

EL REEMPLAZO DE LAS INTELIGENCIAS ARTIFICIALES A LOS ANALISTAS DESARROLLADORES

Leonardo André Farias Marcano
Estudiante de Ingeniería de Sistemas
Universidad Bicentennial de Aragua, Venezuela
leonardoafarias@gmail.com

Resumen

El avance de la inteligencia artificial plantea debates sobre la continuidad laboral de los profesionales tecnológicos en el área de la informática. Este artículo analiza la probabilidad de sustitución integral de los analistas desarrolladores por modelos generativos comerciales. Mediante una metodología exploratoria y analítica de fuentes documentales, se contrastan las capacidades automatizadas frente a los requerimientos de la ingeniería de software. Los resultados indican que herramientas de vanguardia como SWE Bench, Devin AI y Cosine Genie optimizan tareas de codificación e ingeniería inversa, pero muestran deficiencias críticas ante la ambigüedad conceptual. El componente humano permanece indispensable en las etapas de levantamiento de información, diseño arquitectónico macro y traducción de reglas de negocio complejas. Se concluye que la inteligencia artificial operará como un elemento complementario y catalizador de productividad. La transición sectorial forzará la evolución de los desarrolladores hacia roles de supervisión estratégica, arquitectura híbrida y gestión de seguridad de datos.

Palabras clave: Inteligencia Artificial, Analistas Desarrolladores, Componente humano.

THE REPLACEMENT OF DEVELOPER ANALYSTS BY ARTIFICIAL INTELLIGENCE

Abstract

The advancement of artificial intelligence raises debates about the labor continuity of technological professionals in the computing field. This article analyzes the probability of complete replacement of systems development analysts by commercial generative models. Through an exploratory and analytical methodology of documentary sources, automated capabilities are contrasted against software engineering requirements. The results indicate that cutting-edge tools such as SWE Bench, Devin AI, and Cosine Genie optimize coding and reverse engineering tasks, but show critical deficiencies when facing conceptual ambiguity. The human component remains indispensable in the stages of requirements gathering, macro architectural design, and translation of complex business rules. It is concluded that artificial intelligence will operate as a

complementary element and a productivity catalyst. The sectoral transition will force the evolution of developers towards roles of strategic supervision, hybrid architecture, and data security management.

Keywords: Artificial Intelligence, Developer Analyst, Human Component.

LE REMPLACEMENT DES ANALYSTES-DÉVELOPPEURS PAR LES INTELLIGENCES ARTIFICIELLES

Résumé

L'avancement de l'intelligence artificielle suscite des débats sur la continuité de l'emploi des professionnels de la technologie dans le domaine informatique. Cet article analyse la probabilité de remplacement complet des analystes développeurs de systèmes par des modèles génératifs commerciaux. À travers une méthodologie exploratoire et analytique des sources documentaires, les capacités automatisées sont contrastées avec les exigences du génie logiciel. Les résultats indiquent que des outils de pointe tels que SWE Bench, Devin AI et Cosine Genie optimisent les tâches de codage et d'ingénierie inverse, mais affichent des faiblesses critiques face à l'ambiguïté conceptuelle. Le composant humain reste indispensable dans les étapes de collecte des besoins, de conception architecturale macro et de traduction des règles de gestion complexes. Il est conclu que l'intelligence artificielle fonctionnera comme un élément complémentaire et un catalyseur de productivité. La transition sectorielle contraindra l'évolution des développeurs vers des rôles de supervision stratégique, d'architecture hybride et de gestion de la sécurité des données.

Mots-clés: Intelligence Artificielle, Analyste Développeur, Composant Humain

Introducción

La evolución tecnológica ha generado un debate persistente sobre el impacto de la automatización en diversas profesiones, incluyendo el desarrollo de software. En este contexto, surge la interrogante sobre la necesidad de conservar analistas desarrolladores humanos en un futuro dominado por sistemas de inteligencia artificial avanzados. A medida que nuevas herramientas y modelos automatizados emergen en el mercado global, resulta fundamental examinar hasta qué punto pueden reemplazar el trabajo humano en la programación y el diseño arquitectónico de soluciones digitales.

El presente artículo tiene el propósito de analizar la relación operativa entre la inteligencia artificial y el desarrollo de software, explorando las capacidades actuales de los modelos y su potencial real para asumir tareas tradicionalmente ejecutadas por analistas desarrolladores. Asimismo, se busca comprender si estos profesionales poseen la capacidad de adaptarse a nuevas funciones dentro de un entorno tecnológico en constante mutación.

La metodología empleada cuenta con un enfoque analítico y exploratorio, basado en el estudio sistemático de herramientas y modelos de inteligencia artificial aplicados directamente al desarrollo de software. Se consideran ejemplos prácticos y teorías asociadas con la programación asistida, complementados con la postura técnica de expertos en la materia. Además, se evalúa el papel del componente humano en la toma de decisiones y el diseño de soluciones tecnológicas de nivel macro.

En las secciones posteriores se abordan temas como la clasificación de las inteligencias artificiales, sus aplicaciones específicas en la ingeniería de software, las limitaciones estructurales que presentan en la actualidad y los desafíos que implica su implementación en el ámbito profesional. De igual forma, se discute la integridad de los datos y la seguridad de la información en entornos donde las herramientas automatizadas asumen un rol activo.

Revisión Teórica

SWE Bench y su enfoque en la corrección de errores de código

Un conjunto de datos o dataset constituye una colección organizada de datos utilizable para el análisis o procesamiento automatizado. Estructuralmente, estas colecciones suelen ordenarse en matrices, tablas o formatos estructurados específicos como JSON. En términos generales, estos entornos de datos sirven de

insumo para el aprendizaje automático (machine learning), con el objetivo de entrenar, validar y probar modelos en fase de desarrollo.

Bajo esta premisa, SWE Bench representa un entorno de evaluación especializado utilizado para medir el porcentaje de problemas reales de ingeniería de software que un modelo puede resolver de forma autónoma. Dichos problemas se extraen de repositorios de código abierto de la plataforma GitHub. El flujo metodológico de este evaluador inicia con la clonación del repositorio de software, permitiendo que el sistema se nutra del contexto técnico necesario para comprender la estructura del código base.

Posteriormente, el modelo analiza los archivos publicados y aplica algoritmos lógicos para reparar los errores identificados. Tras registrar cada acción ejecutada, el sistema genera un archivo de parche, comúnmente denominado predicción. Este parche se almacena junto con datos clave como el identificador único y el nombre del repositorio, asegurando la trazabilidad de los arreglos efectuados. Si bien este evaluador opera con modelos avanzados como GPT-4, GPT-3.5, Code Llama y Claude, todos requieren ser probados sobre código preexistente creado por humanos, cuyas soluciones demandan supervisión e implementación cautelosa.

Devin IA y Cosine Genie: nuevas herramientas para la automatización del desarrollo

Los modelos de lenguaje de propósito general resuelven problemas diversos, pero carecen de la especialización requerida para el desarrollo de software complejo. Por esta razón, la industria tecnológica ha diseñado agentes orientados exclusivamente a la programación. En este segmento destacan herramientas como Devin AI y Cosine Genie, las cuales han reactivado la discusión sobre el desplazamiento de la mano de obra humana en el sector informático.

Estas tecnologías fueron concebidas con el propósito de automatizar las funciones del desarrollador, registrando índices de resolución de problemas del 13.86% en el caso de Devin y del 30.08% en Genie en entornos de prueba controlados. A pesar del crecimiento proyectado de estos porcentajes, la sustitución total resulta improbable mientras los requerimientos operativos y las necesidades de negocio sigan dependiendo de las ideas de un ser humano. La abstracción y la experiencia humana resultan vitales para configurar propuestas comerciales que no se encuentran del todo definidas en una fase inicial.

Diferencias entre IA débil e IA fuerte

El desarrollo tecnológico actual se encuentra delimitado por la distinción entre capacidades de inteligencia artificial. La mayoría de las herramientas comerciales vigentes corresponden a modelos de lenguaje grandes (LLM), alimentados por volúmenes masivos de texto no clasificado bajo metodologías supervisadas o no supervisadas. Al no contar con filtros semánticos perfectos, estos sistemas presentan inconsistencias y sesgos en sus datos de salida.

Este grupo pertenece a la denominada inteligencia artificial débil o estrecha. Dichos sistemas resultan finitos y se limitan exclusivamente a su campo específico de entrenamiento, operando como imitaciones de las capacidades cognitivas humanas sin poseer conciencia real. Ejemplos de esta categoría incluyen a asistentes como Alexa de Google, Siri de Apple y plataformas de chat como GPT.

Por otro lado, la inteligencia artificial fuerte o general evoca sistemas con la capacidad teórica de comprender, razonar y aprender de forma similar al ser humano, igualando o superando la cognición promedio. Aunque herramientas como Cosine Genie y Devin AI se aproximan conceptualmente a esta clasificación debido a su nivel de autonomía, continúan requiriendo la supervisión directa de un desarrollador. Estos agentes no son capaces de generar de forma independiente la arquitectura completa de un software basándose únicamente en reglas de negocio ambiguas.

Metodología

El presente estudio se desarrolló bajo un enfoque cualitativo y documental de carácter exploratorio, centrado en el análisis del ciclo de vida del desarrollo de software frente a las capacidades técnicas de la inteligencia artificial contemporánea. Para estructurar la comparación, se desglosó el flujo metodológico estándar que rige la actividad de un analista desarrollador, compuesto por seis fases sucesivas:

Análisis del problema: Fase orientada a comprender exhaustivamente los requisitos del proyecto a partir de las necesidades u objetivos del negocio.

Diseño de la solución: Etapa de planificación de la arquitectura macro del software, delimitando la interacción y estructura de los componentes lógicos.

Codificación: Proceso técnico de escritura del código fuente en lenguajes de programación específicos, implementando los requerimientos analizados previamente.

Pruebas: Ejecución de rutinas de control para verificar el correcto funcionamiento del sistema y el cumplimiento de las especificaciones iniciales.

Implementación: Puesta en marcha de la solución tecnológica dentro del entorno de producción de la organización.

Mantenimiento y mejora continua: Actualización periódica del software para adaptar la plataforma a nuevas necesidades operativas o corregir fallas imprevistas.

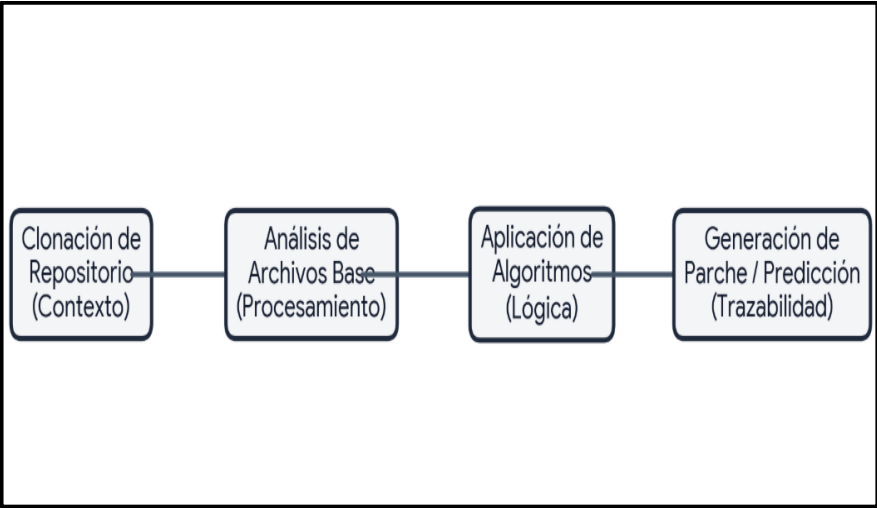
A partir de esta estructura, la investigación analizó la capacidad de trazabilidad y cobertura de las herramientas de inteligencia artificial en entornos macros, evaluando las barreras de comunicación, abstracción e interpretación semántica que limitan el rendimiento de los algoritmos en comparación con el desempeño humano.

Resultados

Los hallazgos reflejan que, si bien las inteligencias artificiales débiles y fuertes demuestran aptitud para la codificación aislada, no logran mantener la trazabilidad integral de un sistema macro de manera autónoma. El análisis de los flujos de trabajo evidencia que escribir código representa únicamente una fracción del ciclo técnico.

Para ilustrar la complejidad que precede a la codificación, la Figura 1 sintetiza la estructura del entorno de evaluación automatizada y su dependencia del contexto preexistente.

Figura 1.
Flujo abstracto del evaluador SWE Bench en repositorios de código abierto

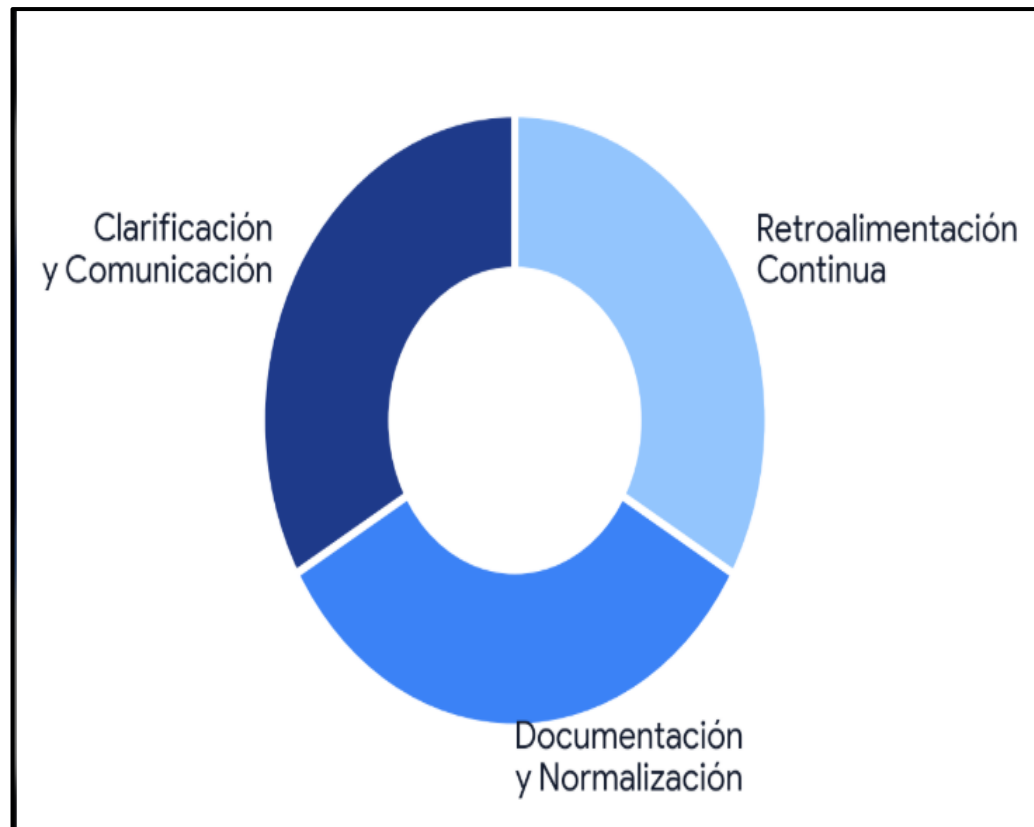


Fuente: Farias (2026)

La automatización de funciones aisladas no equipara la capacidad de mediación técnica. La Figura 2 expone las dimensiones de la interacción humana indispensables para la correcta interpretación de necesidades corporativas.

Figura 2.

Dimensiones del componente humano en la ingeniería de requerimientos

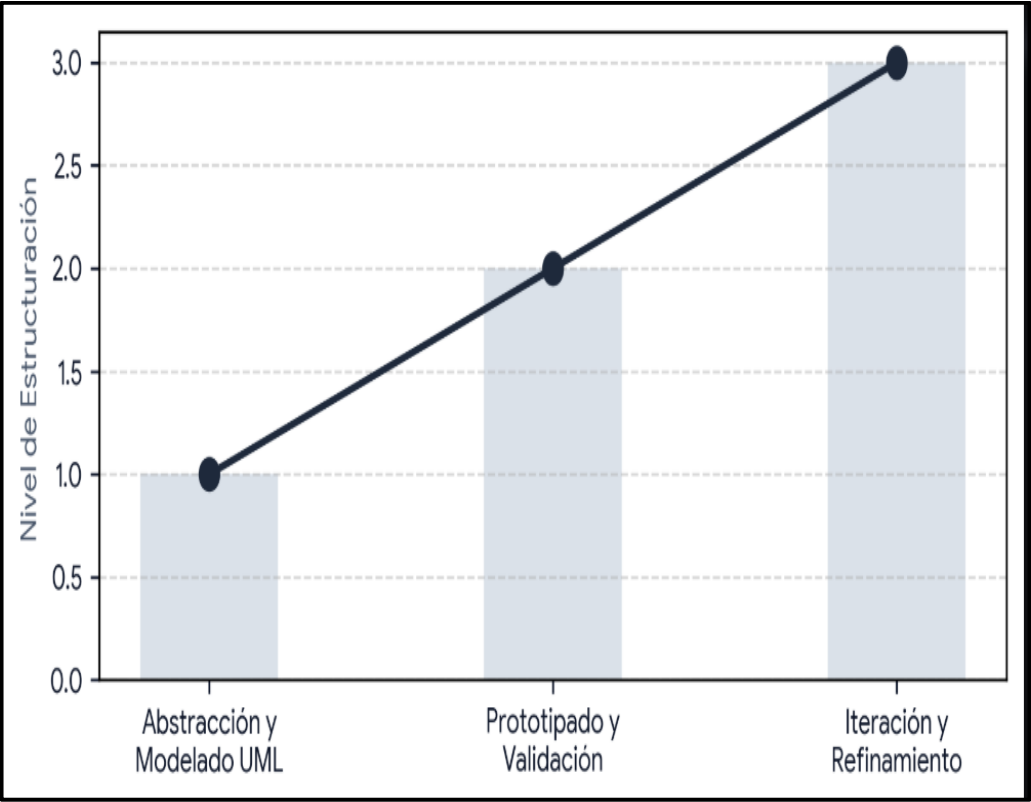


Fuente: Farias (2026)

Asimismo, el proceso de traducción de ideas difusas hacia marcos técnicos formales demanda herramientas de modelado que superan la generación de texto plano. La Figura 3 detalla los instrumentos de abstracción empleados por el analista desarrollador antes de la construcción del código fuente.

Figura 3.

Fases de la transformación de conceptos abstractos en esquemas técnicos ejecutables

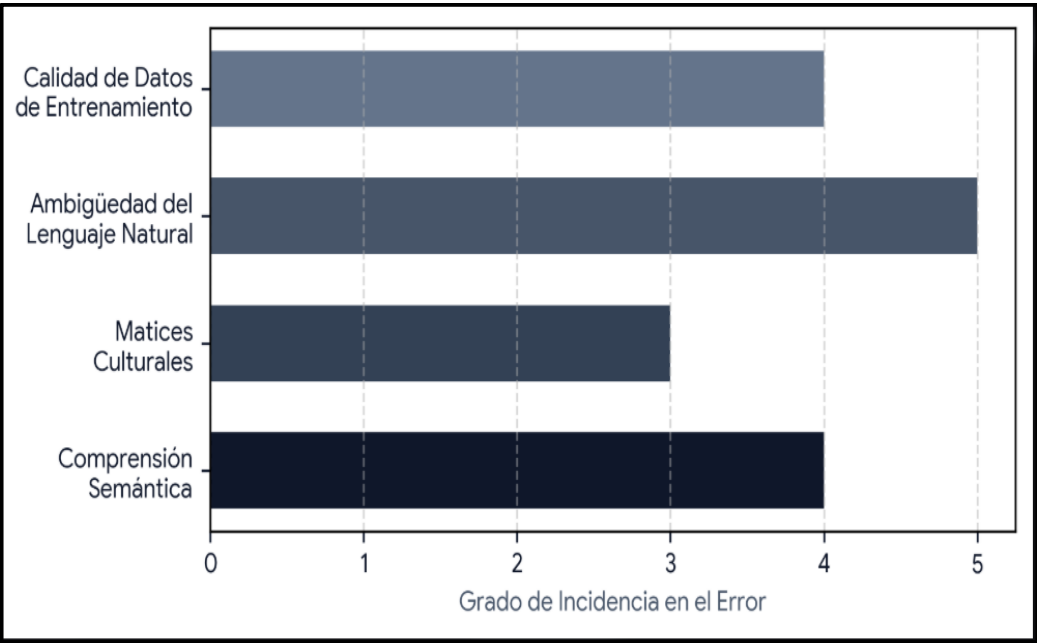


Fuente: Farias (2026)

Finalmente, se determinó que la eficiencia técnica de las inteligencias artificiales se encuentra restringida por barreras semánticas insalvables para los modelos matemáticos actuales. La Figura 4 sistematiza las limitaciones contextuales que condicionan la autonomía de las herramientas de software asistido.

Figura 4.

Factores limitantes de la inteligencia artificial en la comprensión del lenguaje natural



Fuente: Farias (2026)

Discusión

Los datos recopilados ponen de manifiesto que el uso del pseudocódigo y los diagramas de flujo trasciende la mera organización técnica; actúan como canales de comunicación esenciales para traducir la complejidad del software a un lenguaje comprensible para los usuarios finales. Aunque los modelos de inteligencia artificial explican sus líneas de código de forma rudimentaria, carecen de la experiencia de campo para adaptar dichas explicaciones a personas sin trasfondo informático. La presencia del analista humano es indispensable para actuar como puente comunicacional entre los interesados y el equipo técnico. Este rol implica clarificar requerimientos mediante técnicas interpersonales como entrevistas y talleres, extrayendo detalles implícitos que los clientes no logran verbalizar con precisión.

La normalización y documentación estructurada mediante diagramas UML y casos de uso reduce drásticamente las interpretaciones erróneas en las fases avanzadas del proyecto. Las herramientas lógicas actuales procesan texto y detectan patrones a gran velocidad, pero no comprenden matices culturales, emocionales o de contexto empresarial. Esta incapacidad para manejar la ambigüedad inherente al lenguaje natural humano expone a los sistemas automatizados a errores semánticos graves si operan sin supervisión. Por consiguiente, la transformación de ideas nebulosas en módulos técnicos y prototipos iterativos requiere un nivel de juicio crítico y refinamiento arquitectónico que los algoritmos estadísticos no pueden replicar, consolidando al ser humano como el eje rector de la integridad del software.

Conclusiones

El debate en torno a la sustitución de los analistas desarrolladores por sistemas de inteligencia artificial evidencia una transformación profunda en la industria del software, pero a menudo ignora la naturaleza estratégica de la profesión. El análisis de plataformas de última generación como SWE Bench, Devin AI y Cosine Genie demuestra que las capacidades de automatización actuales se concentran principalmente en la etapa de codificación, mostrando deficiencias severas en las fases de análisis, diseño iterativo y validación contextual frente a reglas de negocio dinámicas. Las limitaciones de los modelos lingüísticos ante la ambigüedad del lenguaje natural y la necesidad de resguardar la seguridad y privacidad de datos corporativos confidenciales impiden que el rol del desarrollador sea prescindible en el corto y mediano plazo.

La inteligencia artificial no debe percibirse como una amenaza de reemplazo total, sino como una herramienta complementaria de alta eficiencia que actuará como catalizador del rendimiento técnico. El perfil profesional del analista desarrollador no desaparecerá; se encuentra en un proceso de evolución hacia estructuras híbridas donde el pensamiento crítico, la capacidad de abstracción y la mediación

interpersonal serán las competencias más valoradas. Los especialistas del sector migrarán de la escritura mecánica de código hacia funciones estratégicas de mayor valor, tales como la arquitectura de sistemas complejos, la optimización de infraestructuras críticas y el control ético de los algoritmos, reafirmando el papel del ser humano como componente indispensable en el ciclo de vida del software.

Referencias

- Arbeláez-Campillo, D. F. (2021). Inteligencia artificial y condición humana: ¿Entidades contrapuestas o fuerzas complementarias? Redalyc, 27(2), 510-521. redalyc.org
- Corvalán, J. G. (2018). Inteligencia artificial: retos, desafíos y oportunidades - Prometea: la primera inteligencia artificial de Latinoamérica al servicio de la Justicia. Revista de Internet, Derecho y Política, (26), 1-15. scielo.br
- Galván, P. (2018). Impacto de la Inteligencia Artificial en el Desarrollo de Software. Software Guru, (56), 24-28. sg.com.mx