



DEPARTAMENTO DE
INGENIERIA COMERCIAL
UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Departamento de Ingeniería Comercial
MBA, Magíster en Gestión Empresarial

TRANSFORMACIÓN DEL PROCESO DE ASEGURAMIENTO DE CALIDAD DEL SOFTWARE MEDIANTE LA ADOPCIÓN DE INTELIGENCIA ARTIFICIAL GENERATIVA: UN ENFOQUE ESTRATÉGICO

Tesina de Grado presentada por
Gustavo Eduardo Inzunza Rojas

Como requisito para optar al grado de
MBA, Magíster en Gestión Empresarial

Guía de Tesina: Rodrigo Hermosilla

Octubre de 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: Transformación del proceso de aseguramiento de calidad del software mediante la adopción de inteligencia artificial generativa: Un enfoque estratégico

Nombre del candidato(a): Gustavo Eduardo Inzunza Rojas

Carrera / Grado: MBA, Magíster en Gestión Empresarial

Campus: Santiago Vitacura ; **Departamento:** Ingeniería Comercial

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Rodrigo Hermosilla Puebla, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 30/10/2025

; Firma:

Rodrigo Hermosilla P.

Estudiante o Candidato(a):

Fecha: 30/10/2025

; Firma:

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

TÍTULO DE TESINA: “TRANSFORMACIÓN DEL PROCESO DE ASEGURAMIENTO DE CALIDAD DEL SOFTWARE MEDIANTE LA ADOPCIÓN DE INTELIGENCIA ARTIFICIAL GENERATIVA: UN ENFOQUE ESTRATÉGICO”

AUTOR: GUSTAVO EDUARDO INZUNZA ROJAS

TRABAJO DE TESINA, presentando en cumplimiento parcial de los requisitos para el Grado de MBA, Magíster en Gestión Empresarial de la Universidad Técnica Federico Santa María.

OBSERVACIONES: _____

COMISIÓN DE TESINA:

Rodrigo Hermosilla, Profesor Guía

Macarena Gatica, Profesor Co-Referente Interno

Miguel Acevedo, Profesor Co-Referente Externo

Santiago, Octubre 2025

Todo el contenido, análisis,
conclusiones y opiniones vertidas
en este estudio son de mi exclusiva
responsabilidad.

Nombre: GUSTAVO EDUARDO INZUNZA ROJAS

Fecha: Octubre/2025

AGRADECIMIENTOS

Quiero agradecer a mi amada pareja Karina y mis hijos Josefina, Florencia y Facundo, mis principales fuentes de fortaleza e inspiración para seguir mejorando cada día y convertirme en una mejor persona. Sin ellos mi vida no tendría el mismo sentido y nada de esto habría sido posible.

Quiero agradecer también a mi profesor guía, Rodrigo Hermosilla, por guiarme en este difícil camino, revisar mi trabajo y ayudarme a cumplir el objetivo.

Por último, tal como dijo el rapero Snoop Dogg, quiero agradecerme por creer en mí, quiero agradecerme por hacer todo este arduo trabajo, quiero agradecerme por no tener días libres, y quiero agradecerme por nunca renunciar.

RESUMEN EJECUTIVO

Este estudio propone una innovación tecnológica en el aseguramiento de calidad de software mediante estrategias de prueba adaptativas basadas en GenAI, y se posiciona además como una herramienta estratégica para la toma de decisiones organizacionales. Sus hallazgos proporcionan una base sólida para decisiones de alto impacto relacionadas con la inversión tecnológica, la alineación de iniciativas de calidad con los objetivos del negocio y la aceleración de la transformación digital. Al mejorar la eficiencia, precisión y adaptabilidad de las pruebas, esta propuesta actúa como un habilitador clave para aumentar la competitividad, reducir riesgos operacionales y fortalecer la capacidad de respuesta frente a los desafíos del entorno digital.

En el marco del proceso de desarrollo de software, el aseguramiento de calidad (QA) es sin lugar a duda un eslabón clave para lograr la satisfacción del usuario final, sin embargo, suele ser un proceso frecuentemente infravalorado. A pesar de algunas iniciativas que buscan integrar el QA desde etapas tempranas del ciclo de vida del software, tales como: testing continuo, agilidad y QAOps, en la mayoría de las organizaciones sigue ejecutándose en fases tardías, con tiempos acotados y bajo una fuerte presión del negocio para acelerar la entrega de resultados, pudiéndose generar con esto importantes brechas de calidad y limitarse la capacidad de reacción ante los cambios que habitualmente surgen durante los proyectos informáticos.

Las tecnologías emergentes ofrecen una oportunidad única para modernizar la forma en que hacemos QA y reducir los altos costos asociados a las estrategias reactivas, facilitándonos herramientas que permiten adaptarnos de manera más eficiente a las necesidades particulares de las organizaciones. En esta línea, la Inteligencia Artificial Generativa emerge como un catalizador para transformar el proceso de pruebas, permitiendo la creación de estrategias de prueba adaptativas, con un enfoque dinámico que facilite la adaptación del proceso a medida que evolucionan los requerimientos, cambian las prioridades del negocio o surgen nuevos objetivos transversales, garantizando así una mayor alineación entre el proceso de pruebas y las necesidades del cliente.

Este trabajo busca explorar cómo la incorporación de GenAI en el aseguramiento de calidad de software puede optimizar los procesos de testing, promoviendo un enfoque adaptativo y orientado al valor. El objetivo fundamental es demostrar que la integración continua y dinámica soportada por tecnologías inteligentes, no solo mejora la eficiencia operativa, sino que también fortalece la capacidad de las organizaciones para entregar productos de software con altos estándares de calidad y alineados con los objetivos estratégicos del negocio.

GLOSARIO

Conceptos relacionados a Pruebas de Software:

- **CI/CD**: Continuous Integration / Continuous Deployment
- **DevOps**: Development + Operations
- **DPA**: Digital Process Automation
- **ISO**: International Organization for Standardization
- **iSQI**: International Software Quality Institute
- **ISTQB**: International Software Testing Qualifications Board
- **OWASP**: Open Worldwide Application Security Project
- **QA**: Quality Assurance
- **RPA**: Robotic Process Automation
- **RS**: Requirements Engineering
- **SDLC**: Software Development Life Cycle
- **SRS**: Software Requirements Specification
- **STLC**: Software Testing Life Cycle
- **V&V**: Verification and Validation

Conceptos relacionados a Inteligencia Artificial:

- **AI**: Artificial Intelligence
- **BERT**: Bidirectional Encoder Representations from Transformers
- **CNN**: Convolutional Neural Network
- **DL**: Deep Learning
- **GAN**: Generative Adversarial Network
- **GenAI**: Generative Artificial Intelligence
- **GRU**: Gated Recurrent Unit
- **GPT**: Generative Pre-trained Transformer
- **GPU**: Graphics Processing Unit
- **HPC**: High Performance Computing
- **LLM**: Large Language Model
- **LSTM**: Long Short-Term Memory
- **ML**: Machine Learning
- **NLP**: Natural Language Processing
- **RAG**: Retrieval-Augmented Generation
- **TPU**: Tensor Processing Unit
- **VAE**: Variational Autoencoder

Otros conceptos generales:

- **API:** Application Programming Interface
- **CAGR:** Compound Annual Growth Rate
- **NPS:** Net Promoter Score
- **OCDE:** Organization for Economic Co-operation and Development
- **PII:** Personally Identifiable Information
- **PMO:** Project Management Office
- **RGPD:** Reglamento General de Protección de Datos
- **ROI:** Return on Investment
- **SRS:** Software Requirements Specification
- **UI:** User Interface
- **UNESCO:** United Nations Educational, Scientific and Cultural Organization
- **XPaths:** XML Path Language

TABLA DE CONTENIDOS

RESUMEN EJECUTIVO	5
GLOSARIO	6
ÍNDICE DE FIGURAS	10
1. INDUSTRIA.....	11
2. DESCRIPCIÓN DEL PROBLEMA.....	17
3. ESTADO DEL ARTE	21
3.1 Antecedentes.....	21
3.2 Marco Teórico.....	27
3.2.1 Enfoque Ejecutivo.....	27
3.2.2 Inteligencia Artificial (IA).....	32
3.2.3 Agentes IA	34
3.2.4 Inteligencia Artificial Generativa (GenAI)	35
3.2.5 Large Language Models (LLM)	37
3.2.6 IA en el Contexto Legal	39
3.2.7 Inteligencia Artificial en el Testing	41
3.2.8 Inteligencia Artificial en el Testing Continuo	45
3.2.9 Inteligencia Artificial Generativa (GenAI) en el Testing.....	48
4. HIPÓTESIS	52
5. OBJETIVOS.....	53
5.1 Objetivo General	53
5.2 Objetivos Específicos	54
6. PLAN DE TRABAJO	55
6.1 Enfoque.....	55
6.2 Diseño Metodológico	56
6.3 Técnicas e Instrumentos	57
7. PROPUESTA.....	58
7.1 Evolución de la IA en las Pruebas de Software.....	59
7.2 AI-Driven Testing	62
7.3 Estrategia de Implementación de GenAI en el Testing.....	67
7.3.1 Evaluación de los Procesos de Pruebas Actuales	68
7.3.2 Herramientas e Infraestructura.....	69

7.3.3 Recopilación de Datos y Generación de Información	69
7.3.4 Transformación Cultural	70
7.3.5 Integración con el Flujo de Trabajo de Desarrollo	70
7.3.6 Monitoreo y Mejora Continua.....	71
7.4 Mejoras Percibidas (Métricas).....	72
7.5 Modelo Tecnológico Propuesto.....	79
7.5.1 Retrieval-Augmented Generation (RAG).....	79
7.5.2 Descripción de la Solución	83
7.6 Desafíos en el uso de GenAI	87
7.6.1 Desafíos Tecnológicos	88
7.6.2 Desafíos Éticos	92
7.6.3 Desafíos Humanos & Capacitación	96
7.6.4 Desafíos Ambientales.....	98
8. RESULTADOS OBTENIDOS	99
8.1 Mejoras en Etapas Clave	100
8.2 Resumen	106
9. CONCLUSIONES	107
10. TRABAJO FUTURO	109
10.1 Hiper-Automatización en Pruebas.....	111
10.2 Automatización de Pruebas con Autorreparación	113
10.3 DevOps Potenciado por IA.....	114
10.4 Predicción y Prevención de Defectos Impulsada por IA	115
11. REFERENCIAS BIBLIOGRÁFICAS	116

ÍNDICE DE FIGURAS

Ilustración 1: Ingreso promedio anual en la industria del Testing	12
Ilustración 2: Informe del mercado global de pruebas de software.....	13
Ilustración 3: CPSQ in 2022 in the US.....	15
Ilustración 4: Valores de los profesionales de pruebas	18
Ilustración 5: Software Testing Life Cycle (STLC)	19
Ilustración 6: Etapas de la Ingeniería de Requerimientos	23
Ilustración 7: Problemas asociados a requerimientos cambiantes	26
Ilustración 8: Distribución típica de los tiempos de desarrollo	28
Ilustración 9: Costos ocultos del testing tradicional	30
Ilustración 10: Estimación de los ahorros de costos con GenAI	31
Ilustración 11: Definiciones de Inteligencia Artificial	33
Ilustración 12: Aplicaciones de la Inteligencia Artificial Generativa	36
Ilustración 13: Los mejores LLM de 2025.....	39
Ilustración 14: Beneficios clave de la IA en pruebas de software	45
Ilustración 15: Evolución del testing	47
Ilustración 16: GenAI como apoyo al proceso de Testing	51
Ilustración 17: Metodología de investigación.....	58
Ilustración 18: Beneficios del uso de AI-Driven Testing.....	65
Ilustración 19: Pruebas Tradicionales vs AI-Driven Testing	67
Ilustración 20: Estrategia de Implementación de GenAI en QA.....	71
Ilustración 21: Porcentaje de mejora de la productividad con GenAI	74
Ilustración 22: Ganancias GenAI integrado en el ciclo de vida de las pruebas ...	75
Ilustración 23: Integración de IA Generativa en la Ingeniería de Requerimientos	76
Ilustración 24: Análisis comparativo de rendimiento.....	78
Ilustración 25: Retrieval Augmented Generation (RAG).....	82
Ilustración 26: Revisión de Requerimientos y Creación de Matrices de Pruebas	84
Ilustración 27: Ejecución de Pruebas	85
Ilustración 28: Detección y Análisis de Defectos	86
Ilustración 29: Desafíos asociados al uso de IA Generativa.....	88
Ilustración 30: Sesgo accidental e intencional en algoritmos GenAI	94
Ilustración 31: Desafíos de datos asociados a la implementación de GenAI	95
Ilustración 32: GPT Privado para Inspección de Requerimientos.....	100
Ilustración 33: Ejemplo de Inspección de Requerimiento mediante IA.....	101
Ilustración 34: Comparativo Inspección Manual vs Inspección con IA	102
Ilustración 35: Experimento de creación de Casos de Prueba mediante IA	103
Ilustración 36: Mejoras en Pruebas Automatizadas Impulsadas por GenAI	104
Ilustración 37: Tecnologías en un marco de hiper automatización	112
Ilustración 38: Uso de IA/ML en Procesos de Pruebas	114

1. INDUSTRIA

En la economía digital actual, donde el software impulsa silenciosamente la mayoría de los procesos de negocio, la calidad del producto digital se ha convertido en un factor competitivo determinante. Los defectos ya no solo representan pérdidas económicas: deterioran la confianza del cliente, frenan la innovación y pueden comprometer la continuidad operacional en sectores críticos como lo son las finanzas, la salud o el transporte. En este contexto, la industria del aseguramiento de calidad de software trasciende su papel tradicional de soporte técnico para consolidarse como un habilitador estratégico del crecimiento empresarial. Comprender la evolución, los desafíos y las oportunidades de esta industria permite a los tomadores de decisión optimizar sus inversiones tecnológicas, mitigar riesgos reputacionales y acelerar el time-to-market en un entorno donde el margen de error es cada vez más estrecho.

La industria de las pruebas de software, también conocida como Testing, QA o de forma menos recurrente como V&V (verification and validation). Esta industria se encuentra inmersa dentro del ciclo de vida del desarrollo de software (SDLC) y constituye una pieza crítica y fundamental ya que tiene como objetivo principal evaluar la calidad de un producto de software y reducir el riesgo de fallos en su funcionamiento.

Para tener un entendimiento más acabado de la industria de las pruebas de software, surge la necesidad de tener una visión clara y precisa respecto al concepto de calidad. Si bien es cierto la gran mayoría de las definiciones coinciden en dos puntos principales: la satisfacción del cliente y el cumplimiento de los requerimientos del producto, podemos utilizar como referencia formal la visión que nos entrega la norma ISO 25010 y que define un modelo de calidad para productos de software basado en ocho atributos esenciales que se debe cumplir (Cortés, 2024):

- Adecuación Funcional
- Eficiencia de Desempeño
- Compatibilidad
- Usabilidad
- Fiabilidad
- Seguridad
- Mantenibilidad
- Portabilidad

Según el International Software Testing Qualifications Board (ISTQB) (Cerquozzi et al., 2023) se definen las pruebas de software como el proceso consistente en todas las actividades del ciclo de vida software, tanto estáticas como dinámicas,

concernientes con la planificación, preparación y evaluación de productos software y los productos de trabajo relacionados para determinar que éstos satisfacen los requisitos especificados, para demostrar que se ajustan al propósito y para detectar defectos. En palabras más sencillas se puede decir que las pruebas de software son el proceso de evaluar y verificar que un producto o aplicación de software hace lo que se supone que debe hacer, que cumple con los requerimientos esperados y que está libre de cualquier tipo de defectos.

A nivel global, la industria de las pruebas de software ha experimentado un crecimiento constante, impulsado principalmente por la transformación digital, la adopción del desarrollo ágil (Agile) y el auge de los modelos CI/CD (integración y despliegue continuos). En el último reporte del estado de las pruebas (PractiTest, 2025), se revela un panorama dinámico y en constante evolución para los profesionales del área de pruebas de software, lo que se traduce en un crecimiento salarial que también está ligado a la experiencia en el rubro.

Región	< 1 año	1-2 años	3-5 años	6-10 años	10+ años
Africa	N/A	N/A	N/A	\$45,00	\$55,00
Asia	\$10,00	\$13,33	\$23,00	\$51,00	\$72,78
Eastern Europe	\$10,00	\$20,00	\$36,67	\$37,50	\$64,09
Central Asia	N/A	N/A	\$15,00	\$37,08	\$66,82
Latin America	\$10,00	N/A	\$16,67	\$97,50	\$72,50
North America	N/A	\$96,67	\$105,00	\$131,11	\$127,90
Oceania	N/A	N/A	N/A	\$45,00	\$105,00
Western Europe	\$20,00	\$60,00	\$60,46	\$61,25	\$99,48

**Valor aproximado convertido a dólares estadounidenses (monto bruto, incluyendo salario, bono y beneficios si los hubiera)*

Ilustración 1: Ingreso promedio anual en la industria del Testing

Fuente: (PractiTest, 2025)

El tamaño del mercado de las pruebas de software fue valorado en USD 51.8 mil millones en 2023 y se calcula que registrará un CAGR (tasa de crecimiento anual compuesta) de más del 9% entre 2024 y 2029, impulsado por el aumento de los lanzamientos de productos e innovaciones de las grandes empresas. A medida que la tecnología evoluciona rápidamente, las empresas están constantemente bajo presión para proporcionar soluciones de software superiores para satisfacer las necesidades dinámicas de sus usuarios. Esta creciente demanda de software libre de errores ha estimulado el mercado para herramientas y técnicas avanzadas de prueba.

El crecimiento proyectado de la industria se debe al aumento del uso de aplicaciones móviles, que implica actualizaciones continuas para una mejor experiencia de usuario, al aumento de la inversión de las organizaciones en desarrollo de software con foco en seguridad, el incremento de las transacciones en línea y las preocupaciones asociadas a las leyes de protección de datos. Entre las principales tendencias para el período de análisis se incluyen el aumento significativo de la adopción de la computación en la nube, que conlleva una reducción de la carga de trabajo en infraestructura, la adopción de la inteligencia artificial y a la necesidad de entrega acelerada de productos de software de alta calidad (Global Market Insights, 2025) (The Business Research Company, 2025).

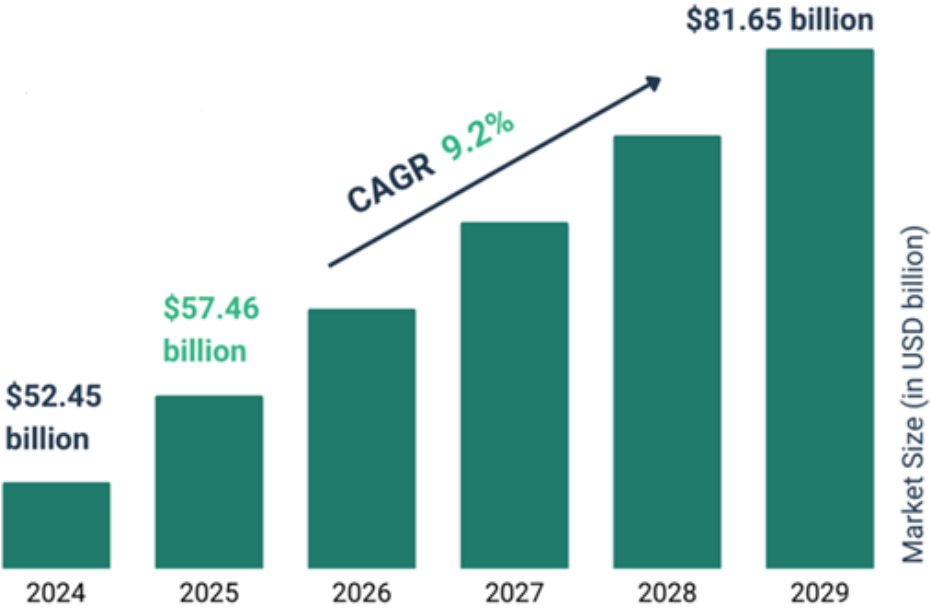


Ilustración 2: Informe del mercado global de pruebas de software

Fuente: (*The Business Research Company, 2025*)

Las organizaciones están invirtiendo en sofisticados marcos de pruebas y herramientas de automatización para simplificar el ciclo de vida del desarrollo de software. Además, la adopción de modelos DevOps han creado una necesidad de pruebas continuas para asegurar que el nuevo software sea probado correctamente antes del despliegue. Esta tendencia podría continuar a medida que las organizaciones traten de ofrecer una mayor ventaja competitiva proporcionando productos de software fiables y sólidos. En consecuencia, el mercado de pruebas de software está preparado para un crecimiento sostenido, impulsado por la demanda incesante de innovación y excelencia de las empresas líderes.

A pesar de la proyección optimista del crecimiento de la industria, resulta importante tener claridad respecto a cuáles son los dos principales desafíos que aparecen en el horizonte y que deben ser abordados de manera proactiva para mantener las buenas expectativas del mercado:

- **Alto costo de soluciones avanzadas.** Aunque las herramientas de automatización y soluciones de pruebas basadas en IA ofrecen múltiples ventajas, sus costos de implementación pueden ser elevados. Adoptar y mantener estas herramientas, sumado a la necesidad de recursos capacitados y expertos, aumentan los costos para las organizaciones. Por lo tanto, estas soluciones suelen ser disuasorias para empresas pequeñas y con presupuestos limitados.
- **Falta de profesionales expertos y capacitados.** La falta de profesionales capacitados, especialmente en áreas de prueba específicas de carácter más técnico, desafía en gran medida el crecimiento general de la industria de las pruebas de software. Además, la complejidad de la arquitectura de software (microservicios, sistemas distribuidos y soluciones en la nube) hace que los procesos de prueba sean más desafiantes.

A nivel global también se evidencia un creciente interés en eventos de promoción de las buenas prácticas y tendencias de la industria de las pruebas de software, además de un mayor énfasis profesional en la obtención de certificaciones asociadas a conocimientos específicos dentro área, destacándose de manera particular el catálogo de cursos que ofrecen iSQI y Brightest, certificaciones que incluso están convirtiéndose en requisitos excluyentes a la hora de participar en licitaciones asociadas a servicios de pruebas (funcionales y no funcionales).

Desde una perspectiva gerencial, la innovación en el ámbito del aseguramiento de calidad no solo responde a la necesidad técnica de mejorar procesos, sino también a una visión estratégica orientada al posicionamiento competitivo y la sostenibilidad del negocio. Los tomadores de decisión reconocen que la calidad del software tiene un impacto directo en la percepción del cliente, la fidelización y la reputación de la marca. Por lo mismo, invertir en nuevas formas de hacer testing permite reducir el riesgo reputacional y financiero asociado a fallos en producción, especialmente en industrias altamente reguladas o con exposición crítica al usuario final. En la misma sintonía, otra motivación clave radica en la necesidad de acelerar el *time to market* sin comprometer la calidad, lo que exige enfoques más ágiles, predictivos y adaptativos. Además, la integración de inteligencia artificial en los procesos de prueba se alinea con las agendas de transformación digital, permitiendo a las organizaciones no solo optimizar recursos, sino también generar capacidades analíticas que fortalezcan la toma de decisiones a nivel ejecutivo.

Resulta importante destacar el impacto económico asociado a un software que no incluya dentro de su presupuesto instancias formales de certificación versus un software que fue sometido a un riguroso proceso de pruebas. Lo primero que hay que tener claro es qué porcentaje del presupuesto de desarrollo de software debería destinarse a pruebas y en esta línea lo que indica la industria es que esto varía según el tipo de sistema (Jurkénas, 2025):

- Para sistemas críticos (sanidad, aeroespacial, finanzas): 40-50%
- Para aplicaciones empresariales: 25-35%
- Para aplicaciones de consumo: 15-25%
- Para aplicaciones móviles: 20-30%

En un estudio realizado por (Krasner, 2022) y plasmado en el reporte del CISQ (Consortium for Information & Software Quality), se estimaba que el costo de la mala calidad del software (cost of poor software quality: CPSQ) en EE. UU. ascendería al menos a USD 2,41 billones, mientras que la deuda técnica acumulada alcanzaría aproximadamente USD 1,52 billones. Estos resultados se muestran en la siguiente ilustración:

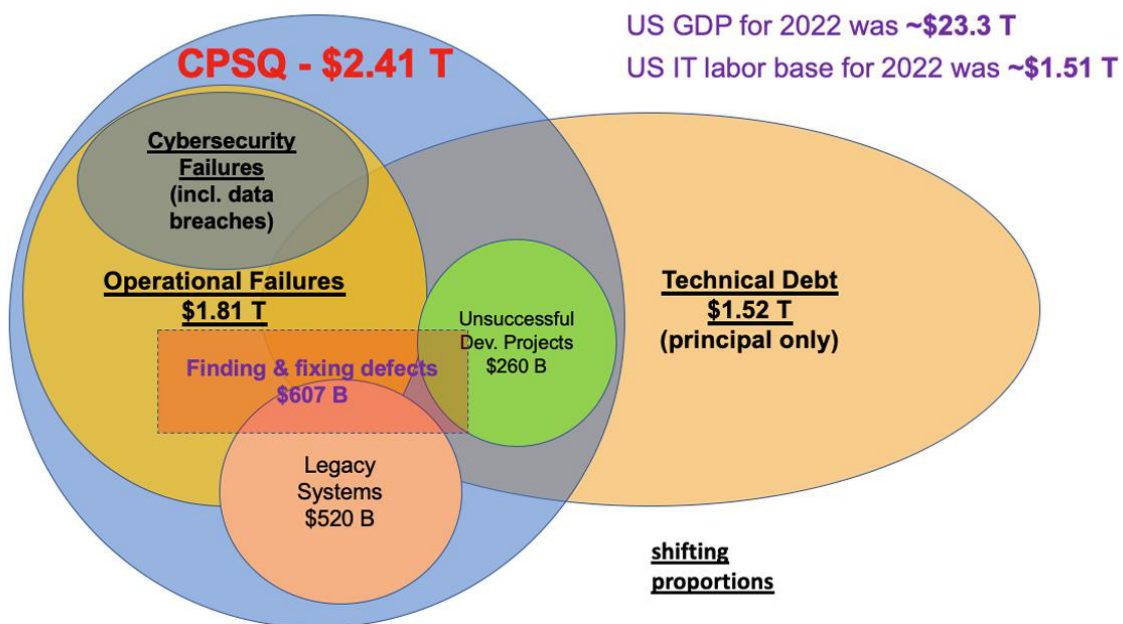


Ilustración 3: CPSQ in 2022 in the US

Fuente: (Krasner, 2022)

Profundizando en estos valores se identifican 3 grandes áreas problemáticas:

1. Las pérdidas por ciberdelincuencia debido a vulnerabilidades de software existentes se dispararon.

2. Los problemas en la cadena de suministro de software con componentes subyacentes de terceros (especialmente software de código abierto, también conocido como OSS) aumentaron significativamente.
3. El creciente impacto de la deuda técnica (TD) se ha convertido en el mayor obstáculo para realizar cambios en las bases de código existentes.

Un reciente informe de transformación de la calidad realizado por (Tricentis, 2025) evidencia que el 81% de las empresas encuestadas afirma que la mala calidad del software le cuesta a su negocio entre 500.000 y 5 millones de dólares al año y que casi el 40% estima pérdidas superiores al millón de dólares, debido a la rotación de personal, la deuda técnica y el aumento de los costes de mantenimiento.

En Estados Unidos casi la mitad (45%) de las empresas informan costes superiores a los 5 millones de dólares anuales asociados a desarrollos de baja calidad. De igual manera, en el Reino Unido, el 45% de las empresas estima que los costes de calidad de su software se sitúan entre 500.000 y 1 millón de dólares anuales, una carga considerable incluso en mercados consolidados. Cuanto mayor sea el negocio, mayor será el precio. Tres cuartas partes de las empresas globales con más de 1000 empleados podrían gastar hasta 5 millones de dólares al año en problemas de calidad del software. Las empresas de servicios financieros son las que más lo están sufriendo, con casi la mitad (49%) perdiendo más de 5 millones de dólares al año, lo que las convierte en el sector más afectado a nivel mundial.

Si bien es cierto los servicios financieros encabezan la lista, la carga es generalizada. Las empresas de energía y servicios públicos (42%), las organizaciones del sector público (38%), el comercio minorista (36%) y los fabricantes (34%) están sufriendo importantes consecuencias financieras debido a las deficiencias en la calidad del software. Además, el riesgo no es solo financiero, sino también operativo, ya que dos tercios (66%) de las organizaciones encuestadas afirman que es probable que sufran una interrupción o un problema grave el próximo año, y casi una cuarta parte (23%) considera que su organización está en grave riesgo.

2. DESCRIPCIÓN DEL PROBLEMA

En el entorno corporativo actual, los retrasos en proyectos tecnológicos, los sobrecostos y las fallas en producción no son simples incidentes operativos: representan amenazas directas a la rentabilidad, la reputación de la marca y la experiencia del cliente. Entre los factores más críticos, y a menudo subestimados, se encuentra el aseguramiento de calidad del software. Su verdadero valor no se limita al costo en horas o herramientas, sino al impacto que genera al prevenir pérdidas millonarias derivadas de fallos en ambiente productivo. Sin una estrategia de QA ágil, adaptativa y bien integrada desde las etapas iniciales del ciclo de desarrollo, las organizaciones se exponen a decisiones poco informadas, retrabajos costosos y la pérdida de oportunidades de mejora que podrían marcar la diferencia en su competitividad.

Es una realidad que la calidad cuesta caro: representa uno de los componentes más costosos dentro del ciclo de desarrollo de software, y por lo mismo, uno de los problemas a los que constantemente se ven enfrentados los tomadores de decisión es cuánto esfuerzo (tiempo y costo) están dispuestos a destinar a tareas de prueba y cómo hacer frente a las desviaciones propias de los proyectos de software que por naturaleza son altamente dinámicos.

Para entender la magnitud del problema podemos extraer algunos valores del análisis realizado por (Ríos, 2023) respecto al costo que tiene un Analista de Pruebas de perfil senior, cuya tarifa/hora se mueve en el rango de UF 0,6 y UF 0,89, y un Test Manager, cuya tarifa/hora se mueve en el rango de UF 0,84 y UF 1,5. Si pensamos en un equipo de pruebas estándar compuesto por un Test Manager y dos Analistas de Pruebas, el costo medio de trabajo por hora asciende a UF 2,42, evidenciando con esto que cualquier desviación que exista en la planificación del proyecto tendrá un impacto económico crítico para el éxito del proyecto, haciendo fundamental agilizar los procesos de prueba y buscar mecanismos que permitan manejar la incertidumbre de los proyectos.

Valores por perfil (UF/HR)			
Perfil	Precio Mínimo	Precio Medio	Precio Máximo
Tester	0,5	0,6	0,64
Analista QA Jr	0,54	0,62	0,68
Analista QA Sr	0,6	0,75	0,89
Ingeniero QA	0,78	0,85	1
Automatizador QA	0,78	0,85	1
Experto en performance	0,8	0,9	1,2
Test Manager	0,84	0,92	1,5
TMO	0,8	0,9	1,2
Jefe de Proyectos	0,8	0,9	1,2
Jefe de Proyectos Sr	0,84	0,92	1,5

Ilustración 4: Valores de los profesionales de pruebas

Fuente: (Ríos, 2023)

Las etapas del proceso de aseguramiento de calidad del software aunque pueden variar según el modelo de desarrollo utilizado (en cascada, ágil, híbrido) suelen seguir un ciclo de vida estructurado (Shah, 2024). En este contexto una de las instancias clave es la etapa de planificación (test planning), ya que es en este punto donde se definirá la estrategia integral de certificación que servirá como hoja de ruta para el proceso de pruebas, permitiendo que todos los miembros del equipo tengan una visión general de qué se probará y en qué condiciones particulares se llevará a cabo la certificación.

El artefacto que se desprende de esta instancia de planificación es el Plan de Pruebas, un documento estratégico que tiene como principal objetivo definir un marco de trabajo que permita garantizar la calidad del producto de software que será evaluado, y que debe contemplar al menos los siguientes elementos como parte de su estructura base (Cerquozzi et al., 2023) (Geeks for Geeks, 2025) (QAlified, 2024) (ISO, 2013):

- Objetivos
- Documentos de referencia
- Alcance de las pruebas
- Métodos y técnicas de prueba
- Estrategia de certificación
- Ambiente y datos de prueba
- Herramientas
- Roles y responsabilidades
- Métricas e indicadores

- Criterios de aceptación
- Criterios de inicio y finalización de pruebas
- Riesgos identificados
- Cronograma de actividades
- Entregables



Ilustración 5: Software Testing Life Cycle (STLC)

Fuente: (Shah, 2024)

El plan de pruebas establece la estrategia de pruebas que será utilizada de manera transversal a lo largo del proceso de certificación, sin embargo, en los procesos tradicionales de aseguramiento de calidad de software, los planes de prueba se suelen diseñar en fases tempranas del proyecto, basándose principalmente en la documentación disponible, tal como: especificación de requerimientos, historias de usuario, actas de reuniones con el cliente, y especificaciones técnicas del sistema.

Una de las problemáticas asociada a este estudio se presenta principalmente debido a la dualidad que se produce entre la naturaleza del plan de pruebas y el comportamiento de los proyectos de software de forma general. Por un lado tenemos un documento estático que se genera en una instancia preliminar y que debido a lo acotado de los tiempos de certificación no se vuelve a modificar a medida que avanzan las pruebas, y en contraposición nos encontramos con que los proyectos de software son por naturaleza dinámicos y en constante evolución, es decir, que lejos de ser procesos rígidos se comportan como organismos vivos que cambian continuamente en función de los ajustes de los requerimientos, retroalimentación de los usuarios, y decisiones que se toman durante el desarrollo.

La naturaleza dinámica de los proyectos de software demanda enfoques flexibles y adaptables, y de manera particular para el proceso de certificación se hace necesario que la estrategia de pruebas se vaya ajustando a los cambios del proyecto y que refleje la realidad de este. En la actualidad, a raíz del contexto cambiante de los proyectos, los planes de prueba tienen una limitación crítica: no están diseñados para evolucionar automáticamente frente a los cambios que naturalmente ocurren durante el ciclo de vida del desarrollo, generando con esto una desalineación entre los objetivos de prueba y el estado real del sistema. Esta evidente incongruencia se traduce en riesgos que comprometen la calidad del producto final, tales como:

- **Requerimientos no validados.** Cada vez que se actualice la documentación del proyecto no se realiza una nueva revisión/inspección de los requerimientos para validar su integridad y su adecuación con el contexto del proyecto.
- **Cobertura.** Los casos de prueba no cubrirán los nuevos requerimientos introducidos durante el desarrollo o los cambios a las funcionalidades ya existentes (y probadas).
- **Desperdicio de esfuerzo.** Se podría gastar tiempo en ejecución de pruebas que ya no son relevantes debido a un cambio en la priorización de los requerimientos, lo que afecta la eficiencia y eleva los costos del proceso.
- **Interpretación de defectos.** Al no existir trazabilidad con las nuevas definiciones los defectos pueden parecer más graves de lo que realmente son, o bien se pueden pasar por alto defectos que impactan directamente en los nuevos objetivos de negocio.
- **Confiabilidad de la certificación.** Los resultados del proceso de pruebas pierden validez y confiabilidad al estar basados en criterios de aceptación obsoletos, alcance desvirtuado y documentación desactualizada.

El problema base se presenta por una baja capacidad de adaptación de los artefactos de prueba, lo cual es recurrente en contextos de alta variabilidad y cambio constante como los que proponen las metodologías de desarrollo modernas. Así mismo, cada cambio introducido en el proyecto implica horas de trabajo adicionales del equipo a cargo de la certificación, lo que no solo se traduce en el no cumplimiento de la planificación establecida originalmente sino también en un costo elevado que generalmente no se encuentra considerado en el presupuesto inicial.

3. ESTADO DEL ARTE

3.1 Antecedentes

En un entorno donde la velocidad del cambio supera con creces la capacidad de reacción de los equipos técnicos y de las organizaciones, la gestión efectiva de los cambios y su impacto en el testing se convierte en un imperativo estratégico. Los líderes empresariales deben reconocer que cada modificación no gestionada adecuadamente, por mínima que parezca, puede derivar en desviaciones presupuestarias, incumplimientos en la entrega, pérdida de confianza de los stakeholders y en los casos más extremos incumplimiento normativo.

En el proceso de desarrollo de software, tanto en la metodología tradicional como en las ágiles, suelen presentarse dos premisas erradas que se complementan y que por su naturaleza afectan todas las etapas de manera transversal, y en particular a las pruebas de software: la existencia de un proceso robusto de levantamiento e inspección de requerimientos, y la percepción de que cada uno de estos requisitos se van a mantener inalterables a lo largo de todo el ciclo de vida de desarrollo.

El levantamiento de requerimientos es la primera etapa dentro del ciclo de desarrollo de software y es donde se establecen las bases sobre las cuales se llevará todo el proceso de desarrollo. La salida de esta etapa es un documento de especificación de requerimientos (funcionales y no funcionales) el cual idealmente debiese contener todos los requerimientos plenamente identificados y que describan de manera clara, concreta, atómica y libre de ambigüedades cada característica del sistema. Independientemente de la metodología de desarrollo de software utilizada, si estas propiedades no se logran a cabalidad probablemente las futuras etapas del desarrollo se verán impactadas negativamente.

A este levantamiento se le conoce formalmente como Ingeniería de Requerimientos, y corresponde al proceso de recopilar, analizar y verificar las necesidades del cliente o usuario para un sistema, teniendo como meta la entrega de una especificación de requerimientos de software correcta y completa (Arias, 2005). Para evidenciar la relevancia de ejecutar adecuadamente esta actividad, podemos referenciar a un estudio realizado por la consultora estadounidense Standish Group sobre 8.000 proyectos de software con el objetivo de determinar el grado de éxito de cada uno de ellos (Cristiá, 2024). Los resultados fueron bastante desalentadores: el 31% de los proyectos fueron cancelados antes de ser terminados, en las grandes compañías solo el 9% de los proyectos fueron entregados dentro del plazo y presupuesto predefinidos y el 16% en las pequeñas y medianas. A raíz de estos

descubrimientos se realizó otro estudio para determinar las causas de semejante fracaso, obteniéndose los siguientes resultados:

- ✓ Requerimientos incompletos (13,1%)
- ✓ Falta de compromiso del usuario (12,4%)
- ✓ Falta de recursos (10,6%)
- ✓ Expectativas no realistas (9,9%)
- ✓ Falta de soporte ejecutivo (9,3%)
- ✓ Requerimientos y especificaciones cambiantes (8,7%)
- ✓ Falta de planeamiento (8,1%)
- ✓ Fin de la necesidad del sistema (7,5%)

La ingeniería de requerimientos es un proceso estructurado, y se identifican como parte de esta actividad 4 etapas esenciales (Visure, 2025):

- **Elicitación.** Implica recopilar información de las partes interesadas a través de diferentes métodos, tales como: entrevistas, encuestas y talleres. Aquí se garantiza que no se pasen por alto necesidades críticas, lo que sienta las bases para unos requerimientos precisos y completos.
- **Análisis.** Se centra en refinar y priorizar los requisitos para garantizar que sean claros, coherentes y factibles. Esta fase aborda las ambigüedades y resuelve los conflictos, asegurando que los requisitos se alineen con los objetivos del proyecto.
- **Especificación.** Los requisitos se documentan en un formato claro y práctico, a menudo adhiriéndose a estándares como las especificaciones de requisitos de software (SRS).
- **Validación.** Implica revisiones, aprobaciones de las partes interesadas y pruebas para confirmar que los requisitos documentados cumplen los objetivos del proyecto.



Ilustración 6: Etapas de la Ingeniería de Requerimientos

Fuente: (elaboración propia)

Para asegurar que el documento de especificación de requerimientos que servirá de insumo de entrada para las posteriores etapas del ciclo de vida de desarrollo (diseño, desarrollo y pruebas) será lo más robusto posible y carente de defectos, inconsistencias, omisiones y/o ambigüedades, es que existe una tarea dentro del aseguramiento de calidad preventivo que tiene como objetivo principal realizar la validación formal de la documentación: la inspección de requerimientos. Sin embargo, en la gran mayoría de los proyectos de software esta tarea no se realiza o se hace artesanalmente, lo que se puede atribuir a factores relativos a tiempos acotados, costos y falta de una cultura de calidad. Esto sumado a instancias deficientes y aceleradas de levantamientos de requerimientos, se traduce en una documentación base que probablemente deba ser ajustada a lo largo del proceso de desarrollo.

Por otra parte, queda en evidencia que la naturaleza cambiante de los requerimientos de software desempeña un papel vital en el éxito o fracaso de cualquier proyecto. En situaciones reales, es difícil especificar todos los requerimientos del software, ya que tanto las necesidades como las circunstancias del escenario están sujetas a cambios. El sistema cambiará, independientemente de la precisión del proceso de especificación de requerimientos o de la profundidad de la comprensión del problema por parte del equipo. Los cambios pueden ocurrir

en cualquier punto del ciclo de desarrollo, desde la fase de levantamiento hasta la fase de mantenimiento, pasando por varias iteraciones. De hecho, es probable que más de la mitad de los requerimientos de un sistema cambien antes de su implementación (Muñoz et al., 2020) (William et al., 2006).

Dentro de los principales motivos que gatillan cambios de los requerimientos podemos mencionar (Jayatilleke et al., 2018) (Bigelow, 2019):

- ✓ Cambios en las necesidades del usuario.
- ✓ Cambios en el entorno del negocio.
- ✓ Descubrimiento de nuevas oportunidades técnicas.
- ✓ Retroalimentación continua en metodologías ágiles.
- ✓ Leyes y políticas gubernamentales
- ✓ Falta de comunicación

Adicionalmente, hay que tener en consideración que los cambios se manejan de manera diferente respecto al enfoque metodológico que se utilice. En una metodología en cascada, el equipo de desarrollo recopila un conjunto completo de requerimientos al inicio del proyecto, a menudo mediante un largo y detallado período de planificación. Los desarrolladores corrigen los requerimientos en las primeras etapas del ciclo de vida del proyecto, lo que dificulta la implementación de los cambios. Una queja sobre la metodología en cascada es que los requisitos fijos generan insatisfacción entre las partes interesadas; los desarrolladores no pueden adaptarse a los cambios. En contraposición, con metodologías de desarrollo continuo como ágil, el ciclo de vida del proyecto consiste en una serie de iteraciones relativamente cortas. El proyecto comienza con un conjunto de objetivos generales, pero el grupo siempre puede evaluar, reevaluar y modificar los requisitos según lo que se está construyendo y cómo debería funcionar. Es imposible e indeseable congelar los cambios en los requisitos del software en las primeras etapas del ciclo de vida del producto, como se hace en un enfoque en cascada. Los usuarios y las partes interesadas pueden aportar nuevas ideas en cualquier momento (Bigelow, 2019).

Un riesgo importante al implementar un cambio en un sistema de software en las últimas etapas del ciclo de vida del producto es la reducción de la calidad (William et al., 2006). Cada vez que se introduce un cambio en los sistemas de software en las últimas etapas del ciclo de vida, existe el riesgo inherente de que la calidad del sistema se degrade, lo que significa que será más difícil de mantener y de probar. Si llevamos esto al contexto particular de las pruebas de software, donde el principal artefacto de entrada al proceso es el documento de especificación de requerimientos, que sirve para establecer la planificación, el alcance del proyecto y la estrategia de certificación que será utilizada, es mucho más evidente el impacto negativo que esto puede tener, ya que si se aplican modificaciones a los requerimientos durante el desarrollo el plan de trabajo no necesariamente se va a

ajustar de manera dinámica, abriendo la posibilidad de no cubrir la totalidad de los nuevos requerimientos del sistema y/o probando funcionalidades que ya no existen o cuya priorización bajó.

En contraste con los tiempos asignados al resto de las actividades dentro del ciclo de vida de desarrollo de software, los tiempos asociados a las pruebas de software suelen ser bastante acotados e incluso se van acortando a medida que las etapas previas se van atrasando. Por lo mismo, cuesta trabajo concebir un proceso de certificación donde cada vez que se genere una nueva versión del documento de especificación de requerimientos, se disponga del tiempo para analizar los requerimientos, ajustar la estrategia de pruebas, la casuística y eventualmente ejecutar nuevos ciclos de prueba. Incluso, como parte de las metodologías de QA se establece de manera preliminar la restricción de que no puede modificarse el documento de entrada y en el caso de que eso pasara se traduciría en un control de cambio.

Como ya hemos visto en los párrafos previos, más allá de que uno esperaría que el documento de especificación de requerimientos no sufriese modificaciones, su naturaleza cambiante es una realidad en la gran mayoría de los productos de software y esto puede acarrear una serie de problemas graves de cara a las pruebas de software:

- **Desalineación entre pruebas y requerimientos.** Las pruebas generadas pueden volverse obsoletas rápidamente si los requerimientos cambian con frecuencia, lo que implica una alta tasa de mantenimiento para mantener la coherencia entre los casos de prueba y la especificación.
- **Dificultad en la trazabilidad.** Con cambios frecuentes, resulta complejo mantener una trazabilidad clara entre los requerimientos y los escenarios de prueba que se desprenden del mismo. Esto dificulta identificar qué pruebas son afectadas por cada cambio.
- **Pérdida de tiempo y recursos.** Reescribir o ajustar pruebas consume tiempo y esfuerzo, lo que se vuelve aún más crítico cuando un cambio implique repetir ciclos de pruebas.
- **Riesgo de cobertura.** Algunos cambios pueden pasar desapercibidos, lo que genera pruebas que no cubren todos los requerimientos actuales o en el caso contrario mantener pruebas que ya no son relevantes.
- **Inconsistencias en pruebas automatizadas.** Las pruebas automatizadas ligadas a especificaciones anteriores pueden fallar o no detectar defectos nuevos (falsos positivos y negativos).



Ilustración 7: Problemas asociados a requerimientos cambiantes

Fuente: (elaboración propia)

En base a todos estos antecedentes se hace evidente la necesidad de buscar mecanismos que permitan elaborar estrategias de pruebas dinámicas, que sean capaces de adaptarse a los cambios de los requerimientos y de las necesidades particulares de los clientes, reflejando en todo momento la realidad del proyecto y que esto se traduzca en un proceso de pruebas mucho más eficiente. Esta problemática cobra aún mayor relevancia en el contexto moderno de desarrollo de software donde los requerimientos y el entorno cambian continuamente, y en donde se hace vital que el área de QA no se convierta en un ralentizador que frene el impulso del modelo de desarrollo ágil.

3.2 Marco Teórico

Tradicionalmente las estrategias de prueba han estado marcadas por enfoques profundamente estáticos que, si bien es cierto han sido eficaces en contextos controlados, presentan limitaciones severas frente a entornos dinámicos, ciclos de desarrollo acelerados y requisitos de calidad en constante evolución. Es en este escenario, que la Inteligencia Artificial Generativa (GenAI) emerge como un aliado transformador, capaz de reconfigurar los paradigmas convencionales de prueba mediante mecanismos adaptativos y autónomos.

3.2.1 Enfoque Ejecutivo

Los tomadores de decisión deben entender que la adopción de nuevas tecnologías y frameworks no solo representa la promesa de una reducción de costos operativos, sino también de una transformación estratégica. Sin embargo, el gran desafío es justificar la inversión que conlleva adentrarse en la implementación de herramientas de inteligencia artificial en las pruebas de software.

En el contexto actual del desarrollo de software, una de las verdades incómodas que los líderes de proyectos y ejecutivos deben reconocer es que el aseguramiento de calidad de software no es una etapa marginal ni prescindible, sino un componente crítico que puede consumir entre un 15 % y un 25 % del esfuerzo total del proyecto, tal como se observa en la distribución típica de tiempos (*ver Ilustración N° 8*) y en modelos de desarrollo ágil los tiempos de QA se igual a los tiempos de implementación ya que se realizan en paralelo. Esta cifra puede sorprender a muchos gerentes acostumbrados a considerar el testing como un “checklist final” o una actividad de bajo impacto estratégico. Sin embargo, subestimar esta fase no solo representa un riesgo técnico, sino que tiene consecuencias directas en costos, plazos y reputación del producto.

Una implementación deficiente de QA conduce inevitablemente a ciclos de retrabajo, fallos en producción, pérdida de confianza del cliente y sobrecostos que pueden comprometer incluso la viabilidad del proyecto. En cambio, una inversión oportuna e inteligente en QA, especialmente si se adoptan enfoques modernos como pruebas adaptativas o estrategias basadas en inteligencia artificial generativa, permite detectar errores temprano, reducir el costo de corrección y mejorar el time-to-market

Fase	Porcentaje de Esfuerzo
Ingeniería de Requerimientos	10% - 15%
Diseño	15% - 25%
Codificación / Implementación	30% - 40%
Testing (QA)	15% - 25%
Despliegue	5% - 10%

Ilustración 8: Distribución típica de los tiempos de desarrollo

Fuente: (elaboración propia)

Lo primero que se debe hacer es analizar el coste de la IA en las pruebas de software, y particularmente las áreas de inversión clave que contribuyen al alto coste percibido de la IA en las pruebas de software:

- **Adquisición de tecnología.** Las pruebas de IA requieren herramientas y plataformas especializadas que puedan:
 - Generar y priorizar casos de prueba de forma autónoma.
 - Predecir defectos utilizando datos históricos.
 - Integrarse con pipelines de CI/CD para pruebas continuas.

Muchas de estas herramientas tienen un coste para la IA que incluye licencias, precios por uso o costes de infraestructura

- **Talento cualificado.** Las pruebas de IA requieren una nueva combinación de experiencia:
 - Ingenieros de control de calidad con conocimientos básicos de aprendizaje automático.
 - Científicos de datos para ayudar a construir o refinar modelos.
 - Arquitectos de pruebas capaces de diseñar flujos de prueba inteligentes.

Esto implica mejorar las competencias del personal existente (lo que requiere tiempo y recursos) o contratar talento especializado, que tiene una alta demanda y un precio elevado.

- **Integración e implementación.** La IA no funciona de inmediato, necesita:
 - Datos históricos limpios y estructurados para entrenar o ajustar algoritmos.
 - Trabajo de integración con sus sistemas de gestión de pruebas, rastreadores de errores y herramientas de control de versiones.
 - Fases piloto para validar la calidad de los resultados y evitar falsos positivos o brechas de cobertura.

Esta configuración requiere tiempo, coordinación e interrupción temporal de los flujos de trabajo actuales.

- **Mantenimiento y reentrenamiento de modelos.** A diferencia de los scripts de automatización tradicionales, los modelos de IA aprenden y se adaptan, lo que implica que también requieren:
 - Actualizaciones continuas de datos.
 - Reentrenamiento regular para reflejar la evolución del comportamiento del software.
 - Supervisión de desviaciones del modelo o degradación del rendimiento.

Estos esfuerzos garantizan que la IA siga aportando valor, pero incrementan la estructura de costos actual.

- **Gestión del cambio.** Muchos equipos subestiman el coste humano de la IA. La resistencia al cambio, el miedo a la pérdida de empleo o la falta de comprensión pueden frenar la adopción. Construir una cultura preparada para la IA requiere:
 - Patrocinio ejecutivo.
 - Comunicación clara.
 - Talleres y programas de formación continua.

Al evaluar el costo de la IA, muchas organizaciones se centran en los gastos visibles: licencias de herramientas, infraestructura en la nube o inversión en talento. Sin embargo, lo que a menudo pasa desapercibido son los costos ocultos que se generan al aferrarse a los enfoques de control de calidad tradicionales. Estos gastos ocultos pueden no aparecer en su hoja de ruta, pero erosionan constantemente la eficiencia de su equipo, la calidad del producto y los resultados. El enfoque ejecutivo debiese ser otro: si aún depende del control de calidad tradicional, manual o basado en scripts, aquí es donde ya está gastando su dinero, sin que se dé cuenta. Ver Ilustración 9. Costos ocultos del testing tradicional.

Las pruebas con IA no solo automatizan tareas, sino que también reducen estratégicamente estos costos ocultos mediante:

- Scripts autorreparables que reducen las actualizaciones manuales.
- Análisis predictivo para priorizar las pruebas y detectar defectos de alto riesgo de forma temprana.
- Informes en tiempo real para mejorar la visibilidad en todo el ciclo de vida del desarrollo de software (SDLC).

- Automatización más inteligente que escala la cobertura de las pruebas sin contratar más analistas de pruebas.

Costo Oculto	Cómo se Ve	Por Qué Duele
Mantenimiento Manual de Pruebas	Los equipos de QA pasan días o incluso semanas actualizando scripts de automatización frágiles después de cada cambio en el producto.	Pérdida de productividad, deuda técnica y ciclos de retroalimentación retrasados que impactan los tiempos de entrega.
Defectos No Detectados	Los errores llegan a producción debido a una cobertura de regresión limitada o falta de análisis predictivo de defectos.	Daño reputacional, pérdida de clientes, costosos <i>hotfixes</i> y aumento en los tickets de soporte.
Lanzamientos Retrasados	Los ciclos de prueba se alargan debido a casos de prueba redundantes y falta de priorización inteligente.	Pérdida de oportunidades de salida al mercado, retraso en la generación de ingresos e incremento de la tensión entre QA y Desarrollo.
Escalar con Más Personal	En lugar de escalar QA con herramientas inteligentes, las empresas aumentan el tamaño del equipo para cubrir la creciente demanda de pruebas.	Mayor costo de nómina, retrasos en la incorporación y rendimientos decrecientes por falta de aprovechamiento de la automatización.
Baja Visibilidad	Los reportes manuales y los datos fragmentados dificultan la identificación de cuellos de botella o brechas de cobertura.	Mala toma de decisiones, planificación reactiva y dificultad para demostrar el valor de QA ante la dirección.
Procesos de Prueba Aislados	Herramientas de prueba desconectadas, colaboración limitada y ausencia de información centralizada entre QA, Desarrollo y Operaciones.	Duplicación de trabajo, falta de trazabilidad e incapacidad de aplicar pruebas <i>shift-left</i> .

Ilustración 9: Costos ocultos del testing tradicional

Fuente: (QA Source Engineering Team, 2025)

Organizaciones de todo el mundo recurren cada vez más a la IA Generativa (GenAI), no solo para encontrar soluciones innovadoras y diferenciarse en el mercado, sino también como un potente motor para aumentar la productividad y el crecimiento de los ingresos. La implementación de GenAI en las pruebas de software ofrece un potencial de ahorro de costes significativo (Jayachandran, 2025). Podemos analizar las principales áreas en las que GenAI puede impulsar importantes eficiencias de costos en sus esfuerzos de pruebas de software:

- **Reducción de las pruebas manuales.** Las pruebas manuales siguen siendo una de las actividades más laboriosas y que más tiempo consumen en el ciclo de vida del desarrollo de software. Al aplicar GenAI estratégicamente, las organizaciones pueden reducir drásticamente las pruebas manuales automatizando tareas repetitivas y rutinarias.
- **Cobertura de pruebas mejorada y mayor calidad.** Utilizar GenAI para mejorar la cobertura de pruebas implica identificar y corregir más errores

durante la fase de prueba. Esto se traduce directamente en una menor cantidad de defectos que se filtran a los entornos de producción.

- **Tiempo de comercialización acelerado.** GenAI puede acelerar significativamente los ciclos de prueba al automatizar diversos aspectos, como la generación, ejecución y mantenimiento de casos de prueba. Unos ciclos de prueba más rápidos y eficientes permiten lanzamientos más rápidos, lo que permite a las organizaciones comercializar sus productos con mayor rapidez y asegurar una ventaja competitiva crucial.
- **Menores gastos de mantenimiento.** Las herramientas GenAI se pueden aprovechar al máximo para lograr capacidades de autorreparación, adaptando y modificando automáticamente los scripts de automatización a los cambios en la aplicación bajo prueba.
- **Utilización optimizada de recursos.** GenAI puede desempeñar un papel fundamental en los enfoques de pruebas basadas en riesgos al priorizar inteligentemente los casos de prueba según su riesgo e impacto. Esto garantiza un uso más eficaz de los recursos de prueba, reduciendo el desperdicio de esfuerzo y optimizando la asignación de recursos.
- **Reducción de Costos de Infraestructura.** Las herramientas de pruebas basadas en IA ofrecen la flexibilidad de escalar la infraestructura de pruebas según la demanda, optimizando así el uso de recursos y reduciendo los costos generales de infraestructura.

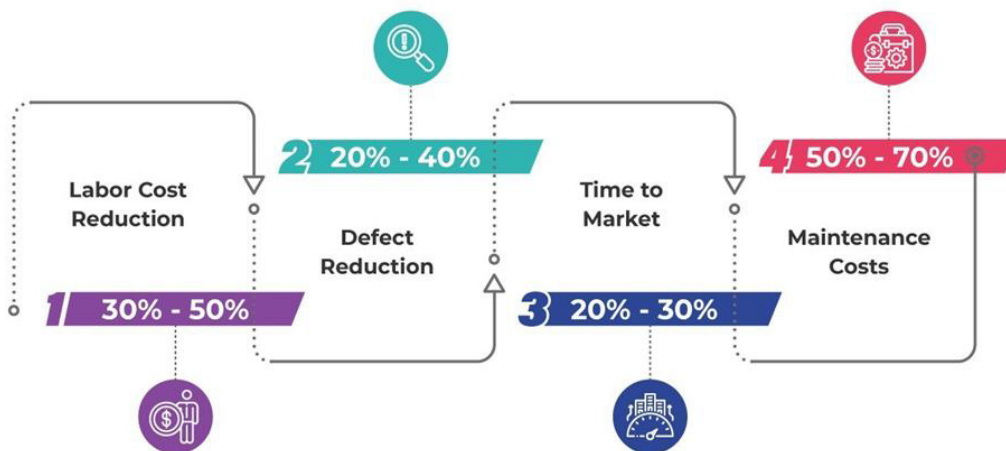


Ilustración 10: Estimación de los ahorros de costos con GenAI

Fuente: (Jayachandran, 2025)

3.2.2 Inteligencia Artificial (IA)

La Inteligencia Artificial constituye uno de los pilares tecnológicos más influyentes en la transformación digital contemporánea. Su creciente presencia en diversos sectores ha motivado un interés particular por comprender no solo su funcionamiento, sino también sus implicaciones, aplicaciones y regulaciones. En el contexto del aseguramiento de calidad de software, la IA ofrece oportunidades significativas para redefinir estrategias, optimizar procesos y mejorar la toma de decisiones.

Como primer paso para comprender con mayor claridad este concepto relativamente nuevo se hará referencia a la definición formal de Inteligencia Artificial que propone la Comisión Europea sobre la IA (Comisión Europea, 2019):

El término “inteligencia artificial” (IA) se aplica a los sistemas que manifiestan un comportamiento inteligente, pues son capaces de analizar su entorno y pasar a la acción — con cierto grado de autonomía — con el fin de alcanzar objetivos específicos. Los sistemas basados en la IA pueden consistir simplemente en un programa informático (p. ej. asistentes de voz, programas de análisis de imágenes, motores de búsqueda, sistemas de reconocimiento facial y de voz), pero la IA también puede estar incorporada en dispositivos de hardware (p. ej. robots avanzados, automóviles autónomos, drones o aplicaciones del internet de las cosas).

El término Inteligencia Artificial alude directamente al concepto de inteligencia, aunque este último, pese a haber sido ampliamente estudiado por disciplinas como la psicología, la biología y la neurociencia, continúa siendo difuso y difícil de definir con precisión. Ante esta ambigüedad, los investigadores en el campo de la IA suelen recurrir al concepto de racionalidad como un enfoque operativo más concreto. La racionalidad se entiende como la capacidad de seleccionar la mejor acción posible para alcanzar un objetivo específico, considerando ciertos criterios de optimización y los recursos disponibles. Si bien la racionalidad no agota por completo el significado de la inteligencia, constituye un componente esencial dentro de su marco conceptual, especialmente en el desarrollo de sistemas artificiales.

De acuerdo con Stuart Russell y Peter Norvig (Russell et al., 2020) un sistema de IA es, ante todo, racional, pero ¿cómo alcanza la racionalidad un sistema de IA? Como señala la definición formal del concepto de IA anteriormente expuesta, lo hace percibiendo el entorno en el que se encuentra inmerso el sistema a través de sensores, recopilando e interpretando datos, razonando sobre lo que percibe o procesando la información derivada de esos datos, decidiendo cuál es la mejor acción que puede realizar y actuando en consecuencia mediante accionadores, pudiendo así modificar el entorno. Los sistemas de IA pueden utilizar normas

simbólicas o aprender un modelo numérico; también pueden adaptar su conducta mediante el análisis del modo en que el entorno se ve afectado por sus acciones anteriores.

Sistemas que piensan como humanos	Sistemas que piensan racionalmente
"El nuevo y excitante esfuerzo de hacer que los computadores piensen... máquinas con mentes, en el más amplio sentido literal". (Haugeland, 1985)	"El estudio de las facultades mentales mediante el uso de modelos computacionales". (Charniak y McDermott, 1985)
"La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje...". (Bellman, 1978)	"El estudio de los cálculos que hacen posible percibir, razonar y actuar". (Winston, 1992)
Sistemas que actúan como humanos	Sistemas que actúan racionalmente
"El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren de inteligencia". (Kurzweil, 1990)	"La Inteligencia Computacional es el estudio del diseño de agentes inteligentes". (Poole et al., 1998)
"El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor". (Rich y Knight, 1991)	"IA... está relacionada con conductas inteligentes en artefactos". (Nilsson, 1998)

Ilustración 11: Definiciones de Inteligencia Artificial

Fuente: (Russell et al., 2020)

La Inteligencia Artificial no se trata de un campo perfectamente delimitado y la inclusión o no de ciertas tecnologías puede dar lugar a extensas discusiones. Dentro de la IA, entre muchas otras tecnologías, se pueden encontrar diversos componentes, siendo los principales (Barrere et al., 2025):

- **Aprendizaje Automático (machine learning).** Es una rama fundamental de la IA que permite a las máquinas aprender patrones y hacer predicciones a partir de datos. Los algoritmos de aprendizaje automático se entrenan utilizando conjuntos de datos y se perfeccionan a medida que obtienen más datos.
- **Redes Neuronales Artificiales.** Están inspiradas en la estructura y función del cerebro humano. Consisten en nodos interconectados que imitan a las neuronas y se utilizan en tareas como reconocimiento de patrones o el procesamiento de lenguaje natural. En particular, son el sustrato de los modelos de lenguaje de gran escala (LLM) que motorizan aplicaciones como ChatGPT.

- **Aprendizaje Profundo (deep learning).** Es un enfoque de aprendizaje automático que utiliza redes neuronales con múltiples capas para aprender representaciones de datos de forma jerárquica. Es muy efectivo para tareas complejas como reconocimiento de imágenes y voz.
- **Procesamiento de Lenguaje Natural.** Se enfoca en la interacción entre las computadoras y el lenguaje humano. Incluye tareas como traducción automática, análisis de sentimientos, resumen de texto y chatbots.
- **Visión por Computadora.** Se refiere a la capacidad de las máquinas para interpretar y comprender visualmente el mundo, como reconocer objetos en imágenes, detección de caras y seguimiento de objetos en vídeos.

3.2.3 Agentes IA

La Inteligencia Artificial ha transformado numerosos aspectos de nuestra vida diaria. Pero sin duda los agentes de IA son una de sus funciones más revolucionarias. Desde asistentes virtuales que responden preguntas en tiempo real hasta sistemas avanzados que toman decisiones críticas en entornos complejos, los agentes de IA están redefiniendo cómo interactuamos con la tecnología (Zunzunegui, 2025). De manera global este tipo de agentes están diseñados para realizar tres tipos de tareas:

- ✓ Interactuar con su entorno
- ✓ Recopilar datos de diferentes fuentes
- ✓ Tomar decisiones de manera autónoma

Los agentes de IA integran modelos grandes de lenguaje (LLM) con diversas tecnologías complementarias, lo que les permite ejecutar tareas complejas que anteriormente solo podían ser realizadas por seres humanos. Estos agentes interactúan activamente con su entorno para recopilar información, identificar los pasos necesarios para alcanzar un objetivo específico y actuar en representación de una persona o entidad. Además, son capaces de planificar estrategias, utilizar herramientas y fuentes de datos externas, tomar decisiones con distintos niveles de autonomía, e incluso colaborar con otros agentes de IA para resolver problemas de forma conjunta.

Los agentes de IA no constituyen una rama separada de la inteligencia artificial, sino una forma avanzada de integración operativa de sus capacidades. Mientras la IA proporciona los modelos y algoritmos necesarios para el razonamiento, el aprendizaje automático y la generación de lenguaje natural, los agentes de IA combinan estas funcionalidades dentro de un marco que les permite percibir su

entorno, planificar estrategias y ejecutar acciones de manera autónoma con el fin de alcanzar objetivos específicos. Esta integración les confiere un nivel de autonomía y adaptabilidad que resulta especialmente útil en contextos dinámicos y complejos, como el aseguramiento de calidad de software basado en pruebas inteligentes.

Los agentes de IA se perfilan como uno de los avances tecnológicos más relevantes de los últimos años en el campo de la inteligencia artificial. A diferencia de las tradicionales IA con las que simplemente interactuamos (como por ejemplo: un chatbot), un agente de IA es un programa capaz de percibir su entorno y tomar decisiones de forma autónoma empleando modelos de inteligencia artificial (Gil, 2025). En otras palabras, no espera instrucciones para cada paso, sino que actúa por sí mismo para lograr un objetivo, adaptándose a la información que recibe. Esta capacidad de operar independientemente ya sea explorando datos, navegando por internet o controlando dispositivos físicos, está transformando la inteligencia artificial, llevándola más allá de generar respuestas para convertirse en un agente activo que puede resolver problemas complejos en nuestro lugar.

3.2.4 Inteligencia Artificial Generativa (GenAI)

La Inteligencia Artificial Generativa (GenAI), es una de las vertientes más avanzadas y disruptivas dentro del campo de la inteligencia artificial. A través del análisis de sus características clave se busca establecer una base conceptual sólida que permita comprender su funcionamiento, su potencial aplicabilidad en diversos dominios, particularmente en el aseguramiento de calidad de software, y su relevancia en el desarrollo de estrategias de prueba adaptativas.

Una definición más formal apunta a una subdisciplina de la inteligencia artificial que se enfoca en la creación de contenido nuevo que resulta prácticamente indistinguible del generado por seres humanos. A diferencia de las técnicas tradicionales de IA, orientadas principalmente al reconocimiento de patrones y la realización de predicciones basadas en datos existentes, la IA generativa tiene la capacidad de producir datos originales a partir de los patrones previamente aprendidos (Dataalytics, 2024).

Uno de los subcampos más relevantes de la IA Generativa (GenAI) en la actualidad es el de los Large Language Models (LLM). Estos modelos consisten en conjuntos complejos de algoritmos diseñados para generar texto de forma predictiva, basándose en enormes volúmenes de datos. A diferencia de enfoques anteriores, los LLM destacan por el crecimiento exponencial de sus capacidades: cuando se entrenan con cantidades limitadas de información, sus respuestas son restringidas;

sin embargo, a medida que aumenta la escala de datos utilizados en su entrenamiento, su capacidad de generar texto coherente, relevante y contextualizado se incrementa de manera notable.

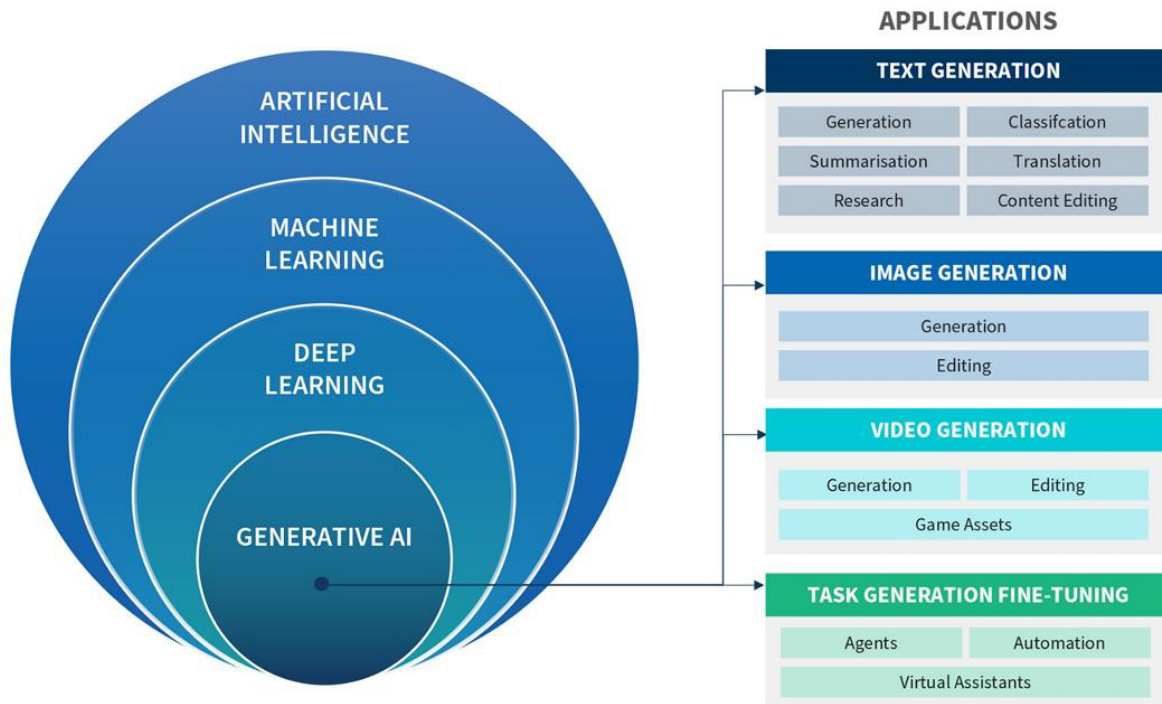


Ilustración 12: Aplicaciones de la Inteligencia Artificial Generativa

Fuente: (Cruz, 2024)

La inteligencia artificial generativa se apoya en modelos generativos y en los avances del aprendizaje profundo para crear una amplia variedad de contenidos de manera eficiente. En el marco del desarrollo continuo del Deep Learning, diversas arquitecturas de redes neuronales han sentado las bases para su evolución, destacando el papel fundamental del aprendizaje no supervisado. Las Redes Neuronales Convolucionales (CNN) han impulsado tecnologías clave como las Redes Generativas Antagónicas (GAN) y los Autoencoders Variacionales (VAE), esenciales para la generación y modificación avanzada de imágenes (García, 2024).

Paralelamente, la evolución de las Redes Neuronales Recurrentes (RNN), junto con sus variantes optimizadas, como las Redes de Memoria a Largo Plazo (LSTM) y las Unidades Recurrentes con Puertas (GRU), ha sido determinante en el avance del Procesamiento de Lenguaje Natural (NLP). No obstante, este panorama cambió drásticamente con la aparición de los Transformadores, una arquitectura innovadora que ha revolucionado el procesamiento del lenguaje natural gracias al desarrollo de Modelos de lenguaje de gran tamaño (LLM), representados por los

Transformadores Generativos Preentrenados (GPT) de OpenAI y las Representaciones Codificadoras Bidireccionales de Transformadores (BERT) de Google.

3.2.5 Large Language Models (LLM)

Los LLM son una categoría de IA generativa que corresponde a una combinación de algoritmos que generan texto predictivo a partir de una gran cantidad de datos. Estos modelos de lenguaje fueron entrenados con grandes volúmenes de información. La gran diferencia respecto a lo que ya conocíamos es el crecimiento exponencial de sus capacidades: cuando entrenamos un modelo con pocos datos, sus capacidades serán limitadas, en cambio cuando el volumen de datos crece, sus capacidades también lo hacen (Datalytics, 2024).

Los LLMs operan mediante arquitecturas avanzadas que capturan patrones lingüísticos complejos, permitiéndoles entender y generar lenguaje humano de manera efectiva. Los transformadores han eliminado las limitaciones de los modelos secuenciales y convolucionales, al permitir focalizarse en partes específicas del texto y determinar cuáles son más relevantes para mejorar la traducción y comprensión, haciéndolos particularmente eficaces en tareas de procesamiento de lenguaje natural a gran escala. Los LLM se entrenan en vastos conjuntos de datos de texto, aprendiendo de miles de millones de palabras para generar texto coherente y relevante. La capacidad de estos modelos para capturar y generar lenguaje se basa no solo en el tamaño de los datos de entrenamiento sino también en la complejidad de sus arquitecturas neuronales, que comprenden innumerables parámetros, permitiendo destacar desde la traducción automática hasta la generación de contenido, generando texto coherente y contextualmente relevante en una variedad de estilos y formatos (García, 2024).

El acceso a los LLM generalmente se habilita a través de terceros, que pueden incluir organizaciones que proporcionan herramientas especializadas desarrolladas o integradas con esta tecnología. Podemos clasificar los LLM en cuatro categorías según la forma en que los usuarios interactúan con ellos (European Medicines Agency, 2024):

1. **De terceros, de código abierto o cerrado, alojado externamente y disponible en línea.** Los usuarios interactúan con el modelo de lenguaje a través de una interfaz en línea. La interacción depende de la interfaz proporcionada por el tercero y puede limitar la personalización o la integración con los sistemas internos.
2. **De terceros, alojados externamente, parte de soluciones empresariales.** La interfaz puede ser más completa y estar diseñada para integrarse en

sistemas más grandes. Aquí, los usuarios interactúan con el LLM con una gama más amplia de herramientas, que potencialmente abarcan la gestión de datos, el análisis y funcionalidades adicionales.

3. **De terceros, de código abierto y alojado internamente.** Además de estar alojado internamente, es similar a la categoría anterior. Los usuarios tienen mayor control sobre su interacción con el LLM, lo que permite la personalización del entorno de alojamiento, la integración con otros sistemas y la posible modificación del modelo (si es de código abierto).
4. **(Re)capacitación interna.** Esta categoría ofrece un nivel único de interacción, totalmente definido dentro de la organización. Esto permite una amplia personalización, incluyendo interfaces a medida, integración con fuentes de datos internas para la recuperación, generación aumentada y optimización del rendimiento.

En este caso, se pueden integrar datos adicionales de la organización en LLM de terceros, o bien, se pueden refinar los LLM creados por entidades externas utilizando los datos de la organización. Además, existe la opción de desarrollar y capacitar LLM dentro de la organización, aprovechando sus propios datos y equipos, aunque esto requiere una inversión significativa.

La forma en que los usuarios interactúan con los LLM puede afectar la flexibilidad, el control, los requisitos de recursos y las posibilidades de integración durante la interacción. Por lo tanto, es fundamental considerar estos factores al seleccionar un tipo de modelo para una línea de trabajo específica.

Se proyecta que para 2030, el mercado global de Modelos de Lenguaje de Gran Tamaño (LLM) alcance los 36.100 millones de dólares, lo que pone de relieve cómo los LLM se han consolidado como un motor fundamental de la innovación empresarial. Ejemplos populares de modelos de lenguaje de gran tamaño, como GPT, LLaMA y Claude, ya ayudan a las empresas a optimizar sus operaciones, automatizar tareas y mejorar la experiencia del cliente (Mudryi, 2025).

A medida que los LLM multimodales continúan evolucionando y adquieren la capacidad de procesar texto, imágenes y videos, las empresas se beneficiarán de capacidades aún más avanzadas. El enfoque se desplazará hacia la mejora de la eficiencia mediante la reducción del consumo de energía y la minimización de sesgos, lo que mejorará aún más su valor en las operaciones comerciales. De cara al futuro, la adopción de LLM será crucial para las empresas que buscan mantenerse competitivas en los próximos años. Quienes integren LLM en sus procesos no solo mejorarán la eficiencia, sino que también abrirán nuevas vías de crecimiento e innovación.

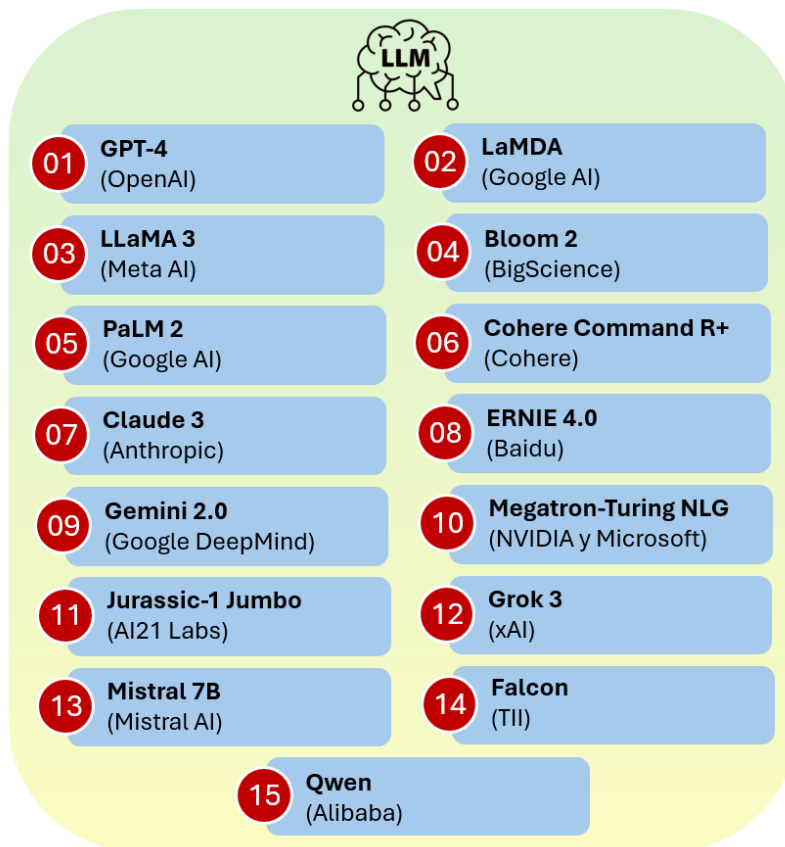


Ilustración 13: Los mejores LLM de 2025

Fuente: (Mudryi, 2025)

3.2.6 IA en el Contexto Legal

La adopción de la Inteligencia Artificial en América Latina está creciendo rápidamente, en especial en ámbitos como la salud, el sector financiero y la industria tecnológica. Este avance plantea diversas dudas y preocupaciones, en particular en lo relacionado con la privacidad y la protección de datos personales. Estas inquietudes no son exclusivas de la región, sino que también se evidencian en otras partes del mundo, como en el debate que se desarrolla actualmente en la Unión Europea. No obstante, este desarrollo acelerado representa un importante reto para los marcos normativos y de gobernanza en América Latina, que con frecuencia no logran mantenerse al ritmo de las transformaciones impulsadas por la IA (Fernández, 2024).

Varios países, como México, Brasil, Panamá, Perú y Chile, han iniciado la formulación de políticas y lineamientos orientados a regular el uso de la Inteligencia

Artificial. Estos esfuerzos se enfocan principalmente en la protección de datos y la ciberseguridad, con el objetivo de establecer un marco legal y normativo que oriente el desarrollo y la implementación responsable de esta tecnología (Veronese et al., 2021). La literatura acuerda que la IA tiene potencial para realizar grandes cambios en la sociedad y la economía del mundo. En una investigación publicada por el Instituto de Estudios Avanzados para la Inteligencia Artificial, un consorcio de entidades brasileñas, se afirma que el ascenso de la IA puede tener un impacto social y económico más significativo que la revolución de Internet y que las consideraciones éticas serán muy importantes (Cóbe et al., 2020).

Se proponen algunos de los principales desafíos que debe enfrentar cualquier aproximación regulatoria de estas nuevas tecnologías (Fernández, 2024):

- **Alineación con Innovaciones Tecnológicas.** La naturaleza dinámica de la IA requiere que los legisladores se mantengan continuamente informados sobre los desarrollos tecnológicos para diseñar leyes que sean suficientemente flexibles y prospectivas. Esto implica un compromiso no solo con la comprensión de la tecnología actual, sino también con la anticipación de futuras tendencias y aplicaciones.
- **Consideraciones Económicas en la Legislación.** Al formular políticas y regulaciones, es crucial considerar el amplio espectro de impactos económicos de la IA. Esto incluye evaluar cómo las innovaciones en IA pueden transformar el mercado laboral, influir en la competitividad entre empresas y sectores, y redefinir modelos de negocio enteros.
- **Promoción del Desarrollo y la Innovación.** Las regulaciones deben buscar un equilibrio entre proteger al público y los intereses empresariales, y fomentar un ambiente propicio para la innovación y el desarrollo económico. Un marco legal bien concebido puede actuar como un catalizador para la inversión en nuevas tecnologías, al mismo tiempo que asegura que sus beneficios se distribuyan de manera justa y ética.

Centrándonos en el escenario chileno, el 7 de mayo de 2024 el Presidente Gabriel Boric, a través del Ministerio de Ciencia junto con otros ministerios, presentó en el Congreso Nacional el proyecto de Ley que regula los sistemas de inteligencia artificial (IA) en Chile, que busca asegurar que el desarrollo y uso de estas tecnologías sea respetuoso de los derechos de las personas, fomentar la innovación y fortalecer la capacidad del Estado para actuar frente a sus riesgos y desafíos. Este proyecto propone reglas claras para quienes desarrollan, implementan o comercializan sistemas de IA, entregando certezas e impulsando el emprendimiento; y promoviendo un uso ético, transparente y responsable de la IA en beneficio de todas y todos. La propuesta recoge el trabajo de diversas instancias, tales como: el Comité de Expertos de IA, las comisiones del Congreso, mesas de

trabajo con la industria, academia y sociedad civil, y también se basa en principios de organismos como la Unesco y la OCDE.

A continuación se detallan los cuatro objetivos clave del Proyecto de Ley:

- ✓ Respetar los derechos fundamentales para garantizar la dignidad, la libertad y la igualdad de todas las personas, promoviendo la convivencia pacífica y el desarrollo social para la base de una sociedad justa y democrática.
- ✓ Promover la innovación, creando condiciones para que empresas, especialmente las pequeñas y medianas, puedan desarrollar nuevas soluciones en un entorno regulado.
- ✓ Entregar certezas legales a quienes usan IA, fomentando un desarrollo responsable, competitivo y alineado con estándares internacionales.
- ✓ Proteger a la ciudadanía frente a riesgos como: decisiones injustas, falta de transparencia en el uso de la tecnología y/o automatización que afecte derechos sin una supervisión adecuada.

3.2.7 Inteligencia Artificial en el Testing

Según los parámetros de referencia del sector (QASource Engineering Team, 2025), el retorno de la inversión (ROI) en IA para el testing incluye:

- Reducción de hasta un 60% en la duración de los ciclos de prueba.
- Reducción del 30% al 50% en los defectos posteriores al lanzamiento.

Justificar el coste de la IA en las pruebas de software se reduce al retorno de la inversión (ROI). Con una implementación adecuada, los retornos no son solo teóricos: son medibles, escalables y transformadores. Veamos cómo la IA ofrece un retorno de la inversión tangible a lo largo del ciclo de vida de las pruebas.

- Ciclos de prueba más rápidos = tiempo de comercialización más rápido
- Menos errores = menor costo de calidad
- Equipos más pequeños, mayor impacto
- Toma de decisiones más inteligente con análisis predictivo
- Mejora continua a escala

La mayoría de los equipos de testing ven el ROI de sus inversiones en IA en los primeros 6 a 12 meses. Las ganancias iniciales provienen de la reducción del tiempo de ejecución de las pruebas y una mejor predicción de defectos. Con el tiempo, a medida que los modelos aprenden y mejoran, el ROI se incrementa gracias a una mayor precisión y eficiencia. De cara a la alta dirección, la mejor forma de justificar el costo de implementar IA podría ser utilizar un caso de negocio centrado en resultados medibles, tales como: ahorro de tiempo en los ciclos de prueba, reducción de problemas de producción, lanzamientos acelerados y reducción de costes.

Los métodos de prueba tradicionales suelen ser lentos y propensos a errores humanos. Las pruebas manuales requieren un gran esfuerzo para crear y mantener casos de prueba, especialmente a medida que el software evoluciona. Aquí es donde la Inteligencia Artificial ofrece una clara ventaja: los sistemas de prueba basados en IA pueden generar, adaptar y ejecutar casos de prueba automáticamente sin necesidad de definir manualmente cada paso. Esto ahorra tiempo y recursos.

Mientras que los métodos de prueba convencionales se basan en scripts de prueba rígidos, la IA puede adaptarse dinámicamente a los nuevos requerimientos. Los modelos de aprendizaje automático aprenden continuamente de los resultados de las pruebas y adaptan los casos de prueba según la estrategia de prueba. Esto mejora la cobertura de las pruebas y aumenta la eficiencia. Los sistemas de IA son insuperables, especialmente para proyectos grandes que implican actualizaciones y cambios frecuentes. Puede concentrarse en un control de calidad de alto nivel mientras la IA se encarga del trabajo repetitivo. La mayor ventaja: la IA minimiza el error humano y garantiza que no se pase por alto ningún escenario importante (Trendig, 2024).

Los sistemas basados en IA pueden generar y adaptar automáticamente casos de prueba, lo que aumenta la cobertura de las pruebas y reduce el tiempo de salida a producción del software. Además, la capacidad de la IA para reconocer patrones y realizar predicciones permite la detección y corrección proactiva de errores antes de que ocurran en la práctica.

En los últimos años, el mundo digital ha sido testigo de un crecimiento exponencial en la frecuencia de lanzamientos de software. Este fenómeno ha intensificado la demanda de procesos de prueba que sean más ágiles, eficientes y confiables, y por lo mismo, la presión por entregar productos de software de alta calidad es más alta que nunca. Frente a este desafío, existe un consenso cada vez más amplio sobre

el valor de la automatización de pruebas, incluso cuando se implementa de manera parcial, como una solución efectiva para reducir la carga y acelerar los ciclos de desarrollo.

Tal como afirma (Artyukhovsky, 2025), la inteligencia artificial está transformando radicalmente las prácticas tradicionales del control de calidad. Lo que antes era un proceso lento, manual y repetitivo, se ha convertido en una actividad más ágil, inteligente y altamente eficiente. Para los equipos de aseguramiento de calidad, la IA no representa simplemente una herramienta adicional, sino un aliado estratégico capaz de enfrentar directamente los desafíos del desarrollo de software en entornos cada vez más dinámicos y exigentes. En este contexto, en una primera revisión se identifican claramente 7 beneficios fundamentales que propone la implementación de la IA como parte de un marco de trabajo en QA:

- **Automatización de Tareas Repetitivas.** Considere la cantidad de tiempo que los equipos dedican a redactar casos de prueba y a detectar defectos de manera manual. Estas actividades, además de ser tediosas, consumen recursos valiosos que podrían destinarse a tareas de mayor impacto. En este escenario, la IA aplicada a la automatización del control de calidad desempeña un papel fundamental, asumiendo las tareas repetitivas de forma eficiente y sin intervención humana constante. De este modo, libera a los equipos de QA para que puedan concentrarse en resolver problemas complejos, tomar decisiones estratégicas y elevar la calidad general del producto.
- **Información Predictiva.** ¿Qué pasaría si pudiera identificar las debilidades de su código antes de que se conviertan en problemas reales? La IA aplicada a las pruebas de software hace precisamente eso. A través del análisis de datos históricos y patrones de comportamiento, la IA es capaz de predecir las áreas de mayor riesgo dentro del código. Esto permite a los equipos de control de calidad actuar proactivamente, abordando posibles fallos antes de que se manifiesten, y reduciendo así la necesidad de costosas correcciones en etapas posteriores del desarrollo.
- **Mejora de la Cobertura de las Pruebas.** Las pruebas de software tradicionales a menudo presentan vacíos, especialmente al abordar casos extremos o validar el comportamiento en entornos variados. La IA redefine este panorama al detectar de forma proactiva escenarios ocultos y ejecutar pruebas bajo una amplia gama de condiciones. Según datos de TestRail, más del 50 % de los profesionales de aseguramiento de calidad reportan mejoras significativas tanto en la cobertura de pruebas como en la productividad gracias al uso de IA.
- **Apoyo a la Implantación Continua.** El principal objetivo de cualquier equipo DevOps es implementar actualizaciones con rapidez sin comprometer la estabilidad del sistema. La IA se integra de forma natural en los flujos de

trabajo CI/CD, proporcionando análisis en tiempo real durante las fases de despliegue. Al identificar de inmediato posibles incidencias, permite aplicar correcciones de forma instantánea, lo que no solo acelera los ciclos de lanzamiento, sino que también refuerza la confianza en la calidad y fiabilidad del software entregado.

- **Mejora de la Eficacia.** En el ámbito del aseguramiento de calidad, la velocidad y la precisión suelen percibirse como objetivos en conflicto. Sin embargo, la IA logra cerrar esa brecha de manera efectiva. Al automatizar y optimizar los procesos de verificación, la IA acelera significativamente el ciclo de pruebas sin comprometer la exactitud de los resultados. Esto permite a los equipos cumplir con plazos ajustados mientras mantienen altos estándares de calidad.
- **Mayor Precisión.** Las pruebas manuales, por muy necesarias que sean, están sujetas a errores humanos. El cansancio, la falta de atención o simples omisiones pueden hacer que ciertos defectos pasen desapercibidos. La IA aplicada al control de calidad reduce significativamente este riesgo al aportar precisión, consistencia y exhaustividad al proceso de prueba. Gracias a su capacidad para identificar patrones y anomalías de forma sistemática, la IA detecta fallos que podrían escapar al ojo humano, contribuyendo así a la entrega de un software más limpio, robusto y confiable.
- **Mantenimiento de pruebas Dinámico.** A medida que el software evoluciona constantemente, los procesos de prueba deben adaptarse en igual medida. Sin embargo, actualizar manualmente los casos de prueba puede resultar una tarea tediosa y una considerable pérdida de tiempo. La IA aborda este desafío de forma eficiente, actualizando automáticamente los casos de prueba en función de los cambios en la aplicación. Esto no solo simplifica el mantenimiento, sino que también permite a los equipos centrarse en nuevos desafíos y en la mejora continua del producto, en lugar de invertir recursos en ajustes repetitivos del pasado.

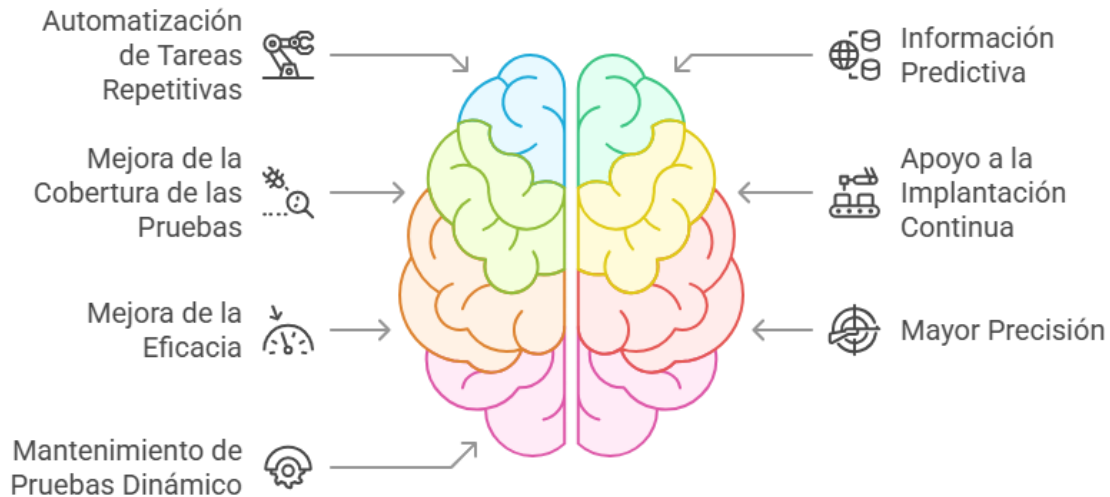


Ilustración 14: Beneficios clave de la IA en pruebas de software

Fuente: (elaboración propia)

3.2.8 Inteligencia Artificial en el Testing Continuo

La integración del QA en DevOps da lugar a pruebas continuas que garantizan la rápida integración de los cambios de código de forma eficaz y eficiente. Además, las pruebas continuas democratizan la calidad al iterar el QA a lo largo del ciclo de vida del desarrollo de software, impulsando la calidad a gran velocidad. Sin embargo, las pruebas continuas suelen presentar cuellos de botella, como la automatización aislada, la falta de visibilidad integral de los requerimientos, un gran volumen de pruebas, etc. Una manera eficiente de abordar estos desafíos apunta justamente a la implementación de un enfoque basado en inteligencia artificial que se apoye en el aprendizaje automático (ML) para orquestar la calidad en todo el proceso de pruebas continuas, lo que permite un QA autónomo y sin intervención.

Tal como propone (Cognizant, 2020) la IA puede optimizar el proceso de pruebas continuas al eliminar la intervención manual. Con la IA, los equipos de control de calidad pueden activar ciclos de pruebas desatendidos, donde se identifican defectos y se aplican medidas correctivas en tiempo de ejecución, basándose en la información obtenida de conjuntos de datos históricos y eventos pasados. De esta forma, el motor de IA garantizará que solo el código robusto avance de una etapa a la siguiente, orquestando la calidad a lo largo del ciclo de vida del desarrollo de software.

Si bien es cierto la gran mayoría de las actividades de control de calidad se automatizan en pruebas continuas, el código aún debe aprobarse manualmente de una compuerta de calidad a otra, según los resultados de las pruebas. Esta automatización aislada, o lagunas de automatización, da lugar al efecto acordeón, o a un flujo interrumpido de elementos debido al estancamiento en el pipeline.

Un motor de IA asume la tarea de verificar el código en los controles de calidad, lo que hace que este proceso sea autónomo. Al analizar los resultados de estas pruebas automatizadas, un algoritmo de aprendizaje automático puede aprobar o rechazar la progresión del código, creando un flujo de trabajo totalmente automatizado. Al orquestar los procesos de control de calidad con IA, los equipos de control de calidad pueden:

- **Automatizar los controles de calidad.** A medida que los algoritmos de aprendizaje automático determinan si el código es “correcto” o “incorrecto” según datos históricos, los equipos de control de calidad pueden confiar al motor de IA la promoción del código o la desactivación de funciones con alta probabilidad de causar interrupciones en la aplicación o defectos de producción.
- **Predecir las causas raíz.** La clasificación o identificación de la causa raíz de un defecto, es una de las razones de los retrasos en el lanzamiento de nuevas funciones. Mediante patrones y correlaciones, los algoritmos de aprendizaje automático pueden rastrear los defectos hasta sus causas raíz, y la IA puede activar pruebas correctivas antes de que el código avance. A medida que la IA toma estas decisiones, el margen de error se reduce significativamente.
- **Aproveche la monitorización precognitiva.** Los algoritmos de aprendizaje automático (ML) detectan síntomas en errores de codificación que previamente se habían pasado por alto. El algoritmo puede entonces señalar estos síntomas, como un alto uso de memoria, como una amenaza potencial que podría provocar una interrupción del servicio. Como medida correctiva, el motor de IA puede iniciar automáticamente un proceso paralelo para optimizar el consumo de recursos del servidor.

En la actualidad los enfoques tradicionales a menudo tienen dificultades para seguir el ritmo de los procesos Agile y DevOps, lo que genera deficiencias en el control de calidad. Es precisamente en este ámbito que surge la IA como una fuerza transformadora que permite el análisis y la ejecución de pruebas en tiempo real. Al aprovechar algoritmos de aprendizaje automático y análisis predictivo, la IA permite a los equipos de control de calidad detectar problemas de forma temprana, adaptarse dinámicamente a los cambios y mantener la calidad del software sin ralentizar el desarrollo. Mediante la automatización inteligente y la información

práctica, la IA hace que las pruebas continuas no solo sean posibles, sino también eficientes y fiables.

Según (Navaneethakrishnan, 2025) la clave del testing continuo impulsado por IA reside en su capacidad para analizar grandes cantidades de datos en tiempo real. Las herramientas de IA monitorizan el rendimiento de las aplicaciones, los resultados de las pruebas y el comportamiento del usuario para identificar patrones y anomalías que podrían indicar posibles problemas. Este enfoque proactivo garantiza la detección de defectos antes de que se conviertan en problemas críticos. Además, los algoritmos de IA pueden priorizar las pruebas en función del riesgo, centrándose en las áreas con mayor probabilidad de fallo. Estas pruebas dirigidas reducen la necesidad de una cobertura exhaustiva de las pruebas, a la vez que mantienen la confianza en la estabilidad del software. Por ejemplo, la IA puede predecir qué partes de un sistema son vulnerables basándose en datos históricos de defectos, guiando a los evaluadores hacia las áreas donde sus esfuerzos son más efectivos. Esta precisión no solo ahorra tiempo, sino que también reduce los costes asociados a las correcciones de posproducción.

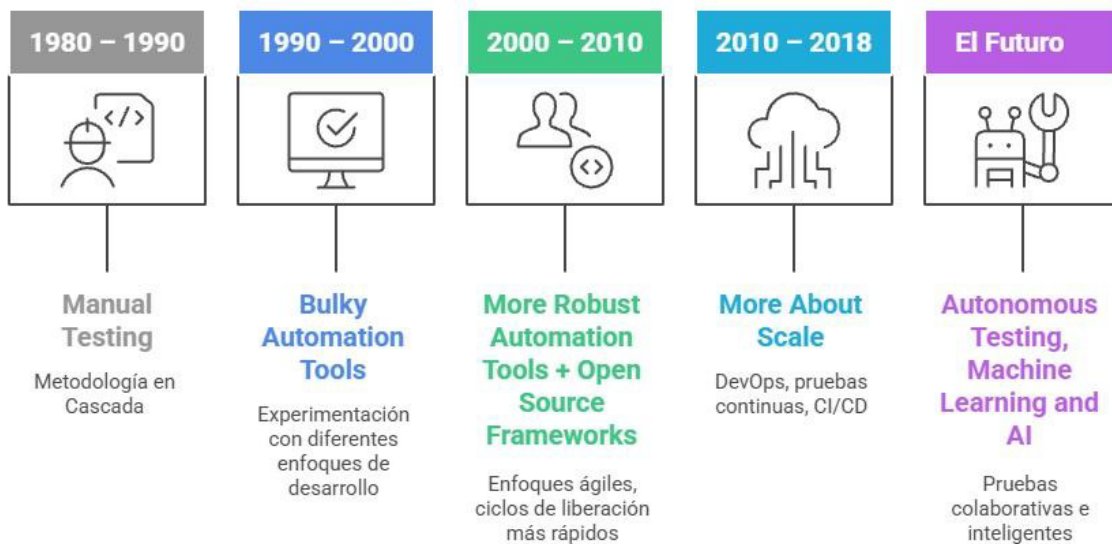


Ilustración 15: Evolución del testing

Fuente: (CloudQA, 2025)

Otra ventaja significativa de la IA en el Testing Continuo es su capacidad de adaptarse a los rápidos cambios típicos del desarrollo moderno. Las aplicaciones evolucionan rápidamente, con actualizaciones frecuentes y la incorporación de nuevas funciones. Las herramientas basadas en IA ajustan dinámicamente los scripts y las configuraciones de prueba para adaptarse a estos cambios, eliminando

el esfuerzo manual necesario para el mantenimiento de los casos de prueba. Esta capacidad de autorreparación garantiza que las pruebas sigan el ritmo del desarrollo, reduciendo retrasos y cuellos de botella. Además, la IA mejora la colaboración entre equipos al proporcionar paneles de control en tiempo real que muestran métricas de prueba, tendencias de defectos e información sobre el rendimiento. Estas vistas centralizadas fomentan la transparencia y permiten a las partes interesadas tomar decisiones informadas, impulsando una cultura de calidad en toda la organización.

3.2.9 Inteligencia Artificial Generativa (GenAI) en el Testing

A diferencia de los sistemas de IA tradicionales, que se basan en reglas explícitas e instrucciones programadas por humanos, la IA generativa emplea técnicas de aprendizaje automático para desarrollar resultados innovadores y únicos basados en patrones y datos con los que ha sido entrenada. Los modelos generados por la IA generativa son capaces de identificar patrones, detectar anomalías y anticipar posibles problemas que pueden afectar la calidad del software. Esta técnica proactiva permite la detección temprana de problemas, lo que ayuda a los desarrolladores y a los equipos de control de calidad a tomar medidas preventivas y a mejorar el estándar general del programa. Además, la IA generativa en las pruebas de software puede automatizar el desarrollo de casos de prueba y facilitar la creación de datos de prueba sintéticos (ITConvergence, 2024).

En esencia, la IA Generativa se basa en modelos de lenguaje de gran escala (LLM) capaces de generar resultados novedosos y útiles, como casos y datos de prueba, sin requerir instrucciones humanas detalladas. Esta capacidad de generación autónoma representa un cambio radical en el ámbito del aseguramiento de calidad, al permitir la creación de pruebas contextualizadas de manera dinámica, reduciendo significativamente la intervención manual y ampliando el alcance y la eficiencia del proceso de pruebas.

Aunque la IA Generativa puede parecer intimidante por la complejidad de los modelos que la sustentan, comprender sus principios fundamentales permite apreciar su enorme potencial en el ámbito del aseguramiento de calidad. Su verdadero valor radica en la capacidad de crear, adaptar y generar pruebas alineadas con las necesidades específicas de un sistema o funcionalidad. Desde la generación automática de casos de prueba a partir de descripciones funcionales hasta la generación de estrategias dinámicas y adaptativas, sus aplicaciones en control de calidad son amplias, versátiles y continúan evolucionando rápidamente. A medida que la IA generativa continúa evolucionando y madurando, promete revolucionar aún más nuestro enfoque de las pruebas, impulsando un futuro donde

las pruebas sean cada vez más completas, autónomas y eficientes (TestRigor, 2024).

La IA generativa aporta numerosos beneficios al aseguramiento de calidad del software. Sus capacidades únicas y técnicas avanzadas abren nuevas posibilidades para mejorar la cobertura de las pruebas, optimizar la detección de defectos y acelerar el desarrollo de software. En esta sección vamos a volver a profundizar en los principales beneficios que ofrece GenAI a la industria de las pruebas (Katalon, 2024) (Andrades, 2025) (Vinayagam, 2024) (Hall, 2024) (Cser, 2024):

- **Mejora en la Cobertura y Eficiencia.** La principal ventaja de la IA Generativa en el control de calidad del software es su capacidad para mejorar la cobertura de las pruebas. Al aprovechar algoritmos y grandes conjuntos de datos, los modelos de IA Generativa pueden generar automáticamente casos de prueba completos que abarcan diversos escenarios y entradas. Esta generación automatizada de casos de prueba reduce el esfuerzo requerido, a la vez que aumenta la minuciosidad y la eficacia del proceso de prueba.
- **Mejora en la Detección de Defectos.** La IA generativa descubre fácilmente problemas complejos de software que pueden ser difíciles de identificar para los analistas humanos. Estas técnicas analizan grandes volúmenes de datos relacionados con el software, como código y registros, para identificar patrones y diferencias con el comportamiento esperado de la aplicación. Al reconocer estas irregularidades, los modelos de IA generativa pueden detectar posibles errores, vulnerabilidades y cuellos de botella de rendimiento en las primeras etapas del proceso de desarrollo.
- **Predicción y Resolución Inteligente de Defectos.** Al incorporar pruebas basadas en IA Generativa, los analistas de calidad pueden anticiparse a los defectos antes de que se manifiesten. Estos modelos analizan datos históricos, identifican patrones recurrentes y detectan posibles fallos desde las primeras etapas del ciclo de vida del desarrollo de software (SDLC). Este enfoque preventivo no solo incrementa la fiabilidad del producto final, sino que también reduce significativamente los costos asociados a la corrección de defectos posteriores al lanzamiento.
- **Incorporación Fluida con Pipelines de CI/CD.** Con la adopción generalizada de metodologías Agile y DevOps, las pruebas continuas se han vuelto fundamentales. La aplicación de IA Generativa en entornos de CI/CD permite integrar herramientas inteligentes con plataformas como GitHub Actions, Jenkins o GitLab CI, automatizando la generación y ejecución de pruebas en tiempo real. Esto facilita la validación continua del código, acelerando los ciclos de entrega y mejorando la confiabilidad de los lanzamientos de software.

- **Autoaprendizaje y Pruebas Adaptativas.** A diferencia de la automatización tradicional, las pruebas impulsadas por IA Generativa tienen la capacidad de aprender de ejecuciones anteriores y adaptarse de forma dinámica a los cambios del alcance. Esto reduce significativamente la necesidad de mantenimiento manual de scripts, ya que los marcos de prueba basados en IA pueden autorrecuperarse y ajustarse automáticamente. El resultado es una implementación de pruebas más robusta, resiliente y eficiente.
- **Reducción de Tiempos y Costos.** Una de las ventajas significativas de la IA generativa en la automatización de pruebas es su capacidad para reducir el tiempo y los costos de las pruebas. Los métodos de prueba tradicionales suelen implicar tareas repetitivas y que consumen mucho tiempo. La IA generativa automatiza estas tareas, liberando a los equipos de control de calidad para que se centren en las fases más críticas del ciclo de vida del desarrollo de software. Esto acelera el proceso de pruebas, garantizando una mayor calidad en las versiones de software.
- **Enfoque Holístico para el Mantenimiento de las Pruebas.** La IA Generativa adopta un enfoque holístico al integrar de manera fluida múltiples fuentes de datos en el proceso de mantenimiento de pruebas. Es capaz de analizar tanto los aspectos visuales como el código HTML subyacente, además de interpretar el contenido legible para los usuarios. Esta visión multidimensional permite mantener los casos de prueba actualizados, relevantes y alineados con los cambios continuos en el desarrollo del software. En la práctica, actúa como un observador inteligente y constante que supervisa, ajusta y optimiza el proceso de pruebas en tiempo real.
- **Análisis de Resultados.** Durante la ejecución de las pruebas, la AI Generativa recopila y analiza registros, métricas de rendimiento y mensajes de error, evaluando el comportamiento de la aplicación bajo distintas condiciones. Gracias a algoritmos de aprendizaje automático, es capaz de detectar anomalías, identificar cuellos de botella y reconocer patrones que podrían indicar la presencia de defectos. Además, realiza un análisis de causa raíz para determinar los factores subyacentes que originan dichos problemas. Como parte de este proceso, GenAI también evalúa el impacto potencial de los defectos sobre la funcionalidad global del sistema y la experiencia del usuario, facilitando una toma de decisiones más informada y proactiva en la gestión de calidad.
- **Apoyo Transversal del Proceso de Testing.** La IA Generativa se pueda utilizar de manera transversal en la totalidad de las actividades que conllevan el proceso de aseguramiento de calidad de software, de manera mucho más ágil, dinámica, en menor tiempo, y con un alto grado de precisión y eficiencia, aplicando mejoras significativas en tareas clave como lo son por ejemplo:

- Análisis de requerimientos
- Definición de estrategias de pruebas
- Generación de escenarios de prueba
- Análisis de resultados
- Generación de informes de resultados
- Predicción de defectos
- Generación de scripts de automatización
- Selección de casos para regresión
- Análisis e inspección de requerimientos



Ilustración 16: GenAI como apoyo al proceso de Testing

Fuente: (elaboración propia)

4. HIPÓTESIS

A través de este estudio se busca demostrar que ante un escenario de desarrollo de software dinámico, con requerimientos cambiantes y con una alta incertidumbre respecto a las preferencias de los usuarios, la implementación de estrategias de prueba adaptativas basadas en inteligencia artificial generativa (GenAI) optimizará significativamente el proceso de aseguramiento de calidad de software de manera transversal, ayudando a incrementar la eficiencia operativa y la cobertura de pruebas, reduciendo los tiempos y costos en actividades clave y mejorando la detección de defectos críticos en instancias tempranas.

Dentro de los beneficios que se postulan con la incorporación de GenAI como parte del proceso de pruebas se consideran 4 dimensiones:

- Análisis de requerimientos
 - Reducción del tiempo de inspección de requerimientos en un 400%.
 - Detección de defectos se incrementará en un 50%.
- Diseño y creación de matrices de prueba
 - Creación de matrices de prueba un 80% más rápido.
 - Cobertura de pruebas (generales) se aumentará en un 40%.
- Ejecución de pruebas
 - Ejecución de pruebas un 30% más rápida.
 - Reducción de falsos positivos en un 20%.
- Generación de informes de resultados
 - Generación de informes de resultados un 80% más rápido.

En particular, se proyecta una mejora transversal en indicadores clave de desempeño, tales como: time-to-market, la eficiencia operacional, la calidad del producto final y la agilidad organizacional para responder al cambio.

La incorporación de GenAI al proceso de pruebas de software permitirá:

- Mejorar la alineación entre requerimientos de negocio y calidad del producto, reduciendo los riesgos de fallas funcionales y técnicas que afectan la satisfacción del cliente y la reputación de la empresa.
- Reducir el time-to-market de nuevas funcionalidades, fortaleciendo la capacidad de respuesta ante oportunidades del mercado y acelerando la entrega de valor al cliente.

- Optimizar la utilización de recursos humanos y tecnológicos, aumentando la eficiencia operativa y reduciendo costos asociados a reprocesos y mantenimiento correctivo.
- Elevar la calidad del producto desde etapas tempranas del ciclo de vida del software, lo que impacta directamente en la fidelización del cliente, la reducción del churn rate y la mejora del Net Promoter Score (NPS).
- Proveer inteligencia accionable mediante reportes automatizados y precisos, lo cual mejora la toma de decisiones y el gobierno de TI en tiempo real.

5. OBJETIVOS

En toda iniciativa de transformación tecnológica, la definición clara de objetivos no solo orienta la ejecución operativa, sino que también asegura la alineación con los intereses estratégicos de la organización. Para los tomadores de decisión, resulta fundamental comprender que los objetivos de este estudio están directamente vinculados con la generación de eficiencia, la reducción de costos y la mejora sostenida de la calidad en los proyectos digitales.

5.1 Objetivo General

Proponer un modelo de optimización del proceso de aseguramiento de calidad de software, basado en estrategias de prueba adaptativas fundamentadas en Inteligencia Artificial Generativa (GenAI), orientado a reducir significativamente los tiempos y costos en actividades clave del proceso, incrementar la eficiencia operativa, mejorar la detección temprana de defectos críticos y ampliar la cobertura de pruebas, especialmente en contextos de desarrollo ágil con requerimientos cambiantes.

5.2 Objetivos Específicos

Dentro de los principales objetivos específicos de este trabajo de investigación se destacan:

- Identificar las principales limitaciones de los enfoques de prueba tradicionales en entornos de desarrollo ágil, particularmente en escenarios con requerimientos cambiantes, que afectan la eficiencia, cobertura y capacidad de respuesta del aseguramiento de calidad.
- Analizar el estado del arte sobre la aplicación de Inteligencia Artificial en el proceso de pruebas de software, con énfasis en el uso de Inteligencia Artificial Generativa (GenAI) para el diseño e implementación de estrategias de prueba adaptativas.
- Diseñar un modelo conceptual basado en GenAI que integre estrategias de prueba adaptativas orientadas a optimizar el proceso de aseguramiento de calidad de software, incluyendo mejoras cuantificables en análisis de requerimientos, generación de casos de prueba, ejecución y reporte de resultados.
- Definir los requisitos técnicos, organizacionales y humanos necesarios para la implementación efectiva del modelo propuesto, considerando aspectos de infraestructura tecnológica, capacitación, automatización y gestión del cambio.
- Evaluar el impacto potencial del modelo propuesto mediante indicadores cuantitativos asociados a eficiencia operativa, reducción de tiempos y costos, cobertura de pruebas y detección temprana de defectos críticos, validando su aplicabilidad en escenarios representativos del desarrollo ágil.

6. PLAN DE TRABAJO

Se propone trabajar sobre una base metodológica con enfoque cuantitativo, así como de los diseños exploratorios y descriptivos, con foco en investigaciones de tecnologías emergentes, innovadoras y poco estudiadas (Sampieri et al., 2014). Se establece un proceso de trabajo no dependiente de procedimientos experimentales complejos, ni recolección masiva de datos cuantitativos, facilitando así una investigación viable en tiempo y alcance, y con foco en un análisis exhaustivo de una tecnología emergente y su impacto en el proceso de pruebas.

Este enfoque metodológico permitirá analizar el impacto del modelo propuesto a través de indicadores cuantificables de eficiencia operativa, cobertura de pruebas, reducción de tiempos y costos, y capacidad de detección temprana de defectos, lo que proporciona evidencia objetiva sobre su viabilidad y efectividad en entornos ágiles y dinámicos.

6.1 Enfoque

Se propone desarrollar una investigación aplicada con enfoque cuantitativo, de tipo cuasi-experimental, combinando fases exploratorias iniciales con una validación cuantificable del modelo propuesto. Este enfoque permitirá medir de forma objetiva el impacto de la incorporación de estrategias de prueba adaptativas basadas en GenAI sobre el proceso de aseguramiento de calidad de software.

Este enfoque debe considerar las siguientes etapas clave:

1. Revisión exploratoria y descriptiva del estado del arte sobre el aseguramiento de calidad en entornos ágiles y la aplicación de GenAI en pruebas de software, para fundamentar el diseño del modelo.
2. Diseño del modelo conceptual que integre GenAI en las distintas etapas del proceso de pruebas, alineado con las dimensiones clave planteadas en la hipótesis (análisis de requerimientos, diseño de pruebas, ejecución y reportes).

3. Implementación controlada del modelo en entornos simulados o casos de estudio reales, donde se pueda aplicar una comparación antes/después respecto a tiempos, costos, cobertura y detección de defectos.
4. Recolección y análisis estadístico de datos cuantitativos, utilizando indicadores definidos para cada dimensión (por ejemplo, tiempo promedio de generación de casos, porcentaje de cobertura, tasa de defectos detectados, etc.).
5. Validación de la hipótesis mediante técnicas de inferencia estadística que permitan determinar si los cambios observados son significativos.

6.2 Diseño Metodológico

Para abordar de manera robusta los objetivos transversales de este trabajo de investigación se ha optado por un diseño metodológico de carácter exploratorio-descriptivo (Sampieri et al., 2014). La naturaleza exploratoria se justifica por la necesidad de examinar un campo relativamente nuevo y en evolución, como lo es el uso de GenAI en estrategias de prueba adaptativas, mientras que la dimensión descriptiva se enfoca en desarrollar el estado actual de las prácticas de testing, identificando sus limitaciones en entornos dinámicos, y detallando los beneficios y desafíos de la GenAI aplicada en ese contexto.

La selección de este diseño metodológico se sustenta en el carácter emergente e innovador de la tecnología estudiada, la cual aún no cuenta con un cuerpo teórico suficientemente consolidado en el ámbito académico y profesional. El componente exploratorio del diseño permitirá indagar preliminarmente el estado del arte, identificando los desafíos actuales del aseguramiento de calidad de software en contextos cambiantes y así detectar cómo las estrategias adaptativas están comenzando a utilizarse en la industria del QA. Posteriormente, el componente descriptivo permitirá caracterizar con mayor detalle las prácticas actuales, tecnologías utilizadas, beneficios observados y limitaciones asociadas a la incorporación de GenAI en los procesos de prueba de software. A través de esta fase se buscará documentar de forma sistemática cómo estas estrategias impactan aspectos como la eficiencia, cobertura y capacidad de adaptación del proceso de testing, con base en evidencia concreta y estudios especializados.

En conjunto, este enfoque metodológico mixto (exploratorio-descriptivo) resulta especialmente pertinente para investigaciones aplicadas que requieren una aproximación flexible pero rigurosa, con el fin de comprender, analizar y proponer mejoras concretas sobre un fenómeno contemporáneo de alta complejidad tecnológica.

6.3 Técnicas e Instrumentos

Para alcanzar los objetivos trazados y garantizar la validez y eficiencia del modelo propuesto, se emplearán diversas técnicas e instrumentos cualitativos que buscan facilitar un análisis profundo y contextualizado del tema de estudio.

- **Revisión sistemática de documentación.** Se aplicará una metodología estructurada para la búsqueda y análisis de estudios académicos, documentos técnicos, artículos científicos y reportes de la industria. Esta revisión permitirá identificar factores clave, tales como: fortalezas y debilidades de la IA, tendencias emergentes en GenAI aplicada al testing y limitaciones de enfoques tradicionales en contextos ágiles.
- **Análisis documental.** Se examinarán marcos de trabajo internacionales de calidad de software (como los propuestos por ISTQB®), documentación y estudios de mecanismos de integración de IA en contextos de pruebas y guías de implementación de GenAI. El análisis se enfocará en identificar principios, estructuras, procesos y métricas aplicables a un modelo adaptativo de testing.
- **Diseño de un modelo conceptual.** A partir del análisis se procederá a diseñar un modelo conceptual que integre estrategias de prueba adaptativas basadas en GenAI dentro del proceso de aseguramiento de calidad del software. Este modelo será representado mediante diagramas de flujo y esquemas funcionales, considerando criterios de eficiencia, escalabilidad, cobertura y alineación con entornos ágiles.
- **Triangulación de fuentes.** Para aumentar la consistencia y confiabilidad de los resultados, se aplicará una estrategia de triangulación metodológica de fuentes de datos, contrastando información proveniente de distintas fuentes (literatura, estándares, casos de éxito, etc.) con el fin de fortalecer las conclusiones y reducir sesgos interpretativos (Gómez et al., 1996).



Ilustración 17: Metodología de investigación

Fuente: Elaboración propia

7. PROPUESTA

Proponer una nueva estrategia tecnológica en el ámbito del aseguramiento de calidad de software no implica únicamente optimizar procesos, sino redefinir su rol como un motor de agilidad, innovación y gestión del riesgo. Esta propuesta busca demostrar que la incorporación de GenAI trasciende la modernización operativa: representa una transformación profunda en la manera en que las organizaciones garantizan la calidad, aceleran sus entregas y minimizan el impacto financiero de los defectos no detectados de manera temprana. Desde una perspectiva gerencial, no se trata solo de una iniciativa viable, sino de una respuesta estratégica a las exigencias del entorno digital actual, caracterizado por su alta competitividad y en constante evolución.

Garantizar la calidad y la fiabilidad de los productos de software es un aspecto esencial en su proceso de desarrollo. No obstante, pese a los constantes avances tecnológicos, los enfoques tradicionales de pruebas suelen resultar insuficientes para adaptarse a la creciente complejidad y envergadura de los sistemas actuales, además de las necesidades de los usuarios finales. Los procedimientos manuales de prueba demandan una alta inversión de tiempo y recursos, además de ser

propensos a errores humanos, lo que genera ineficiencias y aumenta el riesgo de omitir fallas críticas (Deming, 2021). En este contexto, se hace evidente la necesidad urgente de adoptar métodos de prueba de software de vanguardia, capaces de superar estas limitaciones y llevar el aseguramiento de la calidad (QA) a nuevos niveles de eficacia y precisión.

A pesar de los avances en las metodologías de pruebas de software, persiste un importante vacío de investigación en torno a la aplicación eficiente de la automatización, el aprendizaje automático (ML) y la inteligencia artificial (IA). Si bien es cierto la IA y el ML se han adoptado ampliamente en diversos sectores, su integración en los procesos de prueba de software aún se encuentra en una etapa prematura. La mayoría de las publicaciones existentes se centran en marcos teóricos y estudios de prueba de concepto, con escasa validación y pocas aplicaciones en entornos reales. Además, se requiere una investigación más profunda que evalúe la efectividad, escalabilidad y aplicabilidad de las metodologías de pruebas impulsadas por IA en las distintas etapas del proceso de desarrollo de software. En este sentido, es fundamental cerrar esta brecha mediante estudios robustos que vinculen los fundamentos teóricos con metodologías de implementación viables y análisis del impacto en escenarios del mundo real.

Este trabajo pretende explorar la viabilidad de incorporar el aprendizaje automático en los procesos de prueba de software, con el objetivo claro de mejorar la eficacia y eficiencia del aseguramiento de la calidad. Se propone analizar tecnologías de vanguardia, metodologías de trabajo y estrategias de pruebas dentro del contexto de la inteligencia artificial y particularmente en la IA Generativa. Como parte del análisis se identificarán las mejores prácticas de la industria, desafíos y limitaciones asociadas al uso de pruebas basadas en IA en entornos de trabajo reales, además de ofrecer recomendaciones orientadas a facilitar la adopción e integración de estas metodologías por parte de los equipos de trabajo. En base a los antecedentes recopilados en este capítulo se buscará confirmar o descartar las hipótesis planteadas en instancias previas de este trabajo para cada uno de los puntos de mejora identificados.

7.1 Evolución de la IA en las Pruebas de Software

La integración de técnicas de aprendizaje automático en la predicción de defectos ha representado un avance sustancial en el aseguramiento proactivo de la calidad del software. El análisis estático tradicional y la predicción de defectos basada en métricas han sido progresivamente reemplazados por modelos de aprendizaje automático más avanzados, capaces de identificar patrones complejos que anticipan posibles fallas y puntos de mejora en el software (Reddy, 2025). En un

estudio realizado por (Tian, 2020) se llevó a cabo un análisis comparativo de diversos algoritmos de aprendizaje automático aplicados a la predicción de defectos de software, evaluando su efectividad en distintos repositorios de código y contextos de desarrollo. La investigación evidenció que los enfoques de aprendizaje supervisado pueden aprovechar datos históricos de defectos para anticipar futuros problemas de calidad, permitiendo así focalizar los esfuerzos de prueba de manera más eficiente. La incorporación del aprendizaje automático en esta área marcó una transición clave desde enfoques de prueba reactivos hacia un aseguramiento de la calidad de carácter predictivo, transformando significativamente la manera en que las organizaciones asignan recursos de prueba y priorizan intervenciones de mejora.

Las aplicaciones modernas de inteligencia artificial en el aseguramiento de la calidad van más allá de la predicción de defectos, abarcando múltiples fases del ciclo de vida de las pruebas. Entre las implementaciones actuales se destacan el uso de procesamiento de lenguaje natural (NLP) para el análisis de requerimientos y la generación automatizada de casos de prueba, la visión artificial para la validación visual de interfaces de usuario, y el aprendizaje por refuerzo para optimizar la ejecución de pruebas. Estas tecnologías han permitido a las organizaciones adoptar estrategias inteligentes de selección y priorización de pruebas, maximizando la cobertura y reduciendo significativamente los tiempos de ejecución. Asimismo, los sistemas basados en IA están fortaleciendo las pruebas exploratorias al detectar casos extremos y generar escenarios que podrían pasar desapercibidos para los evaluadores humanos. La integración de estas capacidades en entornos de integración continua ha potenciado la adopción de prácticas shift-left testing, donde el aseguramiento de la calidad se inicia en etapas tempranas del desarrollo y se mantiene como un proceso continuo a lo largo del ciclo de vida del producto.

La inteligencia artificial está desempeñando un papel fundamental en los procesos de integración continua, ya que optimiza el proceso de CI al proporcionar información en tiempo real, automatizar el flujo de trabajo de implementación y optimizar los flujos de trabajo. Al integrar la IA en el flujo de trabajo los equipos de desarrollo pueden lograr ciclos de retroalimentación más tempranos, lo que garantiza que los posibles problemas se identifiquen y resuelvan con rapidez, lo que se traduce en lanzamientos de software más fiables y consistentes. En este punto resulta clave profundizar en tres conceptos fundamentales que muchas veces se tienden a confundir y usar erróneamente (King, 2019):

- **AI Testing.** Corresponde a la utilización de tecnologías IA/ML en la ejecución automatizada de actividades de prueba y se conoce formalmente como “AI-Driven Testing”. Más allá de los avances tecnológicos, los analistas (humanos) siguen siendo responsables de definir los objetivos de prueba, adquirir el conocimiento funcional del sistema, diseñar y especificar escenarios de prueba detallados, escribir scripts de automatización, ejecutar

pruebas que no pueden automatizarse y analizar los resultados para identificar posibles defectos.

Dado que la práctica predominante continúa girando en torno a las pruebas manuales y a la creación manual de scripts, tanto investigadores como profesionales han comenzado a explorar cómo la inteligencia artificial (IA) y el aprendizaje automático (ML) pueden habilitar una nueva generación de herramientas de prueba. Esta nueva visión busca aprovechar los avances en IA/ML, computación en la nube y tecnologías de big data para cerrar la brecha entre las pruebas manuales y las pruebas automatizadas.

- **Testing AI.** Corresponde al proceso de realizar pruebas a sistemas o aplicaciones de software que hayan sido estructuradas sobre tecnologías de inteligencia artificial y aprendizaje automático, lo que plantea nuevos desafíos en materia de calidad y pruebas de software. Los sistemas impulsados por IA suelen ser complejos, no deterministas y autoadaptativos, lo que complica su evaluación mediante enfoques tradicionales. Además, introducen problemáticas específicas relacionadas con el cálculo de características, la calidad del muestreo, la tolerancia a valores atípicos, la precisión de las etiquetas, la desviación de calidad y la trazabilidad de los resultados.

Los productos basados en ML, por su parte, pueden requerir un reentrenamiento completo con cada nueva versión, y su comportamiento resulta difícil de predecir y depurar. Ante este escenario, se incrementa la necesidad de contar con técnicas de prueba aplicadas, conjuntos de herramientas estandarizadas y buenas prácticas específicamente diseñadas para validar sistemas basados en IA.

- **Self Testing.** Hace referencia a la capacidad de un sistema o componente para supervisar su propio comportamiento adaptativo y ejecutar pruebas en tiempo de ejecución, ya sea antes o como parte integral del proceso de adaptación.

La adaptación dinámica en sistemas basados en IA puede ser: estocástica (resultante de la selección aleatoria durante el aprendizaje) o ambiental (aprendizaje basado en la retroalimentación que se transmite al sistema desde los usuarios finales u otros sistemas). Sin embargo, independientemente de la fuente, la adaptación provoca que la lógica interna del sistema cambie en tiempo de ejecución.

7.2 AI-Driven Testing

Resulta evidente que la incorporación de la inteligencia artificial ha revolucionado el desarrollo de software en la era moderna, alterando los métodos y enfoques convencionales en múltiples tareas transversales a lo largo del proceso. Las pruebas de software son un campo que ha experimentado una rápida innovación gracias a técnicas de IA, tales como: la automatización y el aprendizaje automático, transformando positivamente las actividades de pruebas para que tengan un carácter mucho más adaptativo a los cambios de los proyectos. En palabras simples las pruebas basadas en IA corresponden a un enfoque de aseguramiento de calidad de software que utiliza técnicas de inteligencia artificial para mejorar, automatizar y optimizar las actividades de prueba.

Los principios fundamentales de las pruebas basadas en inteligencia artificial (AI-Driven Testing) se centran en la capacidad de los sistemas inteligentes para aprender a partir de datos históricos, adaptarse a la evolución del software y tomar decisiones autónomas sobre qué partes de la aplicación deben ser evaluadas. Las herramientas de prueba impulsadas por IA pueden generar casos de prueba de manera automática, predecir posibles fallos y optimizar continuamente sus estrategias a medida que procesan nuevos datos. Esta capacidad adaptativa convierte a este tipo de pruebas en una solución poderosa para enfrentar los desafíos de complejidad y escalabilidad en los sistemas de software modernos, especialmente en entornos ágiles y DevOps, donde los ciclos de desarrollo acelerados y los cambios frecuentes en el código son constantes (Svensson, 2018).

Las tecnologías asociadas a la inteligencia artificial permiten que las herramientas de prueba basadas en IA aprendan automáticamente a partir de ejecuciones anteriores, optimicen continuamente las estrategias de prueba y detecten áreas de alto riesgo dentro del software. Asimismo, el procesamiento de lenguaje natural contribuye significativamente al proceso al traducir requisitos de negocio, historias de usuario o especificaciones técnicas en casos de prueba ejecutables. Esto no solo agiliza la generación de pruebas, sino que también facilita la colaboración entre desarrolladores, analistas de pruebas y partes interesadas no técnicas, al reducir las barreras de comunicación y mejorar la trazabilidad entre los requerimientos y la validación.

Las pruebas basadas en inteligencia artificial se sustentan en varios enfoques fundamentales (Deming, 2021), cada uno con funciones y usos únicos en los procedimientos de control de calidad. Uno de estos métodos es la creación automatizada de pruebas, en la que algoritmos de aprendizaje automático analizan las especificaciones del software y los datos de pruebas anteriores para crear casos de prueba que optimizan la cobertura del código y la detección de fallos

automáticamente. Los probadores pueden concentrarse en tareas de prueba de mayor nivel, ya que la generación automatizada de pruebas minimiza considerablemente el trabajo manual necesario para el diseño de casos de prueba. Otro método crucial es la priorización inteligente de pruebas, que utiliza modelos de aprendizaje automático para clasificar los casos de prueba según su propensión a detectar fallas o vulnerabilidades críticas. La priorización inteligente de pruebas, especialmente en entornos de prueba con limitaciones de tiempo, optimiza los recursos de prueba y acelera la detección de fallos al modificar dinámicamente las secuencias de ejecución de las pruebas.

El rápido posicionamiento de este nuevo enfoque de pruebas ha transformado significativamente el rol de los profesionales del área de aseguramiento de calidad del software. Actualmente, los esfuerzos de los analistas de prueba se orientan más hacia la formulación de estrategias, el diseño de pruebas y el análisis de los resultados generados por herramientas basadas en IA, desplazando progresivamente la ejecución manual. Esta evolución implica la necesidad de adquirir nuevas competencias, ya que los profesionales deben adaptarse a un entorno laboral híbrido que combine la experiencia humana con la inteligencia artificial. En este contexto, la IA representa una vía prometedora para enfrentar los desafíos asociados a entornos de software en constante cambio, aportando la eficiencia necesaria para garantizar la fiabilidad y escalabilidad del software en todos los proyectos.

Esta tendencia promete un futuro donde las organizaciones podrán ofrecer software de alta calidad con mayor rapidez y eficacia, adaptándose a los rápidos ciclos de desarrollo y a la creciente complejidad de las aplicaciones actuales. Sin embargo, resulta conveniente repasar los principales beneficios que conllevan las pruebas basadas en inteligencia artificial y los posibles inconvenientes tras su implementación (ACCELQ, 2025):

BENEFICIOS:

- **Cobertura de pruebas mejorada.** Las pruebas manuales suelen depender de recursos humanos para generar y ejecutar casos de prueba, lo cual puede estar limitado por su experiencia y conocimiento. La IA, por otro lado, puede encontrar y generar una amplia gama de escenarios de prueba mediante el análisis de grandes cantidades de datos, como casos de prueba anteriores, actividad del usuario y registros de aplicaciones. La IA facilita la detección de problemas que, de otro modo, podrían pasar desapercibidos, ampliando y profundizando el alcance de las pruebas.
- **Mejora en la precisión.** La IA reduce los errores humanos al permitir el reconocimiento preciso de fallas y minimizar el riesgo de pasar por alto

problemas críticos. Las pruebas manuales son susceptibles a errores humanos. Las tecnologías de prueba impulsadas por IA evitan estos riesgos al adherirse consistentemente a algoritmos y procesos preestablecidos. También pueden examinar patrones de datos complejos en busca de anomalías que puedan revelar fallas. Este grado de precisión garantiza que los errores se reconozcan y reporten correctamente, lo que resulta en un software más confiable. Además, la IA puede aprender de fallas previas y de los resultados de pruebas para mejorar su precisión con el tiempo.

- **Ahorro de costos.** Las herramientas de prueba basadas en IA son caras, pero el ahorro a largo plazo es considerable. Eliminan la necesidad de un gran grupo de probadores manuales, lo que reduce los gastos de mano de obra. Además, la mayor productividad y rapidez de las pruebas basadas en IA reduce el tiempo total de prueba, lo que tiende a reducir aún más los gastos. Una mayor precisión implica menos fallos sin detectar, lo que reduce el costo de la reparación de errores y el mantenimiento posterior al lanzamiento.
- **Eficiencia y velocidad.** La IA acelera el proceso de pruebas al reducir el tiempo necesario para ejecutarlas y detectar errores, lo cual es fundamental para la integración y la entrega continua. Las tecnologías de pruebas automatizadas basadas en IA pueden realizar numerosas pruebas de forma simultánea y constante, superando considerablemente a los probadores humanos. Esta velocidad es crucial en contextos ágiles y DevOps, donde los ciclos de desarrollo cortos requieren pruebas frecuentes. La IA es capaz de adaptarse rápidamente a cualquier modificación en el producto o la aplicación, garantizando que las pruebas sean relevantes y eficaces.
- **Análisis predictivo.** La IA puede detectar posibles errores y problemas con antelación, lo que permite a los equipos solucionarlos de forma proactiva. Al estudiar datos previos, la IA puede detectar comportamientos y patrones que indican errores. El análisis predictivo también facilita la planificación de recursos y la gestión de riesgos, de modo que los recursos de prueba se utilicen de forma eficiente.
- **Automatización inteligente.** La IA automatiza las actividades de prueba repetitivas, lo que permite a los evaluadores dedicar sus esfuerzos a casos más complejos. Por ejemplo, las pruebas de regresión son necesarias para garantizar que cualquier cambio en el código no afecte su funcionalidad. Sin embargo, realizar todo manualmente requiere mucho tiempo y es laborioso. Estos procesos pueden automatizarse mediante tecnologías basadas en IA, lo que permite una ejecución rápida de las pruebas de regresión.
- **Experiencia de usuario mejorada.** Al recopilar y evaluar datos sobre cómo los usuarios usan una aplicación, la IA puede detectar patrones de uso frecuentes y posibles áreas problemáticas. Estos datos pueden utilizarse

para desarrollar escenarios de prueba que simulen el uso real. La IA también puede imitar las interacciones del usuario para comprender la utilidad de una aplicación. La IA en las pruebas garantiza que el producto final satisfaga las expectativas del usuario y ofrezca una experiencia atractiva.

- **Validación visual.** La inteligencia artificial posee capacidades avanzadas de reconocimiento de patrones e imágenes que permiten realizar pruebas visuales eficaces en las aplicaciones. Estas capacidades facilitan la detección automática de errores visuales, como desalineaciones, elementos faltantes o inconsistencias en la interfaz. Gracias a ello, se garantiza que todos los componentes visuales no solo funcionen correctamente, sino que también mantengan una presentación coherente y atractiva para el usuario final.



Ilustración 18: Beneficios del uso de AI-Driven Testing

Fuente: (TestBytes, 2023)

DESVENTAJAS:

- **Mayor deuda técnica.** El uso de IA podría complicar los procesos de prueba. Si no se gestiona adecuadamente, esto puede generar una deuda técnica creciente, lo que requiere esfuerzo y recursos adicionales para su mantenimiento y reelaboración con el tiempo.

- **El exceso de pruebas consume recursos.** La IA puede, ocasionalmente, generar pruebas excesivas, lo que requiere recursos informáticos y tiempo adicionales. El exceso de pruebas puede generar ineficiencias y sobrecargar la infraestructura de pruebas.
- **Desconocimiento de los procesos de negocio.** Es posible que la IA no comprenda plenamente las complejidades de los procesos de negocio dentro de la aplicación (conocimiento que muchas veces si posee el analista de pruebas). Esto podría generar discrepancias en la cobertura de las pruebas y casos de prueba incompatibles.
- **Necesidad de habilidades técnicas adicionales.** Configurar y gestionar soluciones de pruebas basadas en IA a veces requiere conocimientos técnicos sofisticados. Esto puede ser un desafío para empresas con conocimientos técnicos limitados y poco presupuesto para invertir en instancias de capacitación.
- **Incremento de falsos positivos.** Las pruebas basadas en IA pueden ocasionalmente generar fallos incorrectos, lo que resulta en falsos positivos. Estos falsos positivos pueden tardar tiempo, ya que los equipos analizan problemas inexistentes, lo que reduce la productividad.

Las pruebas basadas en IA representan un cambio de paradigma fundamental, con estudios cuantitativos (Kumar, 2025) que demuestran mejoras mensurables en métricas clave. Las investigaciones indican que la automatización de pruebas impulsada por IA reduce el tiempo de prueba en un 75% y mejora la cobertura de las pruebas en un 60% en comparación con los métodos tradicionales. Las organizaciones que implementan enfoques de pruebas basadas en IA reportan una reducción promedio del 40% en los costos de prueba, a la vez que detectan un 37% más de defectos.

En la misma línea, la automatización de pruebas con autorreparación ha demostrado una eficacia especial. Estudios demuestran que los scripts de prueba basados en IA se adaptan automáticamente a los cambios de interfaz en el 85% de los casos sin intervención humana. Las capacidades de análisis predictivo han demostrado una precisión notable, ya que los modelos de IA identifican correctamente el 68% de los componentes de alto riesgo antes de comenzar las pruebas, lo que permite una mayor focalización de estas. El impacto económico es considerable; estudios indican que los enfoques de pruebas basados en IA ofrecen un retorno de la inversión (ROI) promedio del 283% durante el primer año de implementación. Por último, los beneficios compuestos de la optimización continua son particularmente notables: las investigaciones demuestran que los marcos de prueba de IA mejoran su efectividad aproximadamente entre un 5% y un 8% con

cada ciclo de prueba a medida que los modelos refinan su comprensión del comportamiento de la aplicación.

Indicador Estratégico	Pruebas Tradicionales	Modelo con GenAI	Impacto para la Gestión Ejecutiva
Eficiencia Operacional en QA	Alta dependencia de recursos (70%)	Uso optimizado de recursos (~30%)	Reducción del 40% en consumo de recursos, liberando capital humano y operativo.
Velocidad de Generación de Pruebas	4.5 horas por caso	~1.1 horas por caso	Reducción del 75% en tiempos, acelerando entregas y reduciendo cuello de botella.
Optimización del Presupuesto de TI en QA	35-40% del presupuesto	~20% del presupuesto	Disminución del 15-20% en gastos operativos, alineando inversión a resultados.
Agilidad en Respuesta a Cambios	Actualización manual	85% de automatización	Alta adaptabilidad al cambio, clave en entornos ágiles e inciertos.
Cobertura de Riesgos en QA	40% cobertura de pruebas	~64% cobertura	Mayor mitigación de riesgos técnicos y operativos (~60%).
Reducción del Time-to-Market	Tiempo base	Reducción del 75%	Mayor competitividad mediante ciclos de entrega más cortos.
Reducción de Costos en Testing	Costo base	40% menos	Ahorro directo en procesos críticos sin sacrificar calidad.
Gestión Predictiva de Riesgos Críticos	Análisis manual	68% precisión automatizada	Mejor toma de decisiones en etapa temprana, reduciendo retrabajos.
ROI de la Automatización con GenAI	Bajo	Alto, con mejoras medibles	Retorno acelerado de la inversión tecnológica en QA.

Ilustración 19: Pruebas Tradicionales vs AI-Driven Testing

Fuente: (Kumar, 2025)

7.3 Estrategia de Implementación de GenAI en el Testing

La principal ventaja de las soluciones de inteligencia artificial generativa (GenAI), tales como: ChatGPT, Copilot, entre muchas otras, radica en sus avanzadas capacidades de procesamiento y generación de texto. Estas herramientas resultan especialmente valiosas en el ámbito de las pruebas de software, donde es común enfrentar cargas de trabajo que implican la gestión de grandes volúmenes de datos, múltiples escenarios de prueba y la elaboración de informes de defectos detallados. Además del análisis y la generación de texto, las herramientas GenAI también pueden realizar análisis predictivos basados en sus datos de entrenamiento. Para los analistas de pruebas, esto se traduce en información sobre posibles puntos de fallo, predicciones sobre el comportamiento del usuario y recomendaciones para optimizar los procesos de prueba. Estas predicciones se basan en la capacidad de la IA para aprender de datos históricos e identificar patrones, lo que permite anticipar los problemas antes de que se materialicen.

Una de las capacidades clave de los modelos de lenguaje de gran tamaño (LLM), que sustentan las herramientas GenAI, es su adaptabilidad y capacidad de aprendizaje continuo. En el ámbito de las pruebas de software, esto se traduce en la posibilidad de personalizar estas herramientas para que reconozcan terminologías específicas y se alineen con flujos de trabajo particulares de cada proyecto. A medida que se utilizan con mayor frecuencia, las herramientas GenAI mejoran su desempeño, ofreciendo un soporte cada vez más relevante y personalizado, lo que contribuye significativamente a optimizar la eficiencia y la efectividad del proceso de pruebas (Goyal, 2024). Para implementar con éxito una IA generativa es necesario desarrollar una estrategia meditada que combine tecnologías existentes y nuevas bajo una estructura de gobernanza que habilite capacidades de calidad centrales y esté construida en torno a la colaboración con asociaciones industriales y reguladores.

En el ámbito de las pruebas de software, la estrategia es fundamental. Elaborar una estrategia de pruebas eficaz es una tarea compleja que requiere previsión, adaptabilidad y un profundo conocimiento tanto del software en cuestión como de las metodologías más adecuadas para probarlo. En general, existen seis etapas principales que componen la estrategia de un proyecto de pruebas de software impulsado por GenAI (Zymr, 2024). Cada una de ellas requiere la participación de las partes interesadas, con un profundo conocimiento de las necesidades de software actuales y futuras de la empresa, la dinámica de su equipo de pruebas y los resultados que tendrán mayor impacto al negocio.

7.3.1 Evaluación de los Procesos de Pruebas Actuales

Se deben evaluar los procesos de prueba actuales e identificar las áreas donde la inteligencia artificial puede generar mayor valor. Lo primero es analizar en profundidad los distintos componentes del ciclo de pruebas de la organización, incluyendo los tipos de pruebas que se ejecutan (por ejemplo, pruebas manuales o automatizadas), las competencias del equipo de pruebas, y la eficiencia y efectividad de los flujos de trabajo actuales.

Asimismo, es fundamental identificar los principales puntos débiles y desafíos que enfrentan los equipos a lo largo del ciclo de vida de las pruebas de software. Este análisis permitirá detectar escenarios específicos donde la IA puede tener un impacto significativo, tales como: el análisis de requerimientos, la automatización de pruebas repetitivas, el uso de análisis predictivo para la detección anticipada de defectos, la generación automatizada de casos de prueba, o la optimización del rendimiento general del proceso de aseguramiento de calidad.

7.3.2 Herramientas e Infraestructura

Una vez identificados los casos de uso, es fundamental evaluar las herramientas y tecnologías actualmente empleadas en los procesos de prueba, incluyendo plataformas de gestión de pruebas, marcos de automatización, sistemas de seguimiento de errores y entornos de integración y entrega continua (CI/CD). Asimismo, el análisis detallado del ciclo de vida de los defectos, desde su detección y reporte, hasta su priorización y resolución, permite identificar cuellos de botella críticos que afectan la eficiencia del proceso. Esta evaluación facilita la toma de decisiones informadas para invertir en la infraestructura y las herramientas necesarias que respalden iniciativas de pruebas potenciadas por IA.

Entre estas herramientas se incluyen plataformas especializadas en pruebas basadas en inteligencia artificial, algoritmos de aprendizaje automático y soluciones avanzadas de análisis de datos. Para maximizar el impacto, es imprescindible abordar de manera integral los puntos críticos clave, como la gestión de recursos, las tasas de defectos, la cobertura de pruebas y la capacidad de escalabilidad.

7.3.3 Recopilación de Datos y Generación de Información

Una vez definidas las bases de trabajo, es posible comenzar a establecer prácticas sólidas para la implementación de pruebas basadas en inteligencia artificial. El primer paso clave consiste en diseñar procesos sistemáticos para la recopilación, el preprocesamiento y la gestión de datos. Esto implica reunir conjuntos de datos relevantes y representativos para los objetivos de prueba, priorizando su calidad, relevancia y cumplimiento normativo en materia de privacidad y protección de datos.

Para garantizar que los datos sean adecuados para el análisis, se aplican técnicas rigurosas de preprocesamiento, como la limpieza de datos, la normalización de valores y la ingeniería de características. Estas prácticas permiten estructurar los datos de manera óptima para alimentar modelos de IA y algoritmos de aprendizaje automático. El cumplimiento de las normativas de privacidad asegura el uso ético y legal de la información sensible o personal. Al mantener estándares elevados en la gestión de datos, las organizaciones pueden obtener información más precisa, tomar decisiones basadas en evidencia y mejorar la calidad, confiabilidad y eficiencia de las pruebas de software impulsadas por IA.

7.3.4 Transformación Cultural

Para impulsar la adopción de la inteligencia artificial en los procesos de prueba, es fundamental ofrecer programas de capacitación integrales que proporcionen a los equipos de trabajo los conocimientos necesarios sobre IA y promuevan una cultura de aprendizaje continuo. Capacitar a los evaluadores para experimentar con herramientas y técnicas basadas en IA fomenta la innovación y fortalece sus competencias técnicas. En este punto es valioso poder invertir en certificaciones robustas del uso de GenAI en las pruebas de software, como por ejemplo: AiU-Certified GenAI-Assisted Test Engineer de Brightest.

Paralelamente, es recomendable iniciar proyectos piloto que permitan evaluar la efectividad de la IA en contextos reales. Estos pilotos deben enfocarse en casos de uso representativos y contemplar la medición de indicadores clave de rendimiento, como la cobertura de pruebas, la tasa de detección de defectos y la duración de los ciclos de prueba. Este enfoque iterativo facilitará una integración progresiva y controlada de la IA en los procesos de aseguramiento de calidad, asegurando al mismo tiempo su alineación con los objetivos estratégicos de la organización y permitiendo ajustes continuos en función de los resultados observados.

7.3.5 Integración con el Flujo de Trabajo de Desarrollo

Se debe buscar la integración de las pruebas basadas en inteligencia artificial dentro del flujo de trabajo de desarrollo, con el objetivo de garantizar una colaboración fluida entre los equipos de desarrollo y el aseguramiento de calidad del software. La incorporación de canales de prueba automatizados y la implementación de procesos de integración y despliegue continuo (CI/CD) son elementos clave para acelerar la retroalimentación y los ciclos de iteración.

La integración permitirá optimizar la entrega continua de software, facilitando la detección temprana de defectos y vulnerabilidades a lo largo del ciclo de desarrollo. Las herramientas de prueba impulsadas por IA no solo automatizan tareas críticas, sino que también generan información valiosa sobre posibles problemas, lo que permite adoptar medidas proactivas para preservar la calidad y la seguridad del producto. Al integrar la IA en los procesos de prueba y desarrollo, las organizaciones mejoran significativamente la eficiencia operativa, acortan el tiempo de comercialización y fortalecen la calidad, la seguridad y la productividad general del software entregado.

7.3.6 Monitoreo y Mejora Continua

La monitorización continua de las soluciones de pruebas basadas en inteligencia artificial es esencial para mantener un rendimiento óptimo y sostenido. Para ello, es necesario establecer métricas clave de desempeño, así como mecanismos sistemáticos para recopilar retroalimentación del equipo de pruebas. Esta información permite identificar oportunidades de mejora y ajustar los algoritmos de IA en función de las necesidades reales del entorno. La calibración precisa de los parámetros del sistema, junto con el escalado adecuado de la infraestructura, asegura que las soluciones de IA se mantengan alineadas con los requerimientos cambiantes del proceso de pruebas y continúen entregando resultados consistentes y fiables.

Este enfoque iterativo permite optimizar de forma continua la productividad, la calidad y la seguridad del software, asegurando que las pruebas basadas en IA sigan siendo eficaces y pertinentes frente a la evolución del contexto tecnológico. A través de evaluaciones periódicas y mejoras continuas, las organizaciones pueden maximizar los beneficios de la IA aplicada al aseguramiento de calidad, sin comprometer los más altos estándares en seguridad y desempeño.



Ilustración 20: Estrategia de Implementación de GenAI en QA

Fuente: Elaboración propia

Para finalizar, podemos mencionar algunos factores generales que se deben tener en consideración al momento de implementar pruebas basadas en GenAI:

- Seleccione las tareas correctas. No todos los tipos de pruebas se pueden gestionar con IA. Esta tecnología se adapta mejor a tareas que requieren un

análisis extenso de datos, la búsqueda de patrones y la toma de decisiones. Sin embargo, presenta dificultades para gestionar pruebas que requieren un enfoque centrado en el usuario.

- Considere la cantidad y la calidad de los datos. Para ejecutar pruebas exitosas, las organizaciones deben entrenar modelos de IA con Node o Python para que reconozcan errores en la estructura del código. La calidad y el tamaño de esta información influirán directamente en la eficacia de las pruebas basadas en IA.
- Establezca objetivos e interprete los resultados. La organización debe comprender claramente las tareas que debe realizar la inteligencia artificial y cómo evaluar sus resultados. Establecer objetivos incorrectos conduce a errores o resultados indeseables.
- Integración con el proceso actual. Es vital considerar la IA como un componente del proceso de pruebas, no como su sustituto. Las herramientas que utilizan esta tecnología deben integrarse cuidadosamente en los procesos y prácticas existentes.
- Utilice un enfoque colaborativo. Para comprender las principales ventajas de la IA en las pruebas de software, las organizaciones deben establecer un equipo sincronizado de desarrolladores y probadores. Su experiencia puede ayudar a maximizar los esfuerzos de su implementación.

7.4 Mejoras Percibidas (Métricas)

La incorporación de la Inteligencia Artificial Generativa en el ámbito del aseguramiento de calidad de software hace mucho tiempo ha dejado de ser una promesa futurista para convertirse en una herramienta tangible y transformadora. En esta sección revisaremos las aplicaciones prácticas más relevantes de la GenAI en el contexto de las pruebas, poniendo especial énfasis en los casos de uso que han demostrado beneficios medibles en términos de eficiencia, cobertura, velocidad y precisión. A través del análisis de estudios previos, se explorará cómo la IA generativa potencia estrategias de prueba adaptativas, permitiendo una respuesta dinámica a cambios en los requisitos, arquitectura o comportamiento del sistema.

De manera complementaria se revisarán métricas clave que buscan evidenciar la mejora en las distintas etapas del proceso de pruebas, así como los beneficios tangibles que impactan directamente en la calidad del software y en la reducción de costos de desarrollo. Esta perspectiva práctica busca demostrar no solo la viabilidad técnica de llevar a cabo esta implementación, sino también el valor estratégico que la GenAI aporta al ciclo de vida del software.

A través de un ejemplo detallado (Hall, 2024), se ilustrará cómo la IA Generativa puede gestionar las siguientes tres tareas críticas en un entorno típico de desarrollo de software:

- ✓ Generación automatizada de casos de prueba
- ✓ Ejecución mejorada de pruebas
- ✓ Análisis de resultados de pruebas

Pensemos en un equipo de desarrollo de software trabajando en una plataforma de comercio electrónico compleja. Necesitan garantizar que las nuevas funciones, correcciones de errores y actualizaciones no generen defectos y que la plataforma se mantenga robusta y segura. La plataforma incluye varios componentes como la autenticación de usuarios, el catálogo de productos, el carrito de compra, el procesamiento de pagos y la gestión de pedidos.

- **Generación Automatizada de Casos de Prueba.** La AI Generativa analiza historias de usuario y documentos de requerimientos para comprender la funcionalidad esperada del sistema, aprovechando datos históricos de pruebas para identificar patrones recurrentes y áreas propensas a fallos. A través de técnicas de procesamiento de lenguaje natural (PLN), interpreta el lenguaje natural de los requerimientos y extrae información clave. Esta comprensión le permite reconocer zonas críticas de prueba y generar automáticamente casos que cubren una amplia variedad de escenarios, incluidos casos extremos, condiciones límite y comportamientos típicos del usuario, mejorando así la cobertura y la efectividad del proceso de validación.
- **Ejecución Mejorada de Pruebas.** Como parte de la preparación para la ejecución de pruebas, la AI Generativa configura automáticamente el entorno de pruebas, asegurando que refleje con precisión las condiciones del entorno de producción. Genera los datos necesarios para las pruebas, tales como cuentas de usuario, catálogos de productos y registros de transacciones. A continuación, ejecuta los casos de prueba en paralelo, lo que permite reducir de manera significativa el tiempo total de validación. Además, ajusta dinámicamente los parámetros de prueba en función de la retroalimentación en tiempo real y de los resultados obtenidos. Por ejemplo, GenAI puede ejecutar pruebas de inicio de sesión en múltiples navegadores y dispositivos de forma simultánea, garantizando así la compatibilidad y el rendimiento óptimo en diversas plataformas.

- Análisis de Resultados de Pruebas.** Durante la ejecución de las pruebas, la AI Generativa recopila de forma continua registros, métricas de rendimiento y mensajes de error. Analiza el comportamiento de la aplicación bajo diversas condiciones operativas, identificando anomalías y cuellos de botella que puedan comprometer su desempeño. A través de algoritmos de aprendizaje automático, detecta patrones en los resultados que evidencian posibles defectos y realiza análisis de causa raíz para identificar los problemas subyacentes. Además, evalúa el impacto de los defectos detectados tanto en la funcionalidad del sistema como en la experiencia del usuario, facilitando así una gestión proactiva y efectiva de la calidad.

A continuación se presentan algunas métricas asociadas al porcentaje de mejora de la producción mediante la incorporación de GenAI:

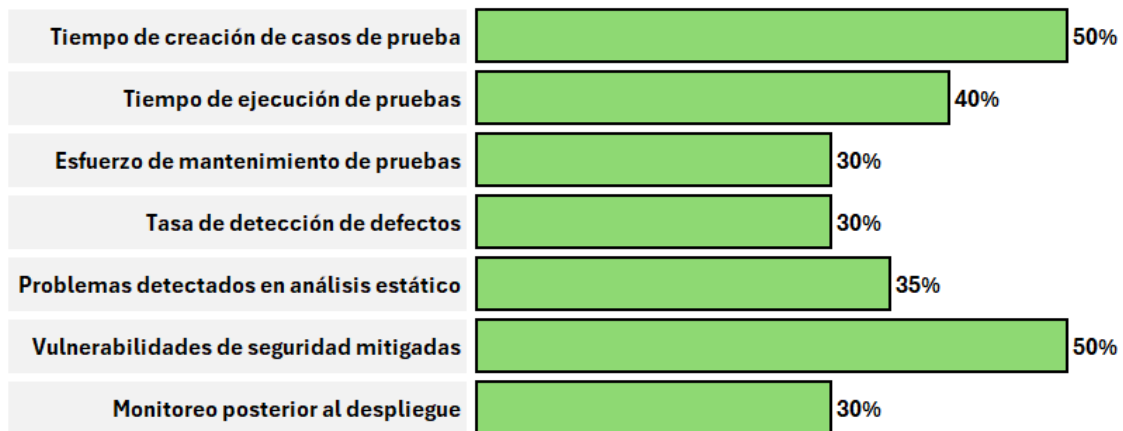


Ilustración 21: Porcentaje de mejora de la productividad con GenAI

Fuente: (Hall, 2024)

Considerando una gran plataforma de comercio electrónico que se basa en una compleja red de aplicaciones para brindar experiencias de usuario fluidas. Al integrar GenAI en sus procesos de prueba de software, la empresa logró mejoras significativas. El tiempo necesario para la creación de casos de prueba se redujo en un 50%, lo que garantizó una cobertura completa de diversos escenarios de usuario. El tiempo de ejecución de las pruebas se redujo en un 40%, lo que permitió lanzamientos más frecuentes y correcciones de errores más rápidas (Hall, 2024).

La mayor relevancia y precisión de los casos de prueba mediante el aprendizaje continuo redujo los esfuerzos de mantenimiento en un 30%. Las pruebas enfocadas en áreas de alto riesgo mejoraron las tasas de detección de defectos en un 25%.

Durante la fase de análisis estático, se detectaron y resolvieron un 35% más de problemas, lo que resultó en una base de código más estable. Las vulnerabilidades de seguridad se identificaron y mitigaron más rápidamente, lo que mejoró la seguridad general de la plataforma. La monitorización posterior a la implementación mejorada permitió la identificación y resolución inmediata de problemas de rendimiento.



Ilustración 22: Ganancias GenAI integrado en el ciclo de vida de las pruebas

Fuente: (Hall, 2024)

Si nos vamos a las instancias más prematuras dentro del ciclo de vida de desarrollo de software, una de las actividades más críticas para el éxito del proyecto es la ingeniería de requerimientos (RE), y la capacidad de generar un documento de especificación con las necesidades de los usuarios lo más robusto posible y libre de errores. El software moderno debe cumplir con diversos requerimiento (funcionales y no funcionales), garantizando al mismo tiempo la confiabilidad, la seguridad y la facilidad de mantenimiento.

A pesar de los avances significativos, los proyectos complejos suelen presentar problemas como retrasos, sobrecostos y defectos del sistema. En un estudio realizado por (Lederer, 1995) se demostró mediante estudios empíricos que las frecuentes solicitudes de cambio por parte de los usuarios y su falta de comprensión de los requerimientos son los principales factores que contribuyen a los sobrecostos de los proyectos. Además, se destaca que solo el 31 % de los proyectos de software se completan a tiempo y dentro del presupuesto, y tan solo el 46 % genera retornos de alto valor. Estos hallazgos subrayan la necesidad crucial de optimizar la fase de ingeniería de requerimientos para mejorar la tasa de éxito de los proyectos de software y reducir costos, y es en esta arista que la inteligencia artificial generativa se convierte en un poderoso aliado.

La inteligencia artificial puede ser utilizada para inspeccionar los requerimientos de un proyecto y evaluar su viabilidad, impacto y prioridad, lo que garantiza que el equipo se concentre en los aspectos más importantes del proyecto, haciendo uso de los algoritmos de procesamiento de datos que pueden revisar y reconocer patrones de grandes volúmenes de documentos, ahorrando tiempo, reduciendo la

carga de trabajo manual y minimizando errores humanos. Las herramientas potenciadas por inteligencia artificial pueden ofrecer datos sobre las interrelaciones entre los requerimientos, realizar análisis de impacto y contribuir a la identificación temprana de conflictos u obstáculos en las etapas iniciales del proyecto (Gaona, 2024).

Al aprovechar el potencial de los LLM, las actividades clave de la ingeniería de requerimientos, como la obtención, el análisis, la especificación, la validación y la gestión de requisitos, pueden automatizarse o ampliarse. Esta posibilidad se ilustra claramente en la ilustración 22, donde se muestra que cada componente del proceso de RE se beneficia de las capacidades específicas de GenAI. Por ejemplo, GenAI puede generar requisitos automáticamente basándose en las aportaciones de las partes interesadas, como lo demuestra la conexión entre el PLN y el proceso de obtención. La capacidad de detectar inconsistencias o ambigüedades en los documentos de requisitos se representa mediante el vínculo entre el “reconocimiento de patrones” y la “fase de análisis”. Además, GenAI puede ayudar a crear historias de usuario, casos de uso y otros artefactos de requerimientos, reduciendo sustancialmente el tiempo y el esfuerzo que requieren los profesionales de esta área (Cheng, 2024).

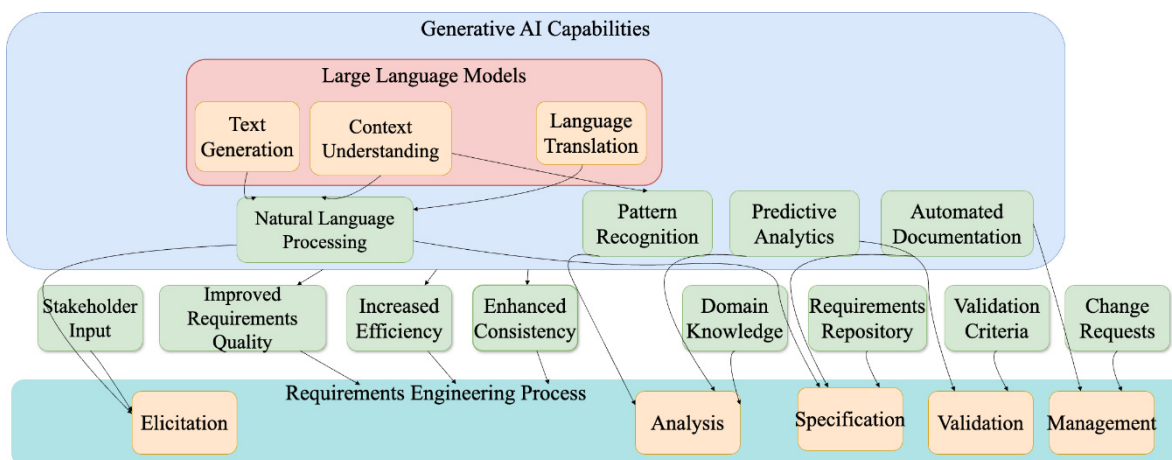


Ilustración 23: Integración de IA Generativa en la Ingeniería de Requerimientos

Fuente: (Cheng, 2024)

A medida que los sistemas de software se vuelven cada vez más complejos e interconectados, la gestión y el seguimiento de los requisitos a lo largo del ciclo de vida del desarrollo se convierte en un desafío clave. GenAI tiene el potencial de mejorar sustancialmente la trazabilidad de requerimientos, el análisis del impacto de los cambios y la verificación de la consistencia, lo que ayuda a mantener la claridad y la cohesión en sistemas en evolución. Es precisamente a través de esta

tecnología que se busca reducir drásticamente el esfuerzo humano en estas tareas, a la vez que se mejora la precisión y se minimizan los errores.

Con el aprendizaje profundo y el procesamiento del lenguaje natural (PLN), la IA puede ayudar a comprender la semántica de un lenguaje. Puede comprender procesos y, al usar conceptos semánticos, puede aplicarse de muchas maneras, por ejemplo: en el contexto de la calidad de los requerimientos. La IA puede aplicar fácilmente reglas y verificaciones y, por lo tanto, permitir enfocar el tiempo de revisión en problemas importantes, resolviendo y automatizando muchos problemas actuales relacionados con requerimientos deficientes. Si buscamos métricas que nos permitan tener una visión general de los beneficios de incorporar la IA en la inspección de requerimientos podemos referenciar el estudio de (Monsalve, 2015), donde se menciona que el número de páginas que en promedio un inspector puede revisar por hora de trabajo fluctúa entre 2 páginas por hora (pph), hasta 15 páginas por hora.

La métrica clave para medir la eficiencia humana en la inspección de requerimientos se conoce como tasa de revisión (rate of review), y según lo que plantean diversos autores y fuentes de estudio se puede considerar una tasa media de 14 páginas/hora. En contraposición, cuando las tareas de inspección se delegan en herramientas de inteligencia artificial se evidencian mejoras concretas relacionadas a tiempos de revisión, tasa de detección de defectos y tasa de defectos no detectados (Böckeler, 2024).

En un estudio llevado a cabo por (Mishra, 2024) se analizó la efectividad de los modelos de inteligencia artificial aplicado a pruebas de software en relación con efectividad en términos de cobertura, precisión, tiempo de ejecución y tasa de falsos positivos. Para esto se revisaron múltiples enfoques de pruebas basados en IA con métodos tradicionales y se evaluaron sus fortalezas y limitaciones. Los resultados experimentales indican que la IA Generativa mejora significativamente la eficiencia de las pruebas. Los resultados de este análisis destacaron que los modelos de IA Generativa (GPT-4, Codex y CodeT5) superaron a los métodos tradicionales en cobertura, precisión y velocidad de ejecución, a la vez que redujeron el esfuerzo manual en la generación de casos de prueba. Sin embargo, retos como los falsos positivos y la dependencia de los datos de entrenamiento de la IA siguen siendo consideraciones clave. A continuación se plantearán las principales conclusiones de este estudio.

- **Cobertura de pruebas.** Los resultados mostraron que GPT-4 logró la mayor cobertura de pruebas, con un 92%, seguido de Codex con un 88% y CodeT5 con un 85%. En comparación, herramientas de prueba tradicionales como Selenium, JUnit y TestNG tuvieron una cobertura de pruebas de entre el 78% y el 80%. La mayor cobertura lograda por la IA Generativa se debe a su capacidad para generar dinámicamente diversos casos de prueba,

incluyendo casos extremos que a menudo se pasan por alto en las pruebas programadas manualmente.

- Precisión de las pruebas.** Los modelos de IA demostraron una alta precisión: GPT-4 alcanzó el 95%, Codex el 93% y CodeT5 el 90%. En cambio, las herramientas de prueba tradicionales tuvieron un rango de precisión del 83% al 85%. La mayor precisión de las pruebas generadas por IA sugiere que los modelos de IA pueden identificar eficazmente problemas funcionales en el software, a la vez que minimizan los errores manuales asociados con las pruebas tradicionales. Sin embargo, las alucinaciones ocasionales en los casos de prueba generados por IA siguen siendo un problema, lo que puede afectar la fiabilidad de los resultados.
- Eficiencia en tiempos de ejecución.** El tiempo de ejecución es un factor crítico para determinar la eficiencia de las herramientas de pruebas automatizadas. Los resultados indican que los modelos de IA Generativa redujeron significativamente el tiempo de ejecución de las pruebas: GPT-4 completó los casos de prueba en 2,3 segundos, Codex en 2,8 segundos y CodeT5 en 3,1 segundos. Los métodos tradicionales, como Selenium, tardaron más, con un promedio de 5,0 segundos por ejecución. Esta mejora se debe a que los modelos de IA generan casos de prueba optimizados que se ejecutan eficientemente, lo que reduce la necesidad de flujos de prueba programados manualmente.
- Análisis de falsos positivos.** Si bien la IA generativa tuvo un buen rendimiento en la mayoría de las áreas, persistieron los falsos positivos: GPT-4 tuvo una tasa del 3%, Codex del 4% y CodeT5 del 5%. Las herramientas de prueba tradicionales, como Selenium, presentaron tasas de falsos positivos más altas, con un promedio del 7-8%. Las tasas más bajas de falsos positivos en los modelos de IA sugieren que las pruebas generadas por IA pueden proporcionar una detección de defectos más precisa, pero aún se requiere cierto nivel de validación humana para garantizar la fiabilidad.

Modelo/Herramienta	Cobertura de Pruebas (%)	Precisión (%)	Tiempo de Ejecución (s)	Tasa de Falsos Positivos (%)
GPT-4	92%	95%	2.3	3%
Codex	88%	93%	2.8	4%
CodeT5	85%	90%	3.1	5%
Selenium	80%	85%	5.0	7%
JUnit	78%	83%	4.7	8%
TestNG	79%	84%	4.5	7%

Ilustración 24: Análisis comparativo de rendimiento

Fuente: (Mishra, 2024)

7.5 Modelo Tecnológico Propuesto

La solución tecnológica propuesta se fundamenta en un modelo de generación aumentada por recuperación (RAG), el cual combina las capacidades de recuperación de información relevante desde fuentes de conocimiento con el poder generativo de los modelos de lenguaje avanzados. Esta arquitectura permite enriquecer el proceso de aseguramiento de calidad de software mediante respuestas más precisas y contextualizadas, adaptadas dinámicamente a los requisitos y escenarios específicos del proyecto.

A través de la integración de GenAI, se optimiza cada una de las etapas críticas del ciclo de pruebas: en el análisis de requerimientos, la GenAI facilita la comprensión semántica profunda y la identificación automática de ambigüedades, en el diseño de casos de prueba, permite la generación automatizada y exhaustiva de escenarios de prueba basados en patrones detectados y mejores prácticas, durante la ejecución, apoya con scripts dinámicos y adaptativos, y en la fase de generación de informes, produce documentación clara, detallada y orientada a la toma de decisiones.

7.5.1 Retrieval-Augmented Generation (RAG)

La generación aumentada por recuperación es una técnica avanzada en el ámbito de la inteligencia artificial que integra la recuperación de información con la generación de respuestas mediante modelos de lenguaje (LLM). Este enfoque innovador permite que los LLM no se limiten únicamente a su conocimiento interno, sino que también accedan a fuentes de datos externas para ofrecer respuestas más precisas, completas y actualizadas. A diferencia de los LLM estándar, que se basan únicamente en sus datos de entrenamiento, los sistemas de IA RAG tienen la capacidad de acceder a bases de conocimiento externas, extrayendo la información más actualizada y relevante para proporcionar respuestas precisas y contextualizadas. Esta capacidad de conectarse con datos en tiempo real hace que RAG sea especialmente valioso para empresas y organizaciones (Richards, 2024).

Los LLM se entrenan con grandes volúmenes de datos y usan miles de millones de parámetros para generar resultados originales en tareas como responder preguntas, traducir idiomas y completar frases. RAG extiende las ya poderosas capacidades de estos modelos a dominios específicos o a la base de conocimientos interna de una organización en particular, todo ello sin la necesidad de volver a entrenar el modelo. Dentro de los principales beneficios de usar un modelo RAG destaca:

- Mejora de la precisión de las respuestas. Al utilizar información externa, RAG puede mejorar significativamente la precisión de las respuestas generadas por los LLM. Esto es especialmente útil en contextos donde se requiere información específica y actualizada.
- Reducción de alucinaciones del modelo. Las alucinaciones ocurren cuando un LLM genera respuestas incorrectas o irrelevantes. Al utilizar datos externos, RAG ayuda a reducir la incidencia de estas alucinaciones, proporcionando respuestas más fiables y precisas.
- Obtención de información más relevante y actualizada. Al integrar datos externos, RAG garantiza que las respuestas generadas por los LLM estén basadas en la información más reciente y relevante disponible, mejorando así la calidad y utilidad de las respuestas.

Para aprovechar al máximo el poder de la IA RAG, es fundamental comprender sus componentes principales. Pensemos en RAG como un sistema sofisticado con dos partes principales que trabajan en armonía: el componente de recuperación y el componente de generación (Richards, 2024).

- El componente de recuperación actúa como un bibliotecario altamente cualificado. Al plantear una pregunta o solicitud a un sistema RAG, este componente entra en acción, examinando extensas bibliotecas de datos, documentos o bases de conocimiento para identificar la información más relevante. No se trata solo de la coincidencia de palabras clave; se emplean algoritmos y técnicas avanzadas, como la búsqueda semántica, para comprender el contexto y la intención de la solicitud, garantizando así la pertinencia de la información recuperada.
- Una vez que el componente de recuperación ha recopilado el conocimiento necesario, el componente de generación cobra protagonismo. Aquí es donde entra en juego la magia de los grandes modelos lingüísticos (LLM). El LLM procesa la información recuperada, integrándola con su propia destreza lingüística para generar una respuesta coherente, informativa y contextualmente relevante.

Tal como hemos revisado previamente, la clave del uso de un modelo RAG es mejorar el rendimiento de los LLM mediante la incorporación de conocimiento externo al proceso de generación. Para entender de mejor manera la anatomía de este tipo de solución podemos extender de forma sencilla cinco componentes clave (Rautenstrauch, 2024):

1. Base de Conocimientos Personalizada

Una colección de información relevante y actualizada que sirve de base. Puede ser una base de datos, un conjunto de documentos o una combinación de ambos.

2. Chunking

La fragmentación es el proceso de dividir un texto de entrada grande en partes más pequeñas. Esto garantiza que el texto se ajuste al tamaño de entrada del modelo de incrustación y mejora la eficacia de la recuperación.

3. Embeddings y modelo de incrustación

Técnica para representar datos de texto como vectores numéricos que pueden introducirse en modelos de aprendizaje automático. El modelo de incrustación se encarga de convertir el texto en estos vectores.

4. Bases de datos de vectores

Una colección de representaciones vectoriales pre calculadas de datos de texto para una rápida recuperación y búsqueda de similitudes, con capacidades como operaciones CRUD, filtrado de metadatos y escalado horizontal.

5. Interfaz de chat de usuario

Una interfaz fácil de usar que permite a los usuarios interactuar con el sistema RAG, proporcionando una consulta de entrada y recibiendo una salida. La consulta se convierte en una incrustación que se utiliza para recuperar el contexto pertinente de la base de datos de vectores.

RAG: Retrieval Augmented Generation

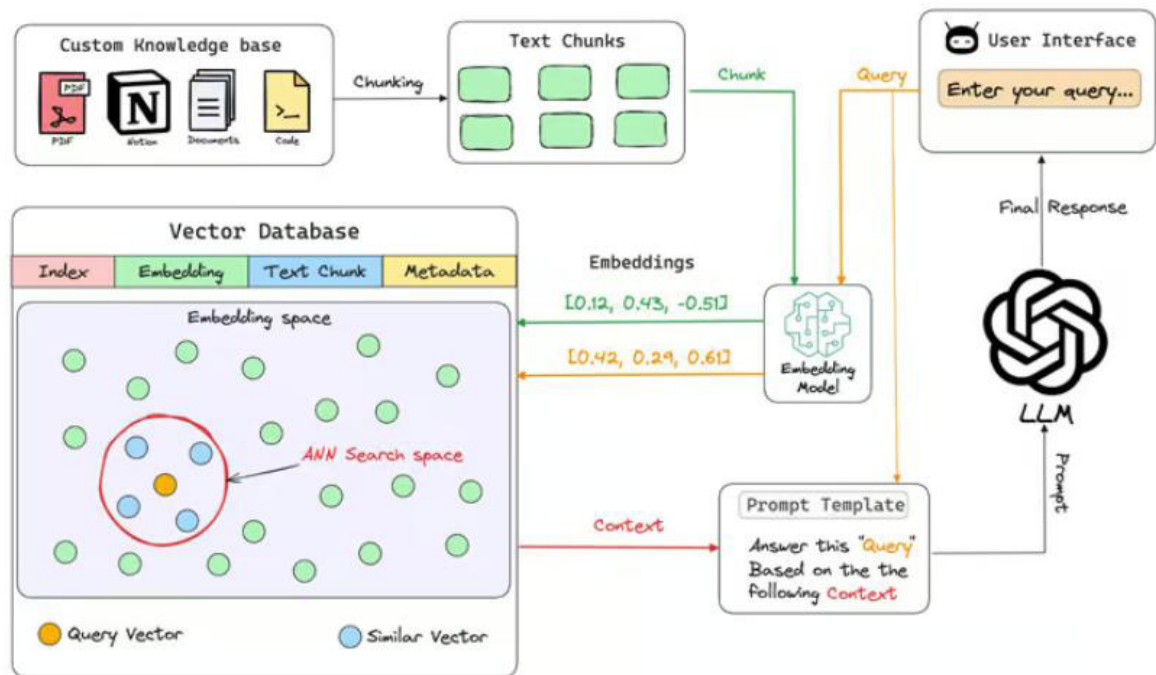


Ilustración 25: Retrieval Augmented Generation (RAG)

Fuente: (Rautenstrauch, 2024)

RAG tiene un rol clave en el contexto de las pruebas de software. Los métodos de prueba tradicionales se basan en gran medida en la creación de scripts y el diseño de pruebas manuales, que son propensos a errores y requieren mucho tiempo. Este modelo mejora esto generando casos de prueba automatizados, reduciendo el esfuerzo manual y agilizando los procesos. Dentro de los beneficios directos dentro del proceso de pruebas se pueden mencionar (Kolhe, 2024):

- **Generación eficiente de casos de prueba.** Con RAG, los analistas de prueba pueden generar casos de prueba exhaustivos y variados. Al extraer información de documentos de requerimientos, casos de prueba anteriores e informes de errores históricos, el sistema genera de forma inteligente escenarios de prueba completos, garantizando que no se pasen por alto casos extremos y condiciones de borde.
- **Identificación y análisis de errores.** Los sistemas RAG pueden ayudar a identificar posibles errores al comparar los problemas reportados con una amplia base de datos de problemas conocidos y sus soluciones. Esto no solo acelera el proceso de depuración, sino que también proporciona información más detallada sobre la causa raíz de los defectos.

- **Informes de defectos inteligentes.** El seguimiento de defectos puede resultar engorroso, con informes duplicados y problemas críticos que se pasan por alto. RAG puede ayudar a los probadores recuperando defectos similares de datos anteriores, comparándolos y generando informes mejorados. Esto no solo reduce la redundancia, sino que también mejora la trazabilidad de los defectos.
- **Documentación mejorada.** Una de las tareas más descuidadas en las pruebas de software es el mantenimiento de la documentación. RAG facilita la creación de documentación de casos de prueba en tiempo real, lo que permite a los evaluadores realizar un seguimiento de los resultados y procesos de las pruebas sin intervención manual. Gracias a su capacidad de recuperación, RAG genera informes detallados y precisos.

7.5.2 Descripción de la Solución

Tal como se profundizó en la sección anterior, la solución propuesta está fundamentada en la incorporación de un modelo RAG (Retrieval-Augmented Generation) como parte integral del proceso de pruebas de software. Esto representa una solución innovadora y altamente efectiva para optimizar cada una de las etapas críticas del aseguramiento de calidad. En primer lugar, en el análisis de requerimientos, RAG permitirá una comprensión contextual profunda al combinar recuperación de información específica del dominio (como especificaciones técnicas, historiales de proyectos, normativas y estándares propios de la industria) con generación de lenguaje natural, facilitando así la identificación de inconsistencias, ambigüedades o lagunas en los requerimientos desde etapas tempranas. En cuanto al diseño y creación de matrices de prueba, el modelo puede generar automáticamente casos y escenarios de prueba bien estructurados, alineados con los requerimientos extraídos y contextualizados, lo que reduce el error humano y acelera el proceso de diseño.

Durante la ejecución de pruebas, RAG actuará como un asistente inteligente que guíe dinámicamente la ejecución, sugiriendo ciclos de prueba basadas en resultados previos, riesgos detectados o cambios recientes en el software, habilitando así estrategias adaptativas y más eficaces en función de los cambios que se produzcan durante el proyecto. Finalmente, en la generación de informes de resultados, el modelo será capaz de producir documentación detallada y comprensible, con explicaciones generadas en lenguaje natural que contextualizan los hallazgos, facilitando la toma de decisiones tanto técnicas como gerenciales. En conjunto, el uso de RAG permitirá una sinergia entre la inteligencia artificial generativa y el conocimiento contextual, transformando el proceso de testing en un flujo más inteligente, automatizado y centrado en la calidad.

De manera general, la solución basada en GenAI y en un modelo tipo RAG, tendrá un impacto en cuatro grandes áreas constituyentes del proceso de certificación (Barnes, 2024): análisis de requerimientos, creación de matrices de pruebas, ejecución de pruebas y generación de informes de resultados. En esta línea, la IA nos permitirá agilizar el proceso de análisis y revisión de requerimientos, en base a las buenas prácticas de la ingeniería de requerimientos, estándares propios de la organización y criterios anexos que quieran ser incorporados, los cuales estarán cargados en la base de conocimientos personalizada del modelo RAG. Por otra parte, ya contándose con una especificación de requerimientos robusta y validada, la IA tomará control sobre el análisis de la documentación y posterior creación de la suite de casos de prueba asociados. Para esto, el motor de la inteligencia artificial hará uso de su conocimiento experto en pruebas de software, además de los documentos específicos cargados en la base de conocimientos para usar como referencia, tales como: documentos ISTQB®, OWASP Testing Guide, ISO/IEC/IEEE 29119, entre muchos otros.

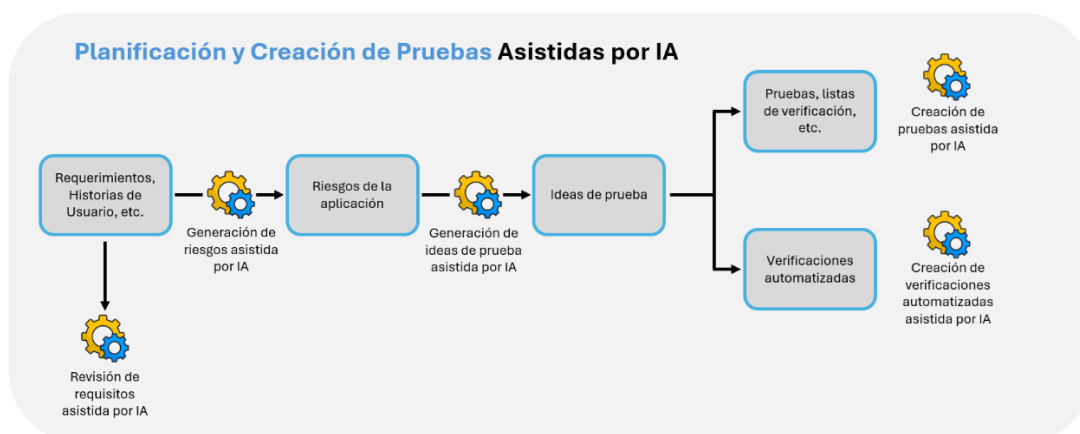


Ilustración 26: Revisión de Requerimientos y Creación de Matrices de Pruebas

Fuente: (Barnes, 2024)

En lo que respecta a la ejecución de pruebas, la IA puede ayudar a identificar el impacto en la certificación de cualquier cambio en el alcance del proyecto (estrategia adaptativa). En esta línea, se pueden identificar qué casos de prueba fueron impactados y se deben editar (o incluso si es necesario incorporar nuevos casos o eliminar los existentes), priorizar las pruebas de regresión según la relevancia de los requerimientos, hacer los ajustes necesarios en los casos que ya se encuentran automatizados (técnicas de auto reparación), etc. Adicionalmente, los resultados obtenidos en el ciclo en curso pueden ser contrastados en línea con los antecedentes históricos (de proyectos o ciclos anteriores) para evaluar tendencias y patrones de comportamiento.

Las pruebas basadas en IA permiten ejecutar pruebas de forma inteligente, priorizándolas según el riesgo, el impacto y otros factores. Esto ayuda a garantizar que las pruebas más críticas se ejecuten primero, reduciendo el riesgo de defectos y mejorando la eficiencia de las pruebas.

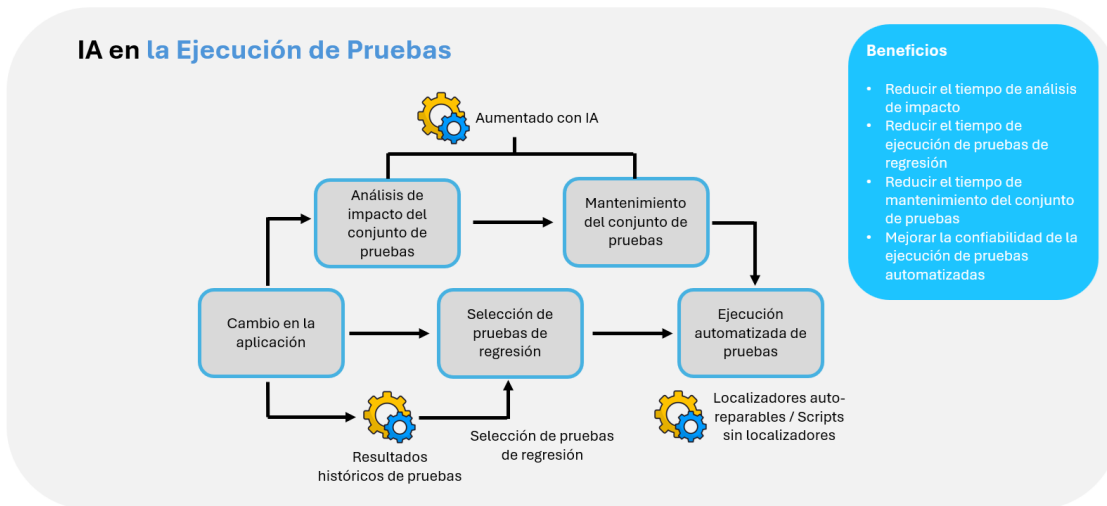


Ilustración 27: Ejecución de Pruebas

Fuente: (Barnes, 2024)

En lo que respecta al análisis de resultados la IA es capaz de procesar grandes cantidades de información, tales como: resultados de herramientas, pantallas con métricas, indicadores, resultados de pruebas y evidencias, identificando niveles de cumplimiento de los criterios de aceptación, categorizando los defectos según tipo y criticidad y generando informes de resultados adaptados según tipo de audiencia. Incluso, en función de ciertos resultados particulares y los registros históricos de comportamiento, la IA es capaz de predecir la aparición de defectos que quizás no son evidentes que podrían materializarse en instancias tardías del proceso.

Resulta clave entender que la inteligencia artificial generativa puede ser una poderosa herramienta si se utiliza alineada con las buenas prácticas de pruebas de software y bajo la supervisión de analistas expertos. En este contexto resulta clave preparar el modelo con una base de conocimientos sólida, tanto en lo que respecta al testing, como los antecedentes propios de la organización sobre la cual se estará trabajando. No todas las organizaciones definen los mismos niveles de calidad y criterios de aceptación para sus productos de software y tampoco se encuentran supeditados a las mismas regulaciones y estándares globales.

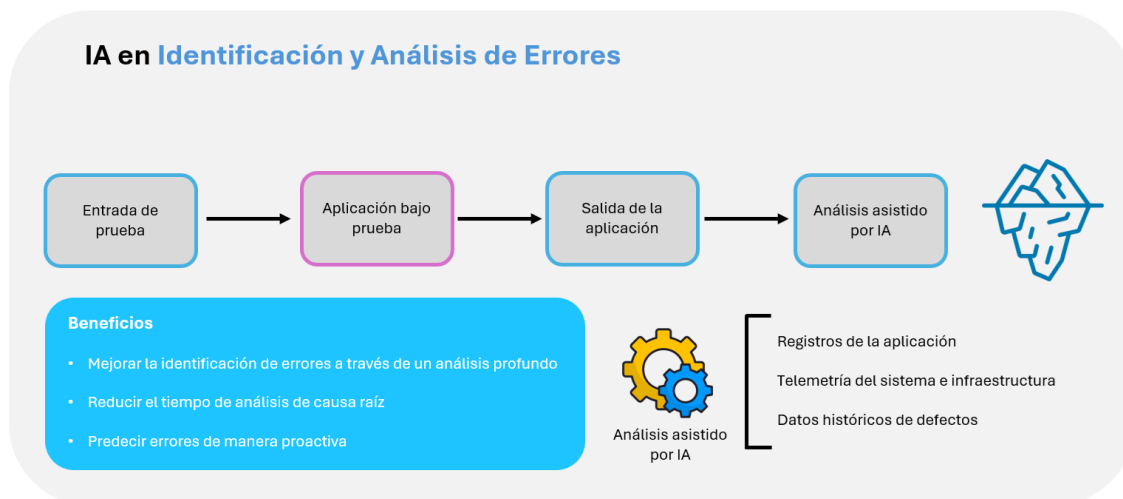


Ilustración 28: Detección y Análisis de Defectos

Fuente: (Barnes, 2024)

Uno de los elementos que será de ayuda para apalancar el proceso de implementación de GenAI como parte de las pruebas de software es la capacitación de los analistas, y de manera particular se debiese definir como punto de partida una formación sólida en ingeniería de prompt (prompt engineering), diseño y optimización de las entradas (prompts) que se utilizan para interactuar con modelos de lenguaje grandes (LLMs) y sistemas de IA generativa, ya que estos modelos (RAG o LLMs), dependen fuertemente de la claridad, precisión y estructura de la entrada para generar respuestas útiles, coherentes y alineadas con los objetivos del testing.

Además, la ingeniería de prompt potencia la interacción contextual con modelos GenAI, permitiendo que los analistas formulen preguntas técnicas, generen escenarios negativos o de borde, identifiquen riesgos de seguridad (alineados con las principales guías de la industria, por ejemplo: OWASP), y adapten dinámicamente sus estrategias. Esta habilidad no solo amplificará el valor del analista como puente entre la inteligencia humana y la IA Generativa, sino que también maximizará el retorno de inversión en este tipo de tecnologías, asegurando una implementación más precisa, confiable y efectiva dentro del ciclo de vida del aseguramiento de calidad del software.

Por último, el Plan de Pruebas seguirá siendo el documento estratégico que definirá el marco de trabajo de todo el proceso de certificación. Sin embargo, con la incorporación de la GenAI se puede generar de forma automática en base a los antecedentes disponibles y las directrices del cliente, cargar en nuestro modelo y actualizarse dinámicamente en función de los diferentes cambios que vaya

sufriendo el proyecto, de tal manera de disponer siempre de un documento actualizado y en sintonía con la realidad del proceso de certificación. Un ejemplo concreto sería que si durante las pruebas se incorpora un nuevo requerimiento que requiera algún tipo de particular de prueba (usabilidad, accesibilidad, seguridad, etc.), de manera automática la IA generará una nueva versión del plan de pruebas con el cambio de alcance para que sea revisado y validado por la contraparte del negocio y el resto de los stakeholders.

La inteligencia artificial tiene el potencial de transformar significativamente la forma en que abordamos las pruebas de software, pero no sustituye el juicio ni el valor que aportamos como profesionales. Es necesario usarla estratégicamente para automatizar tareas repetitivas o demandantes, como lo son la generación de escenarios, datos de prueba o el análisis de grandes volúmenes de resultados, pero es clave mantener siempre una postura crítica. Es necesario cuestionar los hallazgos, validar los resultados y asegurarse de tener el control sobre cómo y cuándo se aplica la IA en el proceso. Es fundamental contar con una comprensión básica de los conceptos que sustentan la inteligencia artificial, cuanto mayor sea tu conocimiento, mejor capacitado se estará para formular las preguntas adecuadas y analizar críticamente las respuestas que ofrece el modelo. El hecho de que una herramienta de IA proporcione una respuesta convincente no garantiza que sea correcta. Por ello, es clave aplicar criterio profesional y experiencia para evaluarla con rigurosidad.

7.6 Desafíos en el uso de GenAI

A pesar del potencial transformador que ofrece la Inteligencia Artificial Generativa (GenAI) en la optimización del aseguramiento de calidad en el desarrollo de software, su implementación conlleva una serie de desafíos significativos que deben ser cuidadosamente considerados. Estos desafíos no solo se limitan a aspectos tecnológicos, como la confiabilidad de los modelos generativos, sino que también abarcan cuestiones éticas relativas al uso responsable de la IA, los sesgos en los datos de entrenamiento y la confidencialidad de la información. Además, la dimensión humana juega un papel fundamental, ya que la adopción efectiva de estas tecnologías requiere una transformación cultural dentro de los equipos de calidad, acompañada de una sólida estrategia de capacitación y reentrenamiento de personal. En este contexto, es imprescindible analizar las barreras clave que pueden obstaculizar el despliegue exitoso de soluciones basadas en GenAI, con el fin de anticiparlas y mitigar sus impactos desde las etapas tempranas del proyecto.

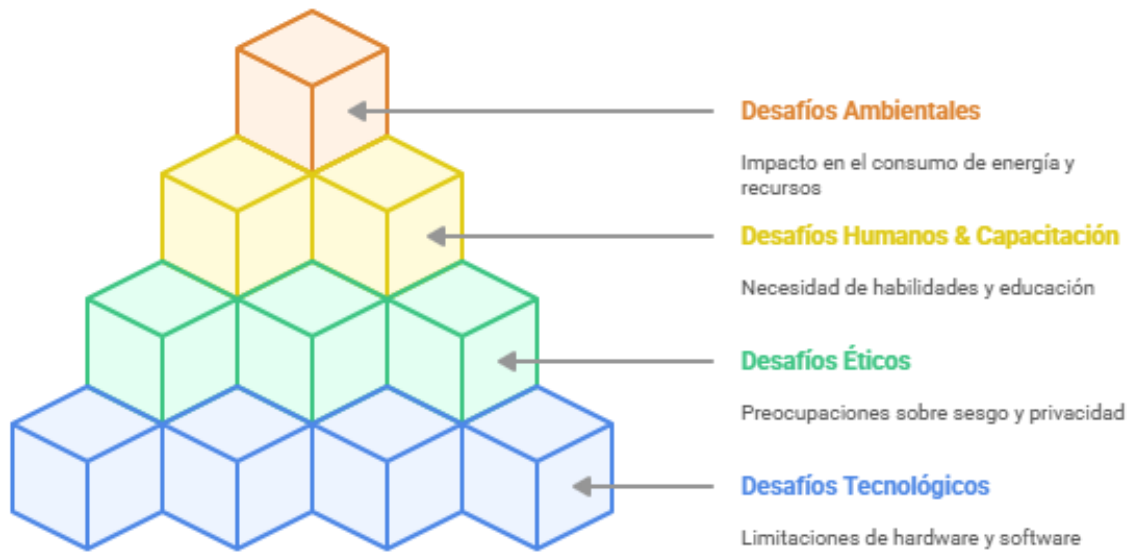


Ilustración 29: Desafíos asociados al uso de IA Generativa

Fuente: Elaboración propia

7.6.1 Desafíos Tecnológicos

En el contexto de los desafíos tecnológicos que deben ser tenidos en consideración al momento de implementar el modelo propuesto en la fase previa, abordaremos los que tienen un mayor impacto transversal al uso de IA:

Alucinaciones

Las alucinaciones de la IA son respuestas incorrectas, inconsistentes o engañosas generadas por los modelos, que no se corresponden con la realidad o los datos disponibles. Estas fallas pueden originarse por múltiples factores, tales como: insuficiencia o baja calidad de los datos de entrenamiento, la presencia de sesgos en los datos, o limitaciones inherentes a la arquitectura del modelo y sus mecanismos de inferencia. Este fenómeno representa un riesgo significativo, especialmente en contextos donde las salidas generadas por la IA influyen en la toma de decisiones críticas, ya que puede conducir a conclusiones erróneas o a acciones inapropiadas si no se detectan y controlan adecuadamente.

El término “antropomorfismo” se expresa en la definición y explicación de la alucinación dentro de los modelos de IA Generativa. Generalmente, la alucinación se refiere a la capacidad de la IA para generar contenido que se desvía de los resultados convencionales o esperados. Sin embargo, la más arriesgada es cuando

el modelo GenAI genera un resultado que coincide perfectamente con lo esperado, pero carece de validación basada en hechos verificados. Este escenario es más preocupante, ya que los usuarios podrían aceptar el contenido generado como preciso, propagando inadvertidamente información no verificada o falsa. El desafío radica en distinguir entre contenido probable, pero sin fundamento, y resultados basados en hechos validados para garantizar un uso responsable de la IA Generativa (NTT Data, 2024).

Las alucinaciones en la IA Generativa pueden manifestarse de diversas formas, según el tipo de modelo y la tarea específica para la que estén diseñadas. Estos son algunos tipos comunes de alucinaciones en la IA Generativa (NTT Data, 2024):

- **Alucinación Visual.** En los modelos de generación de imágenes, la alucinación visual puede implicar la creación de imágenes que representan objetos, escenas o patrones inexistentes. Estas alucinaciones pueden abarcar desde arte surrealista y abstracto hasta objetos o criaturas totalmente inventados.
- **Alucinación Textual.** Los modelos lingüísticos pueden alucinar textos generando oraciones o párrafos que contienen información ficticia o hacen afirmaciones falsas. Las alucinaciones textuales pueden implicar la invención de eventos, detalles o hechos sin fundamento.
- **Alucinación por Expansión de Contenido.** Este fenómeno ocurre cuando un modelo generativo produce más información de la presente en los datos de entrada. Por ejemplo, un modelo podría añadir detalles innecesarios a una imagen o generar narrativas extensas que van más allá de la información proporcionada.
- **Alucinación Inferencial.** En las tareas de procesamiento del lenguaje natural, la alucinación inferencial puede llevar a suposiciones o inferencias incorrectas. Los Modelos de Lenguaje Grande (LLM) pueden extraer conclusiones infundadas de los datos de entrada, lo que genera respuestas que juzgan erróneamente o tergiversan el contexto.
- **Alucinación de Sesgos.** La alucinación de sesgos se refiere a la generación de contenido que refleja o amplifica sesgos ya presentes en los datos de entrenamiento. Esto puede generar resultados que muestren estereotipos, discriminación o incluso puntos de vista poco éticos.
- **Alucinación Contextual.** Los modelos lingüísticos pueden sufrir alucinaciones contextuales, donde generan un texto que parece contextualmente relevante pero que en realidad es incorrecto o no representativo del contexto real.

Comportamiento No Determinista

Para comprender los desafíos del no determinismo, primero debemos comprender la naturaleza probabilística de la IA. A diferencia del software tradicional, la IA opera con base en probabilidades. En los LLM, el modelo genera resultados con una alta probabilidad de ser una respuesta adecuada a la solicitud de entrada. Por ejemplo: si la entrada es “manzana”, el modelo podría predecir “pastel”, “salsa” o incluso “iPhone” basándose en sus datos de entrenamiento. Dependiendo de parámetros como la temperatura, el modelo podría elegir cualquiera de estas palabras, no solo la de mayor probabilidad. Esta aleatoriedad inherente confiere a los modelos una apariencia de creatividad (Monnette, 2024).

Si bien esta capacidad creativa puede resultar extremadamente valiosa, también conlleva riesgos importantes si no se gestiona de forma adecuada. Al operar con funciones probabilísticas en lugar de deterministas, es fundamental estar preparados para enfrentar los desafíos inherentes a la incertidumbre y la variabilidad en los resultados generados:

- **Resultados Inconsistentes.** Uno de los principales desafíos es la variabilidad en las respuestas. Ante la misma solicitud de entrada, un LLM puede generar diferentes resultados cada vez que se consulta.
- **Falta de Reproducibilidad.** El no determinismo dificulta la reproducción de resultados específicos, lo que puede ser problemático cuando abordamos problemas de depuración, se realizan investigaciones científicas y/o se busca garantizar el cumplimiento en las industrias reguladas.
- **Preocupaciones sobre la Confiabilidad.** En aplicaciones que requieren alta precisión y consistencia, como los servicios financieros o la atención médica, la imprevisibilidad de los resultados de LLM puede generar preocupaciones sobre la confiabilidad.
- **Explicación y Transparencia.** La naturaleza de “caja negra” de los LLM, sumada a su comportamiento no determinista, dificulta la explicación de cómo se generan resultados específicos. Esta falta de transparencia puede ser problemática en escenarios que requieren procesos de toma de decisiones claros.

Recursos y Hardware

Los modelos de IA Generativa necesitan de enormes cantidades de datos para ser entrenados. Además, muchos de los modelos fundacionales que se utilizan hoy en día (como GPT-4, Mistral, Gemini o Claude) están basados en redes neuronales compuestos por miles de millones de parámetros. Esto implica que tanto para ser entrenados o utilizados en inferencia requieren de una cantidad significativa de recursos y hardware (Digitales, 2024).

En concreto, el hardware más adecuado para este tipo de modelos son las GPUs y las TPUs, que dada su capacidad de cálculo en paralelo permiten optimizar el rendimiento de estos modelos. Además de ser ampliamente empleadas para desarrollos de Inteligencia Artificial, las GPUs se emplean también en ámbitos como la renderización de gráficos para videojuegos o cine, para la computación de alto rendimiento (HPC), en criptominería, o en la edición de vídeo y foto. Aunque se trata de un mercado global, la producción de este hardware es realizado únicamente por unas pocas compañías donde NVIDIA y AMD son las claras líderes. Esto es debido a que su producción es un proceso complejo que además requiere de grandes inversiones en I+D para permitir a las empresas continuar siendo competitivas. Además, para su producción son necesarios ciertos recursos como son el silicio y ciertos compuestos metálicos, que forman la base de los semiconductores que son la base de las GPUs.

Debido a la alta demanda de GPUs, la escasez de semiconductores necesarios para su fabricación, y el impacto de ciertos efectos geopolíticos, tienen como consecuencia que en ocasiones se produzca escasez global de oferta, provocando un aumento de precios, y afectando a la industria en general y a los consumidores. Un ejemplo reciente de cómo los principales actores de la IA Generativa globales están invirtiendo en desarrollar su propio hardware y disponer de su propia infraestructura es el proyecto Stargate de Microsoft y OpenAI, con fecha de fin estimada para 2028, y que tiene como objetivo ser un centro de datos clave para el desarrollo de sistemas basados en Inteligencia Artificial, y cuya inversión se estima en 100 mil millones de dólares estadounidenses (Reuters, 2024).

Calidad de los Datos

GenAI no consiste únicamente en tener “más datos”, sino en combinar los datos públicos disponibles con tus datos propietarios y los conocimientos exclusivos de tu organización. Gestionar todos esos datos y ofrecer al mismo tiempo accesibilidad, seguridad y rendimiento puede resultar abrumador. El resultado es un dilema sobre los datos de GenAI: para hacerlo bien, se necesitan muchos datos de alta calidad para entrenar los modelos, pero sin la infraestructura de almacenamiento adecuada, todos esos datos pueden ser un obstáculo para el éxito (NetApp, 2024).

La precisión, relevancia y confiabilidad de los resultados generados por un LLM dependen directamente de la calidad de los datos de entrada (insumos) que recibe, tanto durante su entrenamiento como en el uso cotidiano. La calidad de los datos es un pilar fundamental en la implementación exitosa de modelos de IA Generativa, especialmente en contextos técnicos como el aseguramiento de calidad de software. Los modelos generativos aprenden patrones, estructuras y relaciones a partir de los datos con los que son entrenados; por tanto, cualquier sesgo, error, inconsistencia o insuficiencia en esos datos se refleja directamente en la calidad, confiabilidad y utilidad de las salidas generadas. Datos incompletos, desbalanceados o mal etiquetados pueden conducir a resultados inexactos, decisiones erróneas o comportamientos impredecibles del sistema.

7.6.2 Desafíos Éticos

El uso de IA Generativa en entornos críticos, como el aseguramiento de calidad de software, plantea una serie de implicaciones éticas que no debiesen ser ignoradas. A medida que estas tecnologías asumen un rol más activo en la toma de decisiones y en la generación de contenido, se vuelve esencial establecer marcos éticos que guíen su diseño, implementación y supervisión, con el fin de garantizar un uso responsable y alineado con los principios fundamentales de la ingeniería de software.

Uso Ético

La Inteligencia Artificial actual funciona exclusivamente en base a algoritmos, elementos que se expresan mediante fórmulas o conjuntos de fórmulas matemáticas. Esto la hace incapaz de modelar, y mucho menos comprender, elementos tan típicamente humanos como la ética, la moral, la cultura, las emociones, la empatía, el arte o la historia necesarios para que los algoritmos, sus resultados y aplicaciones respeten los derechos fundamentales de todo ser humano (Digitales, 2024).

A lo largo de los últimos años han surgido diversas iniciativas a lo largo del planeta que tratan precisamente de las cuestiones éticas y humanísticas relativas a estos algoritmos. Una de estas iniciativas es la Llamada de Roma por la Ética en la Inteligencia Artificial (ROMA CALL). Esta iniciativa trata de promover un sentido de responsabilidad compartida entre organismos públicos y privados de todo el mundo en un esfuerzo por “el desarrollo de una inteligencia artificial que sirva a cada persona y a la humanidad en su conjunto; que respete la dignidad de la persona humana, para que cada individuo pueda beneficiarse de los avances de la

tecnología; y que no tenga como único objetivo el mayor beneficio o la sustitución gradual de las personas en el lugar de trabajo”.

Por su parte, la UNESCO, dentro de los esfuerzos a nivel internacional para garantizar que la ciencia y la tecnología se desarrollen dentro de un marco ético, elaboró en noviembre de 2021 la primera norma mundial sobre la ética de la IA: la “Recomendación sobre la ética de la inteligencia artificial”, que fue adoptado por los 193 Estados miembro.

Sesgos

Un sesgo es una desviación o tendencia sistemática que afecta el comportamiento de los modelos de IA, llevándolos a tomar decisiones o hacer predicciones que favorecen ciertos resultados de manera injusta o distorsionada. Este sesgo puede surgir en varias etapas del desarrollo de IA y se refleja en el rendimiento del modelo, afectando la precisión y equidad de sus resultados.

Si los sistemas de IA Generativa aprenden a partir de grandes conjuntos de datos y estos contienen sesgos, representación no balanceada de ciertos grupos demográficos, prejuicios culturales, o estereotipos, el sistema aprenderá y perpetuará estos sesgos. La “caja negra” de la IA complica la interpretabilidad y explicabilidad en los procesos de toma de decisiones automatizados, por lo que la falta de mecanismos de supervisión adecuados puede permitir que la discriminación y los sesgos pasen inadvertidos y sin corregirse. Para enfrentarse a estos desafíos, es crucial adoptar prácticas de diseño y revisión ética que incluyan la evaluación y mitigación de sesgos desde las etapas iniciales del desarrollo (García, 2024).

A grandes rasgos podemos identificar 3 tipos de sesgos (Ferrara, 2023):

- **Sesgo de Datos.** Ocurre cuando los datos utilizados para entrenar modelos de aprendizaje automático no son representativos o están incompletos, lo que genera resultados sesgados. Esto puede suceder cuando los datos se recopilan de fuentes sesgadas, o cuando están incompletos, les falta información importante o contienen errores.
- **Sesgo Algorítmico.** Ocurre cuando los algoritmos utilizados en los modelos de aprendizaje automático tienen sesgos inherentes que se reflejan en sus resultados. Esto puede suceder cuando los algoritmos se basan en suposiciones sesgadas o cuando utilizan criterios sesgados para tomar decisiones.
- **Sesgo del Usuario.** Ocurre cuando las personas que utilizan sistemas de IA introducen sus propios sesgos o prejuicios en el sistema, consciente o

inconscientemente. Esto puede suceder cuando los usuarios proporcionan datos de entrenamiento sesgados o cuando interactúan con el sistema de maneras que reflejan sus propios sesgos.

El sesgo del mundo real se introduce en los sistemas GenAI y se devuelve al mundo real cuando los sistemas de IA crean nuevos resultados basados en él. Desde el mundo real, reintroducen los datos de entrenamiento y el proceso se reanuda. El riesgo de sesgo es que GenAI lo amplifique y perpetúe estos estereotipos a gran escala, ampliando las brechas existentes en equidad, representación e igualdad.

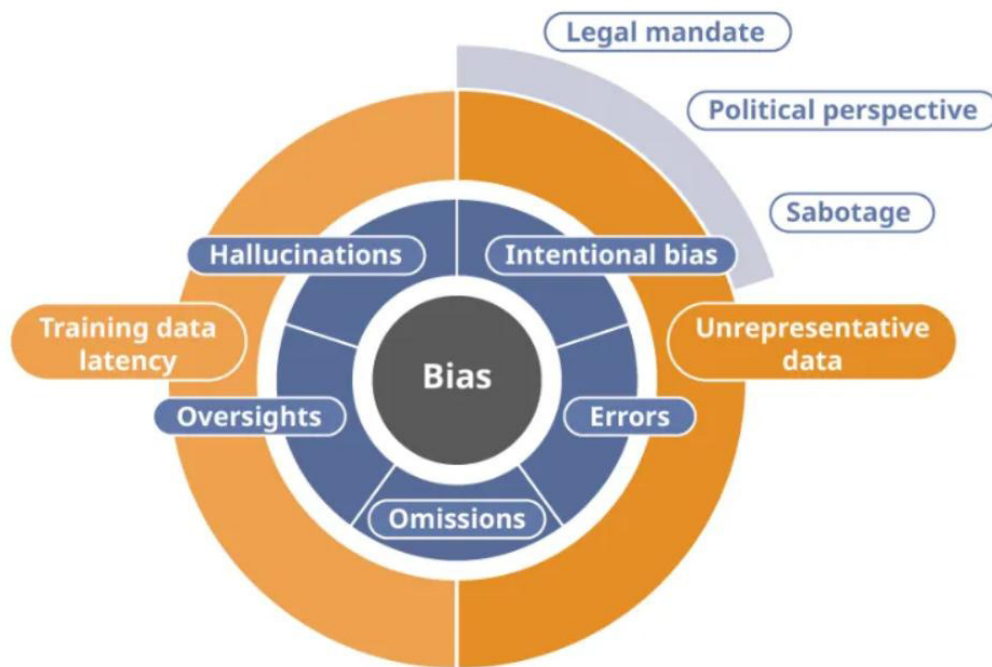


Ilustración 30: Sesgo accidental e intencional en algoritmos GenAI

Fuente: (Yi, 2023)

Privacidad & Seguridad

La experimentación con IA abarca un espectro que va desde el uso de herramientas y servicios de IA predefinidos hasta la creación de soluciones de IA personalizadas desde cero. Si bien es cierto algunas organizaciones pueden avanzar hacia la creación de sus propios modelos y aplicaciones de IA, muchas se limitarán a consumir herramientas de IA de terceros. En estos casos, las herramientas de IA de terceros presentan nuevos riesgos, porque las organizaciones solo tienen controles directos limitados sobre sus configuraciones de seguridad y privacidad.

Un ejemplo de esto sería cuando los empleados utilizan aplicaciones web públicas como ChatGPT. Cada dato que se introduce en un prompt se convierte en un dato que queda fuera del control de una organización. Los usuarios pueden compartir información sensible, confidencial o regulada, como información de identificación personal (PII), datos financieros, propiedad intelectual y código fuente. E incluso si no comparten información confidencial explícita, es posible reconstruir el contexto a partir de las entradas para inferir datos confidenciales. Como protección, los empleados pueden cambiar una configuración para evitar que sus entradas sigan entrenando al modelo, pero deben hacerlo de forma manual. Para garantizar la seguridad, las organizaciones deben evitar que las personas ingresen datos privados (Cloudflare, 2024).

En definitiva, la privacidad y la seguridad representan algunos de los desafíos más críticos para las organizaciones al implementar soluciones basadas en IA y GenAI. Superar estos retos de forma eficaz puede marcar la diferencia entre que estas tecnologías se perciban como un riesgo operativo o como un motor de transformación empresarial alineado con las normativas vigentes. Abordar adecuadamente los aspectos de privacidad implica proteger la propiedad intelectual, resguardar los datos personales o sensibles y establecer controles de acceso robustos sobre los modelos y sus resultados. No obstante, persisten importantes brechas en la protección de los conjuntos de datos utilizados en IA, lo que evidencia la necesidad de seguir fortaleciendo las estrategias de gobernanza, seguridad y cumplimiento normativo en esta área (S&P Global, 2025).

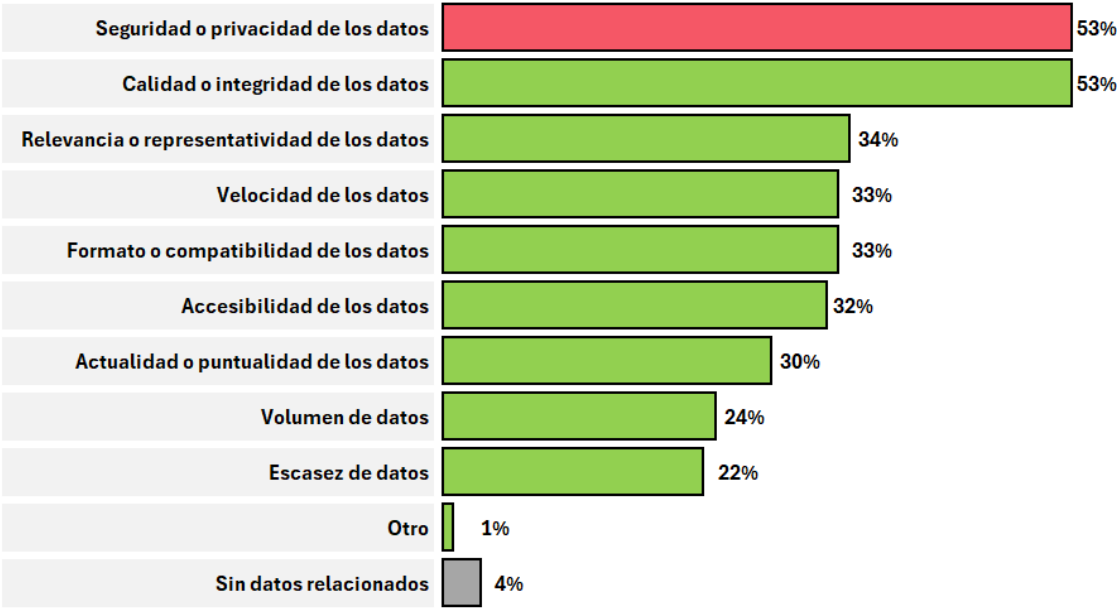


Ilustración 31: Desafíos de datos asociados a la implementación de GenAI

Fuente: (S&P Global, 2025)

Propiedad Intelectual

El uso de materiales con derechos de autor en el entrenamiento de modelos GenAI plantea interrogantes sobre la vulneración de la propiedad intelectual. Si una herramienta GenAI genera contenido muy similar a una obra protegida por derechos de autor, podría dar lugar a demandas legales y disputas sobre la propiedad. Este enfoque pone de manifiesto la incertidumbre legal que rodea el uso de materiales con derechos de autor en el entrenamiento de modelos GenAI y la posibilidad de litigios cuando se alega una infracción de derechos de propiedad intelectual (Tishkoff PLC, 2025).

La IA Generativa ha abierto nuevos horizontes en la creación de contenido, desde textos hasta obras de arte, desafiando los paradigmas tradicionales sobre la autoría y la propiedad intelectual; sin embargo, estas “creaciones” son el resultado de analizar y procesar enormes cantidades de obras humanas previas, pero al depender de obras preexistentes para generar nuevas, desdibuja las líneas entre la inspiración, la derivación y la copia directa, complicando la atribución y la compensación justa a los creadores originales. Si utilizamos como referencia el Reglamento General de Protección de Datos (RGPD) de la Unión Europea, veremos que ofrece un marco para la protección de datos pero no aborda directamente las complejidades asociadas con las creaciones de GenAI, agravando este vacío legal la ausencia de una figura de “autoría artificial” en las legislaciones actuales sobre derechos de autor, que tradicionalmente han protegido las obras originadas en el intelecto humano (García, 2024).

7.6.3 Desafíos Humanos & Capacitación

En el contexto de los desafíos asociados directamente al equipo de trabajo, hay que recordar que la IA Generativa es una herramienta que más allá de su enorme potencial debe ser utilizada por equipos de trabajo (humanos, no máquinas), y por lo mismo, es relevante revisar los principales desafíos que se desprenden de esta arista:

Dinámica Laboral y Resistencia Cultural

En los casos en que se considera la AI Generativa para mejorar procesos tradicionales, es posible que se requieran ajustes en los roles y responsabilidades de los profesionales del equipo de trabajo. Esto probablemente genere tensión en los colaboradores en dos sentidos: el temor a la pérdida de puestos de trabajo y la alteración de los roles tradicionales. Si bien es probable que la GenAI complemente los puestos de trabajo en lugar de reemplazarlos, permitiendo a los analistas centrarse en tareas de mayor nivel que requieren juicio humano, la percepción de pérdida de empleo es un problema importante por considerar (Roberts et al., 2014).

Es altamente probable que una solución tecnológica de la envergadura del GenAI muestra una considerable resistencia cultural al cambio, lo que dificulta la adopción por parte de los equipos. Estas barreras requieren un enfoque proactivo para la gestión del cambio, que implique una comunicación clara y transparente y una amplia participación de las partes interesadas. Al igual que con otras grandes transformaciones digitales, las organizaciones deberán implementar programas de apoyo estructurados y personalizados para ayudar a los profesionales a adaptarse a las funciones mejoradas por la IA. Este apoyo institucional podría incluir programas y cursos de formación específicos. Es probable que se requiera un sistema de formación continua, ya que es probable que las capacidades (y los riesgos) de la IA general sigan evolucionando.

El miedo al cambio es una constante en la historia de la humanidad. Cada revolución tecnológica ha despertado recelos, alarmas y, en ocasiones, resistencia violenta. Sin embargo, con el tiempo, esas mismas innovaciones se integran en la sociedad hasta volverse indispensables. La inteligencia artificial no es una excepción: sigue un patrón predecible de temor inicial seguido por una adopción masiva, aunque con particularidades que podrían acelerar o ralentizar su aceptación.

Capacitación

La adopción generalizada de la inteligencia artificial enfrenta un obstáculo significativo: la brecha de habilidades. Diversas estimaciones indican que actualmente solo existe talento capacitado para cubrir aproximadamente la mitad de los puestos relacionados con IA. Por su parte, un estudio de Capital One reveló que apenas el 36% de los profesionales tecnológicos confía en que su organización posee la experiencia necesaria para implementar proyectos complejos de IA. Esta desconexión entre la creciente demanda y la limitada inversión en el desarrollo de competencias en IA por parte de las organizaciones resulta preocupante. De hecho, un informe del IBM Institute for Business Value de 2023 estima que el 40% de la fuerza laboral necesitará adquirir nuevas habilidades en los próximos tres años debido al avance de la automatización y la inteligencia artificial (Blend, 2025).

La formación general en ingeniería de prompt aporta valor, pero su impacto en la productividad es limitado en comparación con una capacitación personalizada, diseñada según los roles y funciones específicas. Un error común en muchas organizaciones es delegar la validación de modelos de IA en analistas junior, asignando una etapa crítica del entrenamiento a perfiles que carecen del conocimiento técnico especializado. Sin embargo, la inteligencia artificial alcanza su máximo potencial cuando es entrenada y supervisada por expertos en la materia, capaces de optimizar su desempeño y alinearla con los objetivos estratégicos del negocio. La colaboración con especialistas del sector permite incorporar

conocimientos técnicos avanzados y enfoques creativos en el diseño de programas de formación. Al implementar capacitaciones prácticas, orientadas a roles específicos, las organizaciones pueden reducir de forma eficiente las brechas de habilidades existentes y fortalecer la confianza de los empleados en el uso de tecnologías de inteligencia artificial en su entorno laboral.

La reconversión del talento no debe abordarse como un proceso absoluto, sino como una estrategia progresiva y flexible. Las organizaciones deberían ofrecer formación específica según las funciones a aquellos empleados que demuestren interés, incentivándolos además a compartir sus aprendizajes con sus compañeros. Al comprometerse con una capacitación estructurada en inteligencia artificial, las empresas no solo mitigan la escasez inmediata de talento, sino que también promueven una cultura de aprendizaje continuo e innovación, elementos fundamentales para el éxito sostenible en una economía cada vez más impulsada por la IA. En el caso particular de aseguramiento de calidad de software ya están apareciendo capacitaciones y certificaciones especializadas en el uso de GenAI para pruebas de software, siendo la alternativa más reconocida a la fecha la que propone el organismo Brightest y su certificación: AiU Certified GenAI-Assisted Test Engineer (GenAiA-TE).

7.6.4 Desafíos Ambientales

El desarrollo e implementación de sistemas de IA conllevan un consumo significativo de recursos, particularmente energéticos, planteando un desafío ambiental considerable. El entrenamiento de estos modelos puede emitir cantidades de CO₂ comparables a las de un vehículo durante toda su vida. Para mitigar estos efectos, resulta imperativo considerar medidas de eficiencia energética y la transición hacia fuentes de energía renovable.

A raíz de estos surge la necesidad de incluir específicamente a la IA en las estrategias de mitigación del cambio climático y en los esfuerzos de sostenibilidad ambiental. La adopción de una ética del cuidado en el desarrollo de la IA Generativa enfatiza la importancia de considerar las consecuencias a largo plazo de nuestras acciones tecnológicas, promoviendo un enfoque holístico que valora la interdependencia entre la tecnología, la sociedad y el medio ambiente (García, 2024).

8. RESULTADOS OBTENIDOS

Los resultados de esta investigación deben interpretarse no solo como avances técnicos en el proceso de testing, sino como evidencia sólida para respaldar decisiones estratégicas. En un contexto donde la presión por entregar con rapidez y calidad es permanente, disponer de datos que acrediten mejoras reales en eficiencia, cobertura y reducción de errores puede ser determinante para elegir entre mantener un modelo tradicional o evolucionar hacia uno adaptativo impulsado por GenAI.

Previamente revisamos en detalle las mejoras evidentes que conlleva la incorporación de soluciones de inteligencia artificial generativa como parte del proceso de aseguramiento de calidad del software. Los datos obtenidos nos permitirán no solo cuantificar las mejoras logradas, sino también reflexionar sobre el impacto real de esta tecnología en etapas clave del testing, haciendo hincapié en las mejoras significativas de eficiencia, cobertura, precisión y adaptabilidad del proceso al contexto cambiante y profundamente dinámico de los proyectos actuales.

Desde una perspectiva ejecutiva, uno de los hallazgos más relevantes del estudio es la reducción de costos operacionales derivada de la implementación de estrategias de prueba adaptativas basadas en GenAI. Esta tecnología no solo optimiza los tiempos del proceso de pruebas, sino que también apunta a reducir de manera significativa la dependencia de recursos humanos altamente especializados, permitiendo una reasignación estratégica del talento hacia tareas de mayor valor para el proyecto. En términos concretos, este impacto tiene implicancias directas para la rentabilidad de los proyectos de software, ya que permite acortar el time-to-market, reducir la aparición de errores en entorno productivo y mejorar la calidad entregada sin incrementar proporcionalmente la inversión. En consecuencia, la adopción de GenAI en testing se alza como una herramienta estratégica para generar eficiencias operativas sostenibles, facilitando decisiones de negocio más informadas y alineadas con los objetivos financieros de la organización.

8.1 Mejoras en Etapas Clave

Lo primero que debemos tener en consideración para efectos de este análisis es que nuestra hipótesis no abordó la totalidad de las actividades que conforman el proceso de pruebas, pero si se enfocó en 4 áreas clave y estratégicas dentro del mismo: análisis de requerimientos, diseño y creación de matrices de prueba, ejecución de pruebas y generación de informes de resultados.

La instancia de levantamiento de requerimientos es sin lugar a duda la etapa más importante dentro del proceso de desarrollo de software, ya que de ella depende en gran medida, la calidad y el éxito de todas las etapas posteriores. Una fase donde muchas veces no se hace participe al equipo de QA, debido principalmente a la falta de cultura en lo que respecta a la ingeniería de requerimientos y a los costos asociados de introducir al equipo de testing en etapas tempranas. Precisamente en esta línea es que cobran gran relevancia los múltiples beneficios que se desprenden de incorporar herramientas de Inteligencia Artificial Generativa (GenAI) como parte del proceso de inspección de requerimientos, resaltando mejoras significativas en las dos aristas que se levantaron preliminarmente: reducción de los tiempos de inspección y aumento en la detección de defectos.

En lo que respecta a la tasa de revisión (rate of review), los datos recopilados sugieren que la utilización de herramientas de GenAI permiten una reducción de un 60% del tiempo invertido en la inspección, mientras que la detección de defectos se incrementa en un 40% respecto a la revisión manual/visual (por un humano). Como no existen estudios oficiales o valores concretos respecto a estas mejoras, se hicieron pruebas mediante la creación de un GPT Privado entrenado en inspección de requerimientos de software en base a las Normas ISO/IEC/IEEE 29148 (Requirements Engineering), ISO/IEC 25010 (Product Quality Model) y ISO/IEC/IEEE 29119 (Testing).



Inspección de Requerimientos

Por Gustavo Eduardo Inzunza Rojas 

Este GPT sirve de apoyo a la ingeniería de requerimientos, en lo que respecta a las actividades de inspección en base a las buenas prácticas que definen las normas ISO/IEC/IEEE 29148, ISO/IEC 25010 e ISO/IEC/IEEE 29119.

+ Pregunta lo que quieras



Ilustración 32: GPT Privado para Inspección de Requerimientos

Para obtener las métricas asociadas a la inspección realizada por el modelo de IA (inspección de requerimientos) se realizaron inspecciones a 5 documentos base de especificación de requerimientos (SRS), y se compararon los resultados obtenidos respecto a tiempo invertido y defectos detectados con la misma tarea realizada por dos Analistas de Requerimientos, certificados bajo el estándar IREB Certified Professional for Requirements Engineering (CPRE).

Inspección de Requerimientos 5 ▾ 🔗 Compartir

Ya revisé tu documento de **Especificación de Requerimientos (SIGSA)** aplicando buenas prácticas de inspección basadas en **ISO/IEC/IEEE 29148** (calidad de requisitos), **ISO/IEC 25010** (modelo de calidad del producto) y defectos comunes en documentación de requisitos.

Aquí tienes el resultado:

Datos clave de la inspección

- **Duración de la inspección con IA:** 0.2 horas (≈ 12 minutos de procesamiento automatizado).
- **Duración de la inspección estimada por humano:** 3.5 horas (lectura + análisis detallado de ~40 páginas a ritmo de 1-2 pág/hora).
- **Cantidad de defectos detectados:** 15.

Tabla de defectos encontrados

ID	Tipo de Defecto	Descripción	Ubicación	Recomendación
D01	Ambigüedad	Uso de términos vagos: "rápido", "sencillo", "adecuado" sin métricas.	RNF01, RNF05, 2.8.3	Definir métricas (ej. tiempo de respuesta <2s, curva de aprendizaje <1h).
D02	Inconsistencia	RF10 "Eliminar registros" mezcla	RF10	Separar en dos requisitos:

+ Pregunta lo que quieras 🔊 acción lógica y modificación de

Ilustración 33: Ejemplo de Inspección de Requerimiento mediante IA

Una vez promediados los resultados para cada una de las 5 rondas de inspección se tomaron los mejores desempeños entre los Analistas (tiempo y efectividad) y se compararon con los valores resultantes de la interacción con el GPT, y de esta forma se determinaron valores de referencia respecto a las mejoras detectadas:

- Reducción del tiempo de inspección de requerimientos en un 1400%.
- Detección de defectos se incrementó en un 62%

	Analista 1		Analista 2		IA		Mejora al usar GenAI	
	Tiempo (hrs)	N° Defectos	Tiempo (hrs)	N° Defectos	Tiempo (hrs)	N° Defectos	Tiempo (hrs)	N° Defectos
Documento SRS 1	3.4	9	3	5	0.2	15	1500%	66%
Documento SRS 2	3	8	3.2	5	0.3	13	1000%	62%
Documento SRS 3	2.8	4	2.9	6	0.1	10	2800%	66%
Documento SRS 4	3.1	5	3.1	7	0.3	11	1033%	57%
Documento SRS 5	2.2	8	2.5	10	0.2	16	1100%	60%

Ilustración 34: Comparativo Inspección Manual vs Inspección con IA

En lo que respecta a la utilización de GenAI para el diseño y creación de matrices de prueba destaca la capacidad de esta tecnología para generar automáticamente casos de prueba más completos, alineados con escenarios reales y basados en la documentación del proyecto, lo que incrementa la cobertura y reduce la posibilidad de defectos que no sean detectados en la instancia de pruebas. Si usamos como referencia el estudio realizado por (Hall, 2024) que toma como ejemplo el caso del proceso de certificación de una plataforma de comercio electrónico, se evidencia que el tiempo necesario para la creación de casos de prueba se vio reducido en un 50%, garantizando además una cobertura completa de diversos escenarios de prueba. Estos datos se validan y complementan con los planteados por (Kumar, 2025) que postulan una mejora de un 75% en los tiempos de generación de casos de prueba con un incremento en la cobertura de pruebas en un 60%, con una mayor mitigación de riesgos técnicos y operativos.

Para profundizar aún más en la generación de casos de prueba mediante GenAI, podemos analizar el experimento realizado por (Van Hoang, 2025) para descubrir si el uso de IA para generar casos de prueba reduciría el tiempo de generación, mejorando la cobertura de las pruebas (especialmente para excepciones y escenarios complejos) y la coherencia en la estructura de los casos de prueba en comparación con la generación manual. El éxito del experimento se midió por la reducción del tiempo de escritura de los casos de prueba, una mayor cobertura de las pruebas y la reducción de las modificaciones necesarias.

El experimento siguió una metodología controlada en la que se compararon los casos de prueba generados por IA con los creados manualmente utilizando las mismas historias de usuario como entrada. Se establecieron métricas de referencia para la eficiencia del tiempo, la cobertura de las pruebas y la consistencia de la calidad para garantizar una evaluación objetiva.

Dentro de los principales resultados extraídos del experimento se pueden destacar:

- Se demostró la capacidad de crear casos de prueba iniciales mucho más rápido que los métodos de escritura manual. Las métricas muestran una eficiencia de tiempo promedio del 80,07%.
- Destaca la capacidad de mantener un formato y una estructura consistentes en todos los casos de prueba. El índice de consistencia promedió un 96,11%, significativamente superior al de los enfoques manuales tradicionales.
- En historias de usuario sencillas con requisitos funcionales claros, el experimento demostró que GenAI tiene un rendimiento excepcional al cubrir las principales áreas funcionales. La IA alcanzó una cobertura del 98,67 % de los criterios de aceptación y mantuvo una baja tasa de duplicación de tan solo el 4,22 %.

Métrica	Descripción	Método de Medición
Corrección (%)	Porcentaje de casos de prueba que coinciden correctamente con los requisitos	Revisión QA y puntuación: (Casos correctos / Total de casos de prueba manuales) × 100
Cobertura de criterios de aceptación (%)	Cobertura de criterios de aceptación por casos de prueba generados por LLM	Matriz de trazabilidad de requisitos: (Criterios cubiertos / Total de criterios) × 100
Tasa de duplicación (%)	Porcentaje de casos de prueba redundantes	Revisión manual: (Casos duplicados / Total de casos de prueba generados por LLM) × 100
Casos de prueba incorrectos (%)	Porcentaje de casos de prueba incompletos o irrelevantes	Revisión y marcado: (Casos incorrectos / Total de casos de prueba generados por LLM) × 100
Índice de ambigüedad (%)	Porcentaje de casos de prueba con pasos poco claros	Revisión QA de ambigüedad: (Casos ambiguos / Total de casos de prueba generados por LLM) × 100
Índice de consistencia (%)	Adherencia a plantillas de formato estructurado	Análisis de estructura: (Casos bien formateados / Total de casos de prueba generados por LLM) × 100
Impacto de la optimización de prompts (%)	Mejora después del refinamiento del prompt	Puntuación comparativa: [(Puntaje mejorado - Puntaje inicial) / Puntaje inicial] × 100
Eficiencia de tiempo (%)	Reducción de tiempo frente a la creación manual	Seguimiento de tiempo: [(Tiempo manual - Tiempo con LLM) / Tiempo manual] × 100

Ilustración 35: Experimento de creación de Casos de Prueba mediante IA

Fuente: (Van Hoang, 2025)

Adicionalmente, el impacto económico es considerable; diversos estudios de los antes mencionados apuntan que los enfoques de pruebas basados en IA ofrecen un retorno de la inversión (ROI) promedio del 283% durante el primer año de implementación. En líneas generales se pueden deducir los siguientes resultados clave de operar con GenAI:

- Creación de matrices de prueba entre un 75-80% más rápido.

- Cobertura de pruebas (generales) se aumenta en al menos un 60%.

En lo relativo al uso de inteligencia artificial generativa en la etapa de ejecución de pruebas se puede destacar su capacidad de análisis en tiempo real, que se traduce en que GenAI puede identificar patrones de fallos, priorizar casos críticos y adaptar dinámicamente la estrategia de ejecución según el comportamiento del sistema. Esto permite una detección temprana de defectos, una reducción en los tiempos de validación y un uso óptimo de los recursos, garantizando una mayor cobertura de pruebas y una validación más robusta del software. Para profundizar en los beneficios anexos podemos volver a analizar el ejemplo de certificación de una plataforma de comercio electrónico, planteado en el estudio de (Hall, 2024), donde se menciona una reducción del 40% en los tiempos de ejecución lo que permitió lanzamientos más frecuentes y correcciones más rápidas de errores. Por otro lado, (AI Testing Tools, 2024) plantea que gracias a la incorporación de GenAI en los procesos de certificación, se logra una reducción del tiempo invertido en esta actividad en un 50%.

(OptiSol Business Solutions, 2025) en su estudio de cómo las pruebas automatizadas impulsadas por GenAI están revolucionando la calidad del software empresarial plantea que la ejecución de pruebas es un 96% más rápida, con una tasa de detección de defectos de un 300% más efectiva que en el testing tradicional y una disminución de falsos positivos (precisión) en un 85%.

Métrica de Prueba	Antes (Pruebas tradicionales)	Después (Pruebas potenciadas con GenAI)	Mejora
Tiempo de generación de casos de prueba	2-3 semanas	2-4 horas	95% de reducción de tiempo
Tiempo de ejecución de pruebas	8-12 horas	15-30 minutos	96% de ejecución más rápida
Cobertura de pruebas	60-70%	95-98%	40% mejor cobertura
Tasa de detección de defectos	40-60 errores por release	150-200 errores por release	300% más defectos encontrados
Tasa de falsos positivos	25-40%	3-8%	85% de mejora en precisión
Esfuerzo de mantenimiento de pruebas	30-40% del tiempo de prueba	5-10% del tiempo de prueba	75% de reducción en mantenimiento
Frecuencia de pruebas de regresión	Semanal / Quincenal	Continua (24/7)	Frecuencia ilimitada
Pruebas multiplataforma	3-5 plataformas	20+ plataformas	500% más plataformas
Frecuencia de liberación	Mensual / Trimestral	Semanal / Diaria	400% liberaciones más rápidas

Ilustración 36: Mejoras en Pruebas Automatizadas Impulsadas por GenAI

Fuente: (OptiSol Business Solutions, 2025, 2025)

Según lo planteado por (Mishra, 2024), el tiempo de ejecución es un factor crítico para determinar la eficiencia de las pruebas automatizadas. Tal como se evidencia en la Ilustración 23 de análisis comparativo de rendimiento, los resultados de su análisis indican que los modelos de IA Generativa redujeron significativamente el tiempo de ejecución de las pruebas en comparación con otros modelos y/o

herramientas de uso común en equipos de prueba modernos (por ejemplo: Selenium), siendo dos veces más rápidas que los métodos tradicionales y con tasas de falsos positivos más bajas (3-5%) en comparación con otras herramientas (7-8%), lo que sugiere que las pruebas basadas en IA detectan con mayor precisión los defectos reales. De este conjunto de datos podemos desprender las siguientes conclusiones:

- Ejecución de pruebas al menos un 50% más rápida.
- Reducción de falsos positivos en un 30%.

Por último, históricamente los informes de resultados han sido extensos, técnicos y difíciles de interpretar, especialmente para las partes interesadas sin conocimientos técnicos y los tomadores de decisión. GenAI mejora la generación de informes mediante:

- Resumen de los resultados de la ejecución de pruebas en un lenguaje sencillo.
- Evidenciar de manera automática los fallos críticos y sus posibles causas raíz.
- Redactar informes de errores completos (pasos, configuración del entorno, registros, capturas de pantalla y sugerencias).
- Presentar información relevante según tipo de audiencia.

Las herramientas de inteligencia artificial generativa permiten consolidar automáticamente grandes volúmenes de datos de ejecución de pruebas (múltiples ciclos e iteraciones), transformándolos en reportes claros, bien estructurados y personalizados según las necesidades de los usuarios particulares que tendrán acceso al mismo (PMO, desarrollo, gestión, directores, etc.). Además, reducen significativamente la carga manual, minimizan errores humanos y aceleran la disponibilidad de información relevante clave para tomar decisiones respecto al paso a producción de los productos de software bajo evaluación.

Estos informes inteligentes reducen drásticamente el tiempo que los ingenieros dedican a interpretar los resultados de las pruebas, a la vez que proporcionan información más detallada sobre el rendimiento del sistema y las posibles vulnerabilidades que, de otro modo, podrían pasar desapercibidas en los flujos de trabajo manuales tradicionales. Más allá de la teoría que postula mejoras importantes en lo que respecta a la generación de informes de resultados mediante inteligencia artificial generativa, no fue posible identificar estudios o casos prácticos donde extraer métricas que permitan validar de manera consistente la hipótesis inicial respecto a una reducción del 80% de los tiempos asociados a esta tarea.

8.2 Resumen

Desde una perspectiva estratégica y orientada a la toma de decisiones, los resultados obtenidos a través de la incorporación de estrategias de prueba adaptativas basadas en inteligencia artificial generativa (GenAI) revelan una transformación significativa en términos de eficiencia, calidad y velocidad del proceso de aseguramiento de calidad de software. Estos avances no solo cumplen con los supuestos establecidos en la hipótesis inicial, sino que, en la mayoría de los casos, los superan de forma considerable.

En términos de impacto organizacional, estas mejoras se traducen directamente en una reducción sustancial de los costos operativos, debido a la menor inversión de horas-hombre en tareas críticas del proceso de pruebas, al mismo tiempo que se incrementa la capacidad de entrega ágil y se disminuyen los riesgos asociados a errores en producción. Esto posiciona a las organizaciones en un escenario competitivo más favorable, al permitir ciclos de liberación de software más cortos, mayor confiabilidad de los productos entregados y una mejora sostenida en la percepción del usuario final.

En la etapa de análisis de requerimientos, se logró una reducción del tiempo de inspección del 1400%, superando ampliamente el supuesto inicial del 400%, junto con un aumento del 62% en la detección de defectos, superando también el umbral proyectado del 50%. Este resultado evidencia una mejora sustancial en la calidad desde las primeras fases del ciclo de vida del software, lo que disminuye costos de corrección posteriores y refuerza el principio de prevención por sobre la detección. En cuanto al diseño y creación de matrices de prueba, la hipótesis apuntaba a una reducción del 80% en tiempo y un aumento del 40% en cobertura. Los resultados obtenidos se mantuvieron dentro del rango proyectado en cuanto a velocidad (75-80%) y superaron con claridad la expectativa en cobertura, alcanzando un mínimo de 60%, lo que refleja una mayor profundidad y precisión en la validación funcional del software.

Durante la ejecución de pruebas, se alcanzó una aceleración del 50% y una reducción del 30% en falsos positivos, ambas cifras por encima de los valores estimados (30% y 20% respectivamente). Este escenario permite disminuir la carga de retrabajo, optimizar la asignación de recursos humanos y generar resultados de pruebas más confiables y accionables. Finalmente, en cuanto a la generación de informes de resultados, si bien se proyectaba una mejora del 80% en los tiempos de elaboración, no se identificaron métricas concretas para evaluar este punto, lo que sugiere una oportunidad de mejora futura en la instrumentación y medición de esta etapa del proceso.

En conjunto, los resultados obtenidos no solo validan la efectividad de la integración de GenAI en el proceso de testing, sino que también demuestran su potencial como catalizador de eficiencia operativa, calidad técnica y diferenciación competitiva en el aseguramiento de la calidad del software.

9. CONCLUSIONES

Esta investigación demuestra que el uso de GenAI en el aseguramiento de calidad no solo es técnicamente viable, sino estratégicamente imprescindible. En un entorno donde la calidad del software incide directamente en la reputación, la continuidad operativa y la eficiencia financiera, adoptar enfoques adaptativos y automatizados deja de ser una decisión técnica para convertirse en una prioridad empresarial. Las organizaciones que integren estas capacidades no solo reducirán defectos, sino que también incrementarán su velocidad, resiliencia y capacidad de respuesta frente a contextos dinámicos.

El proceso de aseguramiento de la calidad de software (QA), históricamente concebido como un conjunto de prácticas rígidas y de carácter principalmente reactivo, en la actualidad enfrenta desafíos profundos y estructurales derivados de entornos marcados por la alta incertidumbre y el cambio constante. La problemática central radica en la limitada capacidad de adaptación de las estrategias de prueba ante la evolución dinámica de los requerimientos durante el ciclo de vida del desarrollo, algo aún más marcado en metodologías de desarrollo ágil y en modelos DevOps. Esta brecha no solo impacta en un incremento en los costos asociados al proceso de desarrollo, sino que también en la eficiencia operativa de los equipos, comprometiendo la confiabilidad, la estabilidad y la calidad de los productos entregados al mercado.

En este contexto, la Inteligencia Artificial Generativa (GenAI) se perfila como una solución transformadora, capaz de redefinir los paradigmas del proceso de pruebas mediante la incorporación de mecanismos predictivos, adaptativos y automatizados que fortalecen la toma de decisiones en QA. GenAI no solo facilita la automatización de tareas repetitivas, sino que también habilita la detección anticipada de defectos, garantiza la trazabilidad frente a cambios en los requerimientos y permite ajustar dinámicamente los escenarios de prueba y los informes de resultados. La adopción de estrategias de prueba adaptativas impulsadas por esta tecnología constituye un cambio de paradigma: se transita desde un enfoque estático y fundamentado en la invariabilidad de los requerimientos (funcionales y no funcionales) hacia una perspectiva dinámica, estratégica y orientada al aprendizaje continuo. Este nuevo marco no solo fortalece la alineación del proceso de QA con los objetivos

organizacionales, sino que también acelera el time to market, optimiza los costos operativos y garantiza la entrega de productos con estándares superiores de calidad.

Desde la perspectiva de la gestión empresarial, este estudio ofrece una mirada estratégica que trasciende la dimensión tecnológica. La incorporación de GenAI en los procesos de aseguramiento de calidad constituye una oportunidad tangible para que los tomadores de decisión alineen los objetivos de TI con las metas del negocio, optimizando recursos, gestionando riesgos operativos con mayor precisión y acelerando la entrega de valor. En un escenario donde la diferenciación competitiva se sustenta en la agilidad y la confiabilidad de los productos de software, el uso inteligente de estas tecnologías emergentes se consolida como un habilitador crítico de eficiencia, innovación sostenible y ventaja estratégica. Esta visión integrada permite a los líderes empresariales no solo perfeccionar procesos, sino también transformar modelos operativos, fomentar una cultura de calidad basada en datos y fortalecer la resiliencia organizacional frente a entornos de disrupción constante.

Los resultados que surgen de este análisis permiten concluir, con respaldo empírico y evidencia sólida, que la aplicación de estrategias de prueba adaptativas basadas en GenAI constituye un factor decisivo para la evolución del aseguramiento de calidad de software, generando mejoras sustantivas tanto en la dimensión táctica como en la estratégica, y su respectiva optimización del proceso y sus costos. Desde una perspectiva operativa, se destacan incrementos significativos en la cobertura de pruebas, mayor precisión en la detección temprana de defectos, una reducción notable en los tiempos de ejecución y una optimización en la asignación de recursos humanos y tecnológicos, lo que permite redireccionar esa capacidad a otras tareas clave. Estas mejoras no solo se traducen en beneficios cuantificables, como la reducción de costos de mantenimiento, la agilización de los ciclos de prueba o el incremento en la productividad de los equipos, sino que también fortalecen la capacidad organizacional para adaptarse con agilidad a la dinámica del negocio y a la constante evolución de los requerimientos, consolidando así una ventaja competitiva sostenible.

En el plano estratégico, la integración de GenAI representa un salto cualitativo al transformar un enfoque tradicionalmente reactivo en uno proactivo y predictivo, estableciendo las bases de una cultura de calidad continua respaldada por analítica avanzada y decisiones basadas en datos. Este cambio permite maximizar la trazabilidad entre requerimientos y casos de prueba, eliminar ineficiencias asociadas a tareas complejas y/o redundantes y asegurar procesos de certificación más confiables en escenarios de alta volatilidad. La capacidad de adaptación en tiempo real no solo optimiza la gestión de riesgos tecnológicos, sino que también se convierte en un factor diferenciador clave para acelerar la entrega de valor,

fortalecer la confianza del cliente y consolidar la competitividad en mercados donde la rapidez y la calidad del software determinan el éxito empresarial.

Asimismo, la incorporación de modelos de lenguaje generativos y de arquitecturas avanzadas como Retrieval-Augmented Generation (RAG) potencia de manera significativa la capacidad del área para evaluar e interpretar con mayor precisión los requerimientos, automatizar la inspección documental, generar casos de prueba de escenarios complejos y no tan evidentes, además de enriquecer la experiencia transversal del equipo de proyecto. Estos avances vienen también a redefinir el rol del profesional del área de QA, quien evoluciona desde una función tradicional predominantemente operativa hacia una posición estratégica de orquestador y validador del proceso de calidad, apoyado en herramientas inteligentes que amplifican su impacto en la organización.

Los hallazgos plasmados en este trabajo confirman que la adopción de GenAI trasciende más allá de una tendencia tecnológica y se posiciona como una respuesta estratégica y necesaria frente a los desafíos estructurales que enfrenta hoy el aseguramiento de calidad de software. Una implementación adecuada de la inteligencia artificial generativa no solo incrementa la eficiencia y precisión en las pruebas, sino que también redefine el rol y el alcance del QA como disciplina, impulsando procesos más resilientes, adaptativos y alineados al valor de negocio. En definitiva, GenAI no solo optimiza el aseguramiento de calidad, sino que redefine las bases que históricamente han predominado, posicionándolo como un habilitador estratégico en la economía digital y un pilar esencial para el éxito y diferenciación de las organizaciones del área del aseguramiento de calidad de software.

10. TRABAJO FUTURO

A medida que la Inteligencia Artificial (IA) y, en particular, la Inteligencia Artificial Generativa (GenAI) continúan evolucionando, surgen numerosas oportunidades para ampliar y profundizar su aplicación en el aseguramiento de calidad del software. El presente trabajo ha demostrado el potencial de las estrategias de prueba adaptativas impulsadas por GenAI para optimizar los principales procesos de QA, sin embargo, este campo aún se encuentra en una etapa de madurez preliminar. Existen múltiples líneas de investigación y desarrollo que pueden ser exploradas en el futuro, tales como la automatización inteligente de pruebas exploratorias, la generación dinámica de casos de prueba basados en aprendizaje continuo, el uso de modelos generativos para predecir defectos antes de la ejecución del código, y la integración de GenAI en entornos DevOps para lograr pipelines de prueba más ágiles y autónomos.

A medida que el campo de la IA continúa evolucionando, se espera que varias tendencias e innovaciones den forma al futuro de las pruebas de software impulsadas por IA (Hall, 2024), tales como:

- **Mayor adopción de herramientas de prueba basadas en IA.** Se prevé un crecimiento significativo en la adopción de herramientas de prueba basadas en IA. Es probable que cada vez más pymes integren la IA en los marcos de prueba convencionales y aprovechen sus beneficios en sus procesos de prueba.
- **Avances en las pruebas autónomas.** Las pruebas autónomas, en las que los sistemas de IA diseñan, ejecutan y analizan pruebas de forma independiente sin intervención humana, experimentarán importantes avances. Esto reducirá aún más la necesidad de supervisión manual y mejorará la eficiencia y la precisión del proceso de prueba.
- **Análisis predictivo mejorado.** Los sistemas de IA utilizarán cada vez más el análisis predictivo para pronosticar posibles problemas y áreas de preocupación en las aplicaciones de software. Este enfoque proactivo permitirá a los equipos de desarrollo abordar los problemas antes de que afecten a los usuarios finales, lo que se traducirá en un software más estable y fiable.
- **Integración con DevOps y pipelines de CI/CD.** La integración de herramientas de prueba basadas en IA con DevOps y pipelines de CI/CD será más fluida. Esto facilitará las pruebas y la monitorización continuas, garantizando que la calidad del software se mantenga durante todo el ciclo de vida del desarrollo y posteriormente.
- **Mayor énfasis en las pruebas de seguridad.** La IA desempeñará un papel fundamental en la mejora de las pruebas de seguridad. Las herramientas basadas en IA podrán identificar vulnerabilidades con mayor eficacia y simular ciberataques sofisticados para probar la resiliencia de las aplicaciones. Esto será crucial en una era donde las amenazas de ciberseguridad son cada vez más complejas.
- **Pruebas de experiencia de usuario mejoradas por IA.** La IA se utilizará para mejorar las pruebas de experiencia de usuario simulando interacciones reales de usuarios y proporcionando información sobre cómo interactúan con las aplicaciones de software. Esto ayudará a los desarrolladores a optimizar la interfaz de usuario y la experiencia general del usuario.
- **Colaboración entre IA y testers humanos.** Si bien la IA automatizará muchos aspectos de las pruebas de software, la colaboración entre IA y testers humanos seguirá siendo esencial. Los evaluadores humanos se

centrarán en pruebas exploratorias y áreas que requieran creatividad e intuición, mientras que la IA se encargará de tareas repetitivas y que requieren un uso intensivo de datos.

Ahora bien, podemos profundizar en las tendencias que se visualizan con mayor impulso en los próximos años y que eventualmente podrían abrir nuevas aristas de estudio e investigación para seguir explotando la inteligencia artificial como parte de los procesos de pruebas de software.

10.1 Hiper-Automatización en Pruebas

La hiper automatización va más allá de la automatización tradicional, que suele centrarse en la automatización de tareas individuales, para abarcar una gama más amplia de soluciones que permiten automatizar procesos empresariales completos. Su objetivo es lograr una automatización de alto nivel que pueda gestionar las tareas más complejas y desafiantes, a la vez que aprovecha las tecnologías avanzadas para mejorar la toma de decisiones e impulsar los resultados empresariales (Infovision, 2023).

La hiper automatización propone la automatización integral de todos los procesos que sean susceptibles de ser automatizados. En el ámbito del aseguramiento de calidad, esta se logra mediante la convergencia de múltiples tecnologías, como lo son: la automatización de pruebas UI/API, la automatización robótica de procesos (RPA), la automatización digital de procesos de negocio (DPA), la inteligencia artificial y el aprendizaje automático (ML), así como el uso de plataformas low-code. Esta combinación tecnológica permite acelerar la transformación digital al proporcionar un proceso de calidad más eficaz y eficiente, basado en soluciones avanzadas que trabajan en sinergia, en lugar de depender de integraciones fragmentadas de metodologías y herramientas tradicionales.

Es importante destacar que la hiper automatización no busca reemplazar las pruebas manuales, sino potenciar el proceso de control de calidad inteligente. Facilita que los analistas de calidad puedan automatizar rápidamente tareas repetitivas sin requerir conocimientos técnicos especializados, permitiéndoles enfocarse en pruebas exploratorias que revelen defectos funcionales más complejos. En este sentido, contribuye a democratizar el proceso de automatización de pruebas, permitiendo una colaboración más fluida entre analistas y especialistas técnicos, ampliando así la cobertura general de pruebas. Esto, a su vez, permite a las organizaciones adaptarse con agilidad a las cambiantes demandas del mercado mediante la implementación de servicios de automatización inteligente.

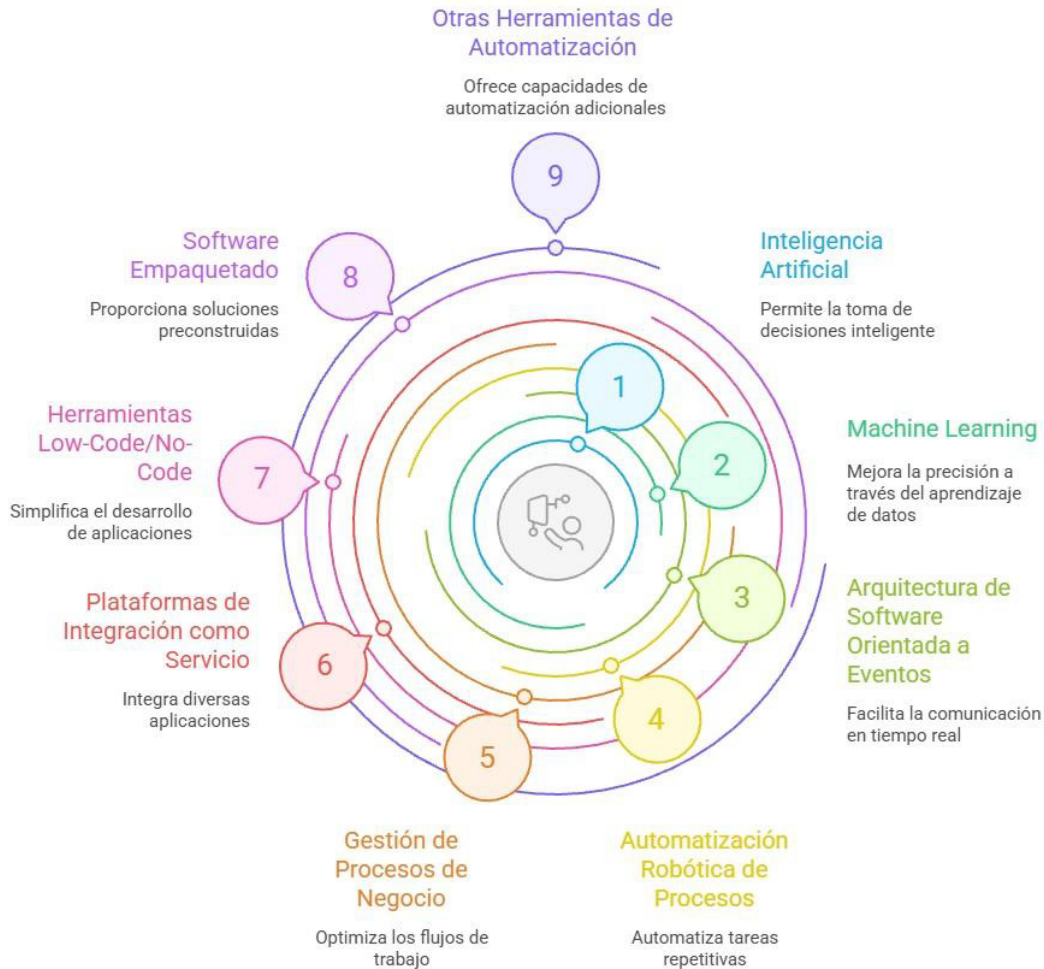


Ilustración 37: Tecnologías en un marco de hiper automatización

Fuente: (Infovision, 2023)

La integración de GenAI con las técnicas de hiper automatización promete transformar profundamente la creación y ejecución de casos de prueba en el aseguramiento de calidad del software. Los sistemas basados en GenAI son capaces de crear casos de prueba complejos de manera integral, directamente desde historias de usuario y especificaciones de requerimientos, lo que permite reducir de forma significativa el tiempo dedicado a la elaboración manual de pruebas. Además, estos sistemas no solo generan datos de prueba pertinentes, sino que también validan automáticamente los resultados y ofrecen retroalimentación en tiempo real durante la ejecución. Los modelos generativos, por su parte, pueden simular procesos de negocio y anticipar el comportamiento futuro del sistema, lo que permite ajustar continuamente los escenarios de prueba para mantener su relevancia a lo largo del ciclo de vida del software. Este enfoque no solo disminuye los requerimientos de tiempo y experiencia técnica necesarios para

construir suites de prueba robustas, sino que también incrementa la calidad y cobertura de las pruebas en entornos complejos e integrados.

10.2 Automatización de Pruebas con Autorreparación

Los scripts de prueba tradicionales se diseñan utilizando localizadores estáticos, como identificadores de elementos o XPath, para la identificación e interacción de elementos de la interfaz de usuario. Cuando la interfaz de usuario cambia, estos localizadores pueden cambiar, lo que puede provocar un fallo. La autorreparación permite una cobertura de pruebas más amplia, ya que las pruebas funcionarán a la perfección incluso con aplicaciones en constante evolución. Proporciona resultados de prueba precisos y ayuda a localizar defectos reales con mayor rapidez, en lugar de falsos positivos debido a pequeños cambios en la interfaz de usuario.

El sistema de autorreparación puede adaptarse manualmente a cualquier cambio necesario para garantizar la fiabilidad de las suites de pruebas, incluso si transforma aún más la aplicación. Este proceso permite el aprendizaje continuo sin necesidad de ir un paso más allá e intervenir manualmente cada vez, lo que convierte a este proceso de formación continua en una herramienta eficaz para la monitorización del sistema. Las capacidades de autorreparación basadas en IA/ML garantizan la detección automática de cambios en la aplicación bajo prueba y el ajuste de los scripts de prueba correspondientes. De esta forma, se reduce la presión de mantenimiento sobre los evaluadores, que tienen que lidiar con modificaciones frecuentes que ocurren dentro de las aplicaciones (Hemanth, 2025).

Los marcos modernos de automatización con autorreparación han revolucionado los enfoques de mantenimiento de pruebas. Un estudio exhaustivo de (Chiranjeevi, 2024) revela un aumento del 156% en las tasas de adopción entre las empresas entre 2021 y 2023. Este auge en la implementación se debe a resultados convincentes, ya que las organizaciones informan una reducción promedio del 82% en los esfuerzos de mantenimiento de pruebas al utilizar mecanismos de autorreparación basados en IA. Esta estrategia reduce el esfuerzo necesario para el mantenimiento de pruebas al adaptarse a los cambios. Ya sea una nueva función o una actualización de la funcionalidad existente, los mecanismos de autorreparación garantizan que los scripts de prueba se mantengan alineados con la aplicación de software.

10.3 DevOps Potenciado por IA

Muchas organizaciones creen que DevOps potenciado con IA es más adecuado para tareas que requieren mucho tiempo y esfuerzo. A medida que aumenta la velocidad de lanzamiento, los pipelines deben procesar un mayor volumen de datos con mayor frecuencia, lo que ejerce presión sobre todos los aspectos del proceso, especialmente las pruebas. A medida que las organizaciones maduran y escalan sus prácticas de DevOps, la automatización de pruebas se vuelve necesaria, sin embargo, es más fácil decirlo que hacerlo, incluso en organizaciones con DevOps consolidadas, las pruebas suelen ser el principal obstáculo.

La fusión de GenAI en las pruebas de software con DevOps dará lugar a canales de CI/CD hiper automatizados que pueden predecir fallos y resolverlos en tiempo real. Los principales avances que se visualizan apuntan:

- Análisis de la calidad del código impulsado por IA, que garantiza que cada confirmación cumpla con las mejores prácticas antes de la implementación.
- Detección predictiva de defectos, donde la IA pronostica posibles problemas antes de que afecten a la producción.
- Mecanismos de reversión inteligentes, donde la IA determina si se debe implementar, revertir o modificar una versión basándose en evaluaciones de riesgos.



Ilustración 38: Uso de IA/ML en Procesos de Pruebas

Fuente: (Rothman, 2024)

Según un estudio presentado en (Rothman, 2024) el 25 % de los encuestados consideran que la IA mejora las pruebas al reducir el mantenimiento de los casos de prueba (49%), permiten centrar las pruebas en las áreas de mayor riesgo (44%) e identificar la causa raíz de las pruebas fallidas (43%).

10.4 Predicción y Prevención de Defectos Impulsada por IA

La transición hacia un enfoque de aseguramiento de la calidad preventivo constituye una de las transformaciones más relevantes en la evolución de las metodologías de prueba de software. En este nuevo paradigma, la predicción de defectos impulsada por inteligencia artificial utiliza algoritmos de aprendizaje automático para anticipar posibles fallos antes de que se manifiesten en los entornos de producción, permitiendo así la ejecución de pruebas proactivas y la aplicación de medidas correctivas focalizadas. A continuación revisaremos los principios técnicos que sustentan estos sistemas (Seelamneni, 2025).

Las pruebas “Shift-Left” enfatizan la identificación temprana de defectos y han cobrado gran importancia en la industria del desarrollo de software. Según un análisis exhaustivo del sector, la corrección de los defectos identificados durante las primeras fases de desarrollo cuesta aproximadamente de 4 a 5 veces menos que la de los descubiertos en entornos de producción. La predicción de defectos basada en IA mejora sustancialmente las capacidades de desplazamiento a la izquierda al analizar los cambios de código, los patrones de desarrollo y el historial de defectos para identificar modificaciones de alto riesgo antes de las pruebas formales. Las investigaciones demuestran que los modelos predictivos que implementan algoritmos de árboles de decisión han alcanzado tasas de precisión en la predicción de defectos del 71-75% al entrenarse con datos históricos exhaustivos, mientras que los enfoques de bosque aleatorio elevan esta precisión al 84-86% en implementaciones maduras.

La eficacia de los sistemas de predicción de defectos depende fundamentalmente de la calidad de los datos y de los enfoques de optimización de modelos. Las investigaciones indican que los modelos de predicción integrales requieren conjuntos de datos de entrenamiento que abarquen al menos entre 180 y 250 defectos confirmados en múltiples categorías para lograr un rendimiento base aceptable. La selección de características representa un factor crítico de éxito, y las técnicas de reducción de dimensionalidad, como el análisis de componentes principales, demuestran la capacidad de mejorar la eficiencia del modelo entre un 15% y un 20%, manteniendo o mejorando la precisión de la predicción. importancia del refinamiento continuo del modelo utilizando datos internos de defectos.

11. REFERENCIAS BIBLIOGRÁFICAS

A continuación, se presentan las fuentes académicas y científicas utilizadas para la investigación y el desarrollo de esta tesina:

ACCELQ. (2025). AI in Testing: What works and What doesn't?. <https://sapinsider.org/wp-content/uploads/2025/01/White-paper-AI.pdf>

AI Testing Tools. (2024). How AI Refines Test Cases for Better Accuracy. <https://www.testingtools.ai/blog/how-ai-refines-test-cases-for-better-accuracy/>

Andrades, G. (2025). Smarter, Faster Testing with Generative AI-Powered Autopilot. <https://www.accelq.com/blog/generative-ai-in-software-testing/>

Arias, M. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. <https://www.redalyc.org/pdf/666/66612870011.pdf>

Artyukhovsky, A. (2025). La IA en el aseguramiento de la calidad y las pruebas de software: ¿exageración o realidad?. <https://innowise.com/es/blog/ai-in-software-testing/>

Barnes, L. (2024). Exploring AI in Software Testing. <https://fortegrp.com/insights/exploring-ai-in-software-testing>

Barrere, R., Matas, L., Sokil, J., Trama, L. (2025). Dossier Inteligencia artificial. UNESCO

Bigelow, S. (2019). How to tame ever-changing requirements in software development. <https://www.techtarget.com/searchsoftwarequality/tip/How-to-tame-ever-changing-requirements-in-software-development>

Blend. (2025). Los siete desafíos críticos: Estrategias comprobadas para escalar la AI. https://cdn.prod.website-files.com/64d08b14d93ac10636508f22/6800013eed53053f9d95c555_Critical%20%20eBook%20-%20Spanish%20Version_compressed.pdf

Böckeler, B., et al. (2024). Uso de IA para el análisis de requisitos: un estudio de caso. <https://www.thoughtworks.com/insights/blog/generative-ai/using-ai-requirements-analysis-case-study>

Cerquozzi, R., et al. (2023). Programa de Estudio de Nivel Básico - ISTQB®. <https://isqi.org/ISTQB-Certified-Tester-Foundation-Level-4.0-CTFL/CT-FL-4.1534>

Cheng, H., et al. (2024). Generative AI for Requirements Engineering: A Systematic Literature Review. <https://arxiv.org/html/2409.06741v1#F3>

Chiranjeevi, S. (2024). Self-Healing Test Automation Framework using AI and ML. <https://pdfs.semanticscholar.org/bd40/928a55eec9cbc0ca8b8d2f685209cdf09244.pdf>

CloudQA. (2025). AI in Testing: The third wave of automation. <https://cloudqa.io/ai-in-testing-the-third-wave-of-automation/>

Cloudflare. (2024). Cómo garantizar prácticas seguras de IA. https://cf-assets.www.cloudflare.com/slt3lc6tev37/3uUO2mTWptwSt4N8Qb5vEu/ebac0d0923884ad559ec3a7eb28ba178/ebook_Ensuring-safe-AI-practices_Spanish-LATAM_20241017.pdf

Cóbe, V., Nonato, Luiza., Novaes, S., Ziebarth, José., (2020). Rumo a uma política de Estado para inteligência artificial. <https://revistas.usp.br/revusp/article/view/167914>

Cognizant. (2020). From Continuous to Autonomous Testing with AI. https://www.cognizant.com/en_us/insights/documents/from-continuous-to-autonomous-testing-with-ai-codex3910.pdf

Comisión Europea (2019). Una definición de la inteligencia artificial: Principales capacidades y disciplinas científicas.

Cortés, H., et al. (2024). Uso de la norma ISO 25010 para establecer requerimientos de calidad en el diseño de un prototipo tecnológico educativo basado en realidad aumentada para la enseñanza de programación básica en estudiantes de educación media. <https://doi.org/10.69639/arandu.v11i2.452>

Cristiá, M. (2024). Introducción a la Ingeniería de Requerimientos. <https://www.fceia.unr.edu.ar/~mcrisia/publicaciones/ingreq-a.pdf>

Cruz, E. (2024). Inteligencia Artificial y Ciberseguridad. https://7a-jornadaciberseguridad.cudi.edu.mx/sites/default/files/imagenes/recursos/Presentaciones/Plenaria_lunes_Inteligencia_Artificial_Ciberseguridad.pdf

Cser, T. (2024). The Power of Generative AI Testing. <https://www.functionize.com/automated-testing/generative-ai-in-software-testing>

Datalytics. (2024). ¿Qué es la IA generativa y cómo funciona? <https://www.datalytics.com/blog/que-es-la-ia-generativa-y-como-funciona/>

Deming, C., et al. (2021). Software Testing in the Era of AI: Leveraging Machine Learning and Automation for Efficient Quality Assurance. <https://ajase.net/article/download/88/97/182>

Digitales. (2024). Libro Blanco de la Inteligencia Artificial Generativa. https://www.coit.es/sites/default/files/digitales_libro_blanco_ia_generativa.pdf

European Medicines Agency. (2024). Guiding principles on the use of large language models in regulatory science and for medicines regulatory activities. https://www.ema.europa.eu/en/documents/other/guiding-principles-use-large-language-models-regulatory-science-medicines-regulatory-activities_en.pdf

Fernández, O. (2024). Inteligencia Artificial y Derecho: Retos y Oportunidades. <https://centrocompetencia.com/wp-content/uploads/2024/08/Inteligencia-Artificial-y-Derecho-Retos-y-Oportunidades.pdf>

Ferrara, E. (2023). Fairness and Bias in Artificial Intelligence: A Brief Survey of Sources, Impacts, and Mitigation Strategies. <https://arxiv.org/pdf/2304.07683>

Gaona, X. (2024). El impacto de la Inteligencia Artificial en la Gestión de Requerimientos de Software.

<https://revistas.udistrital.edu.co/index.php/tia/article/download/22221/20952/150818>

García, B. (2024). Aplicaciones de la Inteligencia Artificial Generativa (IAG) en el Contexto de la Seguridad. UOC (Universitat Oberta de Catalunya)

Geeks for Geeks. (2025). Test Plan - Software Testing. <https://www.geeksforgeeks.org/software-testing/test-plan-software-testing/>

Gil, V., (2025). Agentes de IA: el nuevo salto en inteligencia artificial. <https://ia-espana.org/wp-content/uploads/2025/03/Agentes-Razonamiento-y-Deep-Research.pdf>

Global Market Insights. (2025). Mercado de pruebas de software. <https://www.gminsights.com/es/industry-analysis/software-testing-market>

Goyal, V. (2024). Embracing Generative AI in Software Testing. <https://www.persistent.com/wp-content/uploads/2024/03/whitepaper-embracing-generative-ai-in-software-testing.pdf>

Gómez, G., et al. (1996). Metodología de la investigación cualitativa. Ediciones Aljibe.

Hall, S. (2024). The Impact of Generative AI on Software Testing. <https://isg-one.com/articles/the-impact-of-generative-ai-on-software-testing>

Hemanth, N. (2025). Comparative Review on Automated Test Failure Detection and Healing Tools. <https://www.internationaljournalsrsg.org/IJEEE/2025/Volume12-Issue2/IJEEE-V12I2P113.pdf>

Infovision. (2023). Hyperautomation: A New Era of Testing. <https://www.infovision.com/sites/default/files/2023-03/Hyperautomation%20A%20New%20Era%20of%20Testing.pdf>

ISO. (2013). ISO/IEC/IEEE 29119 – Software and Systems Engineering — Software Testing. <https://wildart.github.io/MISG5020/standards/ISO-IEC-IEEE-29119-3.pdf>

ITConvergence. (2024). How Generative AI Can be used for Software Quality and Testing. <https://www.itconvergence.com/blog/how-generative-ai-can-be-used-for-software-quality-and-testing/>

Jayachandran, J. (2025). The ROI of AI: How Generative AI Slashes Testing Costs. <https://blog.aspiresys.com/testing/the-roi-of-ai-how-generative-ai-slashes-testing-costs>

Jayatilleke, S., et al. (2018). A systematic review of requirements change management. <https://doi.org/10.2298/CSIS161130041J>

Jurkėnas, R. (2025). Understanding the Cost of Software Testing in Software Development: A Complete Breakdown. <https://idealink.tech/blog/understanding-software-testing-costs-development-breakdown>

Katalon. (2024). Benefits of Generative AI in Ensuring Software Quality. <https://katalon.com/resources-center/blog/benefits-generative-ai-software-testing>

King, T., et al. (2019). AI for Testing Today and Tomorrow: Industry Perspectives. <https://davidadamojr.com/wp-content/uploads/2019/05/ai-for-testing-today-and-tomorrow-industry-perspectives.pdf>

Kolhe, A. (2024). The Power of Retrieval Augmented Generation (RAG) in Software Quality Testing. <https://medium.com/generative-ai-101/the-power-of-retrieval-augmented-generation-rag-in-software-quality-testing-4727cd3bc7eb>

Krasner, H. (2022). The Cost of Poor Software Quality in the US: A 2022 Report. <https://www.it-cisq.org/wp-content/uploads/sites/6/2022/11/CPSQ-Report-Nov-22-2.pdf>

Kumar, V. (2025). The Transformative Impact of AI and Machine Learning in Enterprise Software Testing: A Focus on SAP and Salesforce. https://journalwjarr.com/sites/default/files/fulltext_pdf/WJARR-2025-1736.pdf

Laviche, Y. (2018). Test Automation - Automation of What?. <https://ieeexplore.ieee.org/abstract/document/8411740>

Lederer, A., et al. (1995). Causes of inaccurate software development cost estimates. <https://www.sciencedirect.com/science/article/abs/pii/0164121294000922>

Mishra, A. (2024). Generative AI in Automated Software Testing: A Comparative Study. https://www.ijstjournal.com/wp-content/uploads/journal/published_paper/volume-2/issue-1/IJST241006.pdf

Monnette, J. (2024). Gen AI is Non-Deterministic: Why it Matters and How it Changes the Way We Work With Software. <https://www.linkedin.com/pulse/gen-ai-non-deterministic-why-matters-how-changes-way-we-monnette-vrldc/>

Monsalve, C., et al. (2015). Estudio del tamaño de los documentos de requerimientos de software como factor para la estimación del esfuerzo de inspección de requerimientos de software. https://www.researchgate.net/publication/280181535_Estudio_del_tamano_de_los_documentos_de_requerimientos_de_software_como_factor_para_la_estimacion_del_esfuerzo_de_inspeccion_de_requerimientos_de_software

Mudryi, V. (2025). 15 best large language model solutions for your business. <https://intelliarts.com/blog/list-of-large-language-models/>

Muñoz, L., et al. (2020). Las causas del cambio en los requerimientos de software. <https://www.redalyc.org/journal/3783/378365833008/html/>

Navaneethakrishnan, K. (2025). Continuous Testing with AI: Bridging the Gap Between Speed and Quality. <https://www.karthikanav.com/2025/01/20/continuous-testing-with-ai-bridging-the-gap-between-speed-and-quality/>

NetApp. (2024). Almacenamiento de Datos: El Secreto del Éxito de la IA Generativa. <https://www.netapp.com/es/media/117693-na-1134-netapp-ai-infographic.pdf>

NTT Data. (2024). All Hallucinations are Not Bad. Acknowledging Gen AI's Constraints and Benefits. https://www.nttdata.com/global/en-/media/nttdataglobal/1_files/insights/generative-ai/all-hallucinations-are-not-bad---acknowledging-gen-ais-constraints-and-benefits.pdf?rev=11728fac2b124b64920958684908fc9c

PractiTest. (2025). State Of Testing Report 2025. <https://www.practitest.com/state-of-testing/>

QAlified. (2024). Guía Completa para Elaborar un Plan de Pruebas de Software. <https://qalified.com/es/blog/plan-pruebas-software/>

QASource Engineering Team. (2025). Cost vs. ROI: How to Justify ROI on AI Investments in Software Testing. <https://blog.qasource.com/roi-ai-investments-software-testing>

OptiSol Business Solutions. (2025). How GenAI-Powered Automation Testing Is Revolutionizing Enterprise Software Quality. <https://online-optisol.medium.com/how-genai-powered-automation-testing-is-revolutionizing-enterprise-software-quality-15f369612029>

Rautenstrauch, R. (2024). La generación mejorada por recuperación (RAG) explicada. <https://www.consultor365.com/ia/explicacion-rag/>

Reddy, J. (2025). AI-Driven Quality Assurance: Integrating Generative Models, Predictive Analytics, and Self-Healing Frameworks in Software Testing. <https://ejournals.org/ejcsit/wp-content/uploads/sites/21/2025/06/AI-Driven-Quality-Assurance.pdf>

Reuters. (2024). Microsoft, OpenAI plan \$100 billion data-center project, media report says. <https://www.reuters.com/technology/microsoft-openai-planning-100-billion-data-center-project-information-reports-2024-03-29/>

Richards, D. (2024). Unlocking the Power of RAG AI: A Practical Guide to Testing and Fine-Tuning. <https://ragaboutit.com/unlocking-the-power-of-rag-ai-a-practical-guide-to-testing-and-fine-tuning/>

Ríos, G. (2023). Plan de negocios para implementar una consultora de servicios ti especializada en el aseguramiento de la calidad de software. <https://repositorio.uchile.cl/bitstream/handle/2250/197551/Plan-de-negocios-para-implementar-una-consultora-de-servicios-TI-especializada-en-el-aseguramiento-de-la-calidad-de-software.pdf>

- Roberts, L., Jayasena, R., Khanna, S., Arnott, L., Lane, P., Bain, C. (2025). Challenges for implementing generative artificial intelligence (GenAI) into clinical healthcare. <https://onlinelibrary.wiley.com/doi/pdf/10.1111/imj.70035>
- Rothman, M. (2024). AI-Augmented DevOps: The Next Frontier. <https://techstrongresearch.com/wp-content/uploads/2024/02/TS-Tricentis-July22-v2a.pdf>
- Russell, S., et al. (2020). Inteligencia Artificial: Un Enfoque Moderno. Prentice Hall (4ta ed.).
- Sampieri, R., et al. (2014). Metodología de la investigación. McGraw-Hill (6ª ed.).
- Seelamneni, A. (2025). AI in QA: Transforming test automation and software quality through intelligent solutions. https://journalwjaets.com/sites/default/files/fulltext_pdf/WJAETS-2025-0244.pdf
- Shah, V. (2024). Software Testing Life Cycle (STLC): How Apps Get Tested Before Release. <https://www.inexture.com/phases-of-software-testing-life-cycle/>
- S&P Global. (2025). Generative AI: Balancing innovation and privacy. <https://www.vmware.com/docs/451-research-business-impact-brief-gen-ai>
- Svensson, E., et al. (2018). The Future of Software Development: AI-Driven Testing and Continuous Integration for Enhanced Reliability. <https://www.ijtsrd.com/computer-science/other/14427/the-future-of-software-development-aidriven-testing-and-continuous-integration-for-enhanced-reliability/dr-erik-svensson>
- TestRigor. (2024). Generative AI in Software Testing. <https://testrigor.com/generative-ai-in-software-testing/#ethicalConsiderations>
- TestBytes. (2023). Strategies for Testing in AI Applications Best Practices and Key Considerations. <https://www.testbytes.net/blog/strategies-for-testing-ai-applications/>
- The Business Research Company. (2025). Software Testing Global Market Report 2025. <https://www.thebusinessresearchcompany.com/report/software-testing-global-market-report>
- Tian, Z., et al. (2020). Software Defect Prediction based on Machine Learning Algorithms. <https://ieeexplore.ieee.org/abstract/document/9064412>
- Tishkoff PLC. (2025). Privacy and Legal Risks of Generative AI for Businesses. <https://tish.law/wp-content/uploads/2025/02/Generative-AI-and-the-Law-for-Businesses-eBook-min.pdf>
- Trendig. (2024). Artificial intelligence in the everyday life of software testers: Efficient testing thanks to AI. <https://trendig.com/en/blog/ai-in-softwaretesting-for-efficient-testing/>

Tricentis. (2025). 2025 Quality Transformation Report. <https://be.tricentis.com/media-assets/pdf/Tricentis-report-Tricentis-2025-quality-transformation-report.pdf>

Van Hoang, T. (2025). Can we use generative AI to generate test cases from user stories?. <https://www.thoughtworks.com/insights/blog/generative-ai/can-we-use-generative-ai-to-generate-test-cases-from-user-stories>

Vinayagam, R. (2024). Testing Transformed: How Generative AI is Supercharging Software Testing. <https://contact-rajeshvinayagam.medium.com/testing-transformed-how-generative-ai-is-supercharging-software-quality-6cdec8ab8d6d>

Visure. (2025). ¿Qué es la ingeniería de requisitos? Proceso para software y sistemas. <https://visuresolutions.com/es/gu%C3%ADa-de-limosna/ingenier%C3%ADa-de-requisitos>

Williams, B., et al. (2006). Change Risk Assessment: Understanding Risks Involved in Changing Software Requirements. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=54cea51b6ac6eb610a104990b36a53a77dca46b7>

Yi, M. (2023). The Ethical Challenges of Generative AI Applications. <https://yulleyi.medium.com/the-ethical-challenges-of-generative-ai-applications-8478ecdf2a4>

Zunzunegui, A., (2025). Guía completa sobre agentes de Inteligencia Artificial: qué son, cómo funcionan y cómo usarlos. <https://acumbamail.com/blog/agentes-ia/>

Zymr. (2024). The AI-Driven Software Testing Services Handbook: Strategies, Challenges, and Predictions. [https://cdn.prod.website-files.com/6523ed2d670117e5922bd1d3/664f1f97743809dc15e36c3f_EBook-Zymr-2024%20\(1\)-compressed.pdf](https://cdn.prod.website-files.com/6523ed2d670117e5922bd1d3/664f1f97743809dc15e36c3f_EBook-Zymr-2024%20(1)-compressed.pdf)