de arquitecto de pruebas. El alcance actual permite la automatización de la ejecución y el diseño básico, pero el pensamiento estratégico y la validación ética siguen siendo prerrogativas humanas.

**Desafíos éticos:** Si la IA se entrena con código defectuoso o sesgado, las pruebas que genere heredarán esos errores, creando una falsa sensación de seguridad.

**Propiedad intelectual:** Como se discute en [1], el código generado puede contener fragmentos sujetos a licencias. El QA debe validar que las herramientas de IA no introduzcan riesgos legales en el software financiero.

**Transparencia:** En finanzas, “la IA dijo que estaba bien” no es una defensa válida ante un regulador. La IA explicable es el siguiente paso necesario para auditar cómo se validó el software.

## REFERENCIAS

[1] D. Zingaro y L. Porter, Learn AI-Assisted Python Programming, 2.<sup>a</sup> ed., Shelter Island, NY: Manning Publications, 2024.

[2] R. S. Pressman y B. R. Maxim, Software Engineering: A Practitioner’s Approach, 9.<sup>a</sup> ed., New York, NY: McGraw-Hill Education, 2020.

[3] J. G. Enríquez y J. J. Gutiérrez, «Quality Assurance en la era de la Transformación Digital», Journal of Software Engineering Research and Development, vol. 12, n.º 4, pp. 45-60, 2024.

[4] International Software Testing Qualifications Board (ISTQB), Certified Tester AI Testing (CT-AI) Syllabus, Bruselas: ISTQB, 2023.

[5] S. Russell y P. Norvig, Artificial Intelligence: A Modern Approach, 4.<sup>a</sup> ed., Hoboken, NJ: Pearson, 2021.

[6] M. Fowler, Refactoring: Improving the Design of Existing Code, 2.<sup>a</sup> ed., Boston, MA: Addison-Wesley, 2019.