Ciencia y Educación
(L-ISSN: 2790-8402 E-ISSN: 2707-3378)
Vol. 7 No. 8.1
Edición Especial VIII 2026

DEUDA TÉCNICA, SEGURIDAD Y REVISIÓN HUMANA EN CONTRIBUCIONES PYTHON GENERADAS POR AGENTES DE INTELIGENCIA ARTIFICIAL

TECHNICAL DEBT, SECURITY, AND HUMAN REVIEW IN PYTHON CONTRIBUTIONS GENERATED BY ARTIFICIAL INTELLIGENCE AGENTS

Autores: [1]Diego Marcelo Reina Haro, [2]Andrea Fernanda Criollo Cantos, [3]Carlos Miguel Campoverde Santos y [4]Estalin Fabián Mejía Hidalgo.
[1]ORCID ID: https://orcid.org/0000-0002-7757-6919
[2]ORCID ID: https://orcid.org/0009-0007-8869-3604
[3]ORCID ID: https://orcid.org/0009-0006-1615-4065
[4]ORCID ID: https://orcid.org/0009-0006-0215-2237
[1]E-mail de contacto: dreina@unach.edu.ec
[2]E-mail de contacto: fernanda.criollo@unach.edu.ec
[3]E-mail de contacto: miguel.campoverde@unach.edu.ec
[4]E-mail de contacto: estalin.mejia@unach.edu.ec
Afiliación: [1*2*3*4*]Universidad Nacional de Chimborazo, (Ecuador).
Artículo recibido: 28 de Julio del 2026.
Artículo revisado: 30 de Julio del 2026.
Artículo aprobado: 30 de Julio del 2026.
[1]Ingeniero en Sistemas Informáticos de la Escuela Superior Politécnica de Chimborazo, (Ecuador). Magíster en Informática Aplicada de la Escuela Superior Politécnica de Chimborazo, (Ecuador).
[2]Ingeniera en Tecnologías de la Información de la Universidad Nacional de Chimborazo, (Ecuador).
[3]Ingeniero en Tecnologías de la Información de la Universidad Nacional de Chimborazo, (Ecuador). Máster Universitario en Inteligencia Artificial de la Universidad Internacional de La Rioja, (España).
[4]Ingeniero en Biotecnología Ambiental de la Escuela Superior Politécnica de Chimborazo, (Ecuador). Máster Universitario en Ingeniería Hidráulica y Medio Ambiente de la Universidad Politécnica de Valencia, (España). Magíster en Matemática Aplicada de la Universidad del Azuay, (Ecuador).

Resumen

El estudio comparó la deuda técnica, las posibles alertas de seguridad y los patrones de revisión humana presentes en contribuciones de código Python generadas por agentes de inteligencia artificial y por desarrolladores humanos en proyectos de código abierto. Para ello, se adoptó un enfoque cuantitativo, observacional y retrospectivo, sustentado en datos procedentes de AIDev y GitHub. Se analizaron 200 contribuciones realizadas por agentes y 200 contribuciones humanas, emparejadas según el repositorio, el tipo de tarea y el alcance de la modificación. Además, se controlaron variables como la temporalidad, la cantidad de líneas modificadas, el número de archivos intervenidos y los commits efectuados. Los cambios producidos antes y después de cada contribución se evaluaron mediante las herramientas Ruff, Radon y Semgrep, diferenciando las revisiones efectuadas por personas de las intervenciones automáticas. Los resultados no evidenciaron diferencias estadísticamente significativas en la densidad de nuevos hallazgos relacionados con la mantenibilidad, la variación del índice de mantenibilidad, la complejidad ciclomática ni las alertas de seguridad validadas. Sin embargo, las contribuciones generadas por agentes recibieron revisión humana con menor frecuencia que las realizadas por desarrolladores humanos, con porcentajes de 36,5 % y 67,5 %, respectivamente. Asimismo, las contribuciones de agentes presentaron una menor tasa de fusión, correspondiente al 52,0 %, frente al 83,5 % registrado en las contribuciones humanas. En los casos en que ambos integrantes del par fueron sometidos a revisión, no se identificaron diferencias relevantes en la cantidad de mensajes, revisores, rondas de revisión ni tiempos empleados. Los modelos ajustados confirmaron que las contribuciones generadas por agentes tenían una menor probabilidad de ser revisadas y fusionadas, sin que las métricas técnicas permitieran explicar esta diferencia. En consecuencia, se concluyó que el principal

contraste no se relaciona con un deterioro técnico detectable del código, sino con la existencia de patrones diferenciados de supervisión e integración, los cuales varían según el agente y el repositorio analizado.
Palabras clave: Inteligencia artificial, Deuda técnica, Revisión de código, Seguridad de software, Python, GitHub, Ingeniería de software.

Abstract

The study compared technical debt, potential security alerts, and human review patterns in Python code contributions generated by AI agents and human developers in open-source projects. A quantitative, observational, and retrospective approach was adopted, based on data from AIDev and GitHub. Two hundred agent-generated and two hundred human-generated contributions were analyzed, matched by repository, task type, and scope of modification. Variables such as timing, number of modified lines, number of files affected, and commits were also controlled. Changes before and after each contribution were evaluated using Ruff, Radon, and Semgrep, differentiating between human and automated revisions. The results showed no statistically significant differences in the density of new maintainability findings, maintainability index variation, cyclomatic complexity, or validated security alerts. However, agent-generated contributions received human review less frequently than those made by human developers, with percentages of 36.5% and 67.5%, respectively. Likewise, agent contributions showed a lower merge rate, at 52.0%, compared to 83.5% for human contributions. In cases where both members of the pair were reviewed, no relevant differences were identified in the number of messages, reviewers, review rounds, or time spent. The adjusted models confirmed that agent-generated contributions were less likely to be reviewed and merged, and technical metrics could not explain this difference. Consequently, it was concluded that the main contrast is not related to detectable technical deterioration of the code, but rather to the existence of differentiated monitoring and integration patterns, which vary depending on the agent and the repository analyzed.
Keywords: Artificial intelligence, Technical debt, Code review, Software security, Python, GitHub, Software engineering.

Sumário

O estudo comparou a dívida técnica, os potenciais alertas de segurança e os padrões de revisão humana em contribuições de código Python geradas por agentes de IA e desenvolvedores humanos em projetos de código aberto. Uma abordagem quantitativa, observacional e retrospectiva foi adotada, baseada em dados do AIDev e do GitHub. Duzentas contribuições geradas por agentes e duzentas contribuições geradas por humanos foram analisadas, pareadas por repositório, tipo de tarefa e escopo de modificação. Variáveis como tempo, número de linhas modificadas, número de arquivos afetados e commits também foram controladas. As alterações antes e depois de cada contribuição foram avaliadas usando Ruff, Radon e Semgrep, diferenciando entre revisões humanas e automatizadas. Os resultados não mostraram diferenças estatisticamente significativas na densidade de novas descobertas de manutenibilidade, na variação do índice de manutenibilidade, na complexidade ciclomática ou nos alertas de segurança validados. No entanto, as contribuições geradas por agentes receberam revisão humana com menos frequência do que as feitas por desenvolvedores humanos, com percentuais de 36,5% e 67,5%, respectivamente. Da mesma forma, as contribuições de agentes apresentaram uma taxa de merge menor, de 52,0%, em comparação com 83,5% para as contribuições humanas. Nos casos em que ambos os membros do par foram revisados, não foram identificadas diferenças relevantes no número de mensagens, revisores, rodadas de revisão ou tempo gasto. Os modelos ajustados confirmaram que as contribuições geradas pelo agente tinham menor probabilidade de serem revisadas e integradas, e as métricas técnicas não conseguiram explicar essa diferença. Consequentemente, concluiu-se

que o principal contraste não está relacionado à deterioração técnica detectável do código, mas sim à existência de padrões diferenciados de monitoramento e integração, que variam dependendo do agente e do repositório analisados.
Palavras-chave: Inteligência artificial, Dívida técnica, Revisão de código, Segurança de software, Python, GitHub, Engenharia de software.

Introducción

Los agentes de programación basados en modelos de lenguaje han dejado de operar únicamente como sistemas de autocompletado. Ahora pueden interpretar una incidencia, modificar varios archivos, ejecutar pruebas, abrir una pull request y responder a observaciones con una autonomía creciente. AIDev documenta esta transición a escala de repositorios reales y reúne cientos de miles de contribuciones atribuidas a Codex, Devin, GitHub Copilot, Cursor y Claude Code (Li et al., 2026). Esta expansión convierte al agente en un participante visible del proceso de integración, no solo en una herramienta privada del desarrollador.

Sin embargo, una pull request no se evalúa exclusivamente por la corrección aparente del código. Los mantenedores valoran su alcance, coherencia con la arquitectura, pruebas, seguridad, facilidad de revisión y ajuste a normas locales. La revisión moderna cumple funciones de detección de defectos, transferencia de conocimiento, coordinación y control de cambios (McIntosh et al., 2016; Sadowski et al., 2018). Por ello, medir líneas modificadas o tasas de fusión sin reconstruir la interacción de revisión ofrece una lectura incompleta de la incorporación de agentes en proyectos abiertos. La evidencia previa sobre calidad y seguridad de código generado por inteligencia artificial tampoco es uniforme. Los estudios controlados y centrados en fragmentos han identificado recomendaciones inseguras y una confianza excesiva de los usuarios en los asistentes (Pearce et al., 2022; Perry et al., 2023). No obstante, esos escenarios no reproducen por completo una contribución real, donde intervienen convenciones del repositorio, pruebas, herramientas automáticas y decisiones de mantenedores.

En sentido inverso, una tasa elevada de fusión tampoco prueba calidad: puede reflejar tareas simples, revisión automática o flujos de integración específicos. Los trabajos recientes sobre agentes en GitHub han avanzado desde la caracterización general hacia la comparación de tamaños, mensajes, mantenimiento y procesos de revisión. Ogenrwot y Businge (2026) mostraron que las contribuciones de agentes difieren de las humanas en estructura y commits; Sawada et al. (2026) observaron perfiles de mantenimiento posteriores distintos; Watanabe et al. (2026) identificaron variación entre agentes en la presentación de las pull requests y en la respuesta de revisores. Duma et al. (2026) advirtieron, además, que una proporción importante de la actividad registrada como revisión proviene de sistemas automáticos. Estos resultados obligan a separar la supervisión humana de la interacción entre bots.

Pese a esos avances, persisten tres problemas. Primero, muchos análisis comparan grupos agregados que difieren en repositorio, tarea y tamaño. Segundo, suelen medir el estado final del código y pueden atribuir a la contribución problemas que ya existían. Tercero, tratan la revisión como un recuento único, sin distinguir la probabilidad de recibir supervisión de la intensidad del trabajo una vez iniciada. Esa distinción es crítica: una contribución con cero comentarios puede haber sido aceptada sin

fricción, descartada tempranamente o evaluada fuera de GitHub. Este estudio afronta esos problemas mediante una comparación emparejada dentro del mismo repositorio.

Se analizaron las contribuciones Python generadas por Codex, Devin y GitHub Copilot frente a contribuciones humanas sin huella observable de agente. Reconstruimos el cambio exacto antes y después de cada pull request, medimos indicadores de mantenibilidad, complejidad y seguridad, y clasificamos por separado las intervenciones humanas y automáticas. Planteamos cuatro preguntas: ¿difieren ambos grupos en deuda técnica y mantenibilidad?, ¿introducen alertas de seguridad distintas?, ¿cambian la recepción y la intensidad de la revisión humana?, y ¿se asocia el origen con la fusión después de considerar el tamaño y las métricas técnicas?

Materiales y Métodos

Se aplico un diseño cuantitativo, observacional, retrospectivo y comparativo de cohortes emparejadas. La unidad de análisis fue la pull request. Utilizamos AIDev como marco de identificación de contribuciones generadas por agentes y como fuente inicial de metadatos. Complementamos cada registro mediante la interfaz de programación de GitHub para recuperar estado, commits, archivos, revisiones formales, comentarios en línea, comentarios generales y fechas de apertura, cierre y fusión. El corte temporal de AIDev correspondió al 1 de agosto de 2025. Restringimos la población a repositorios públicos con más de 500 estrellas, porque el subconjunto humano de AIDev procede de ese universo. Seleccionamos proyectos cuyo lenguaje principal fuera Python o cuyo cambio estuviera dominado por archivos Python. Excluimos pull requests abiertas, cambios exclusivamente documentales o configuracionales, actualizaciones automáticas de dependencias, archivos generados, estados de código no reconstruibles y casos con datos esenciales contradictorios. El grupo de agentes incluyó contribuciones atribuidas a OpenAI Codex, Devin y GitHub Copilot. Verificamos la atribución mediante la cuenta autora, la autoría observable de commits y los metadatos de AIDev. Denominamos al segundo grupo "contribuciones humanas sin huella observable de agente", porque los datos públicos no permiten descartar el uso privado de asistentes por parte de una persona.

La depuración produjo 2.853 candidatas iniciales. Enriquecimos 2.158 registros y obtuvimos 1.145 contribuciones técnicamente elegibles: 775 de agentes y 370 humanas. Emparejamos sin reemplazo dentro del mismo repositorio y tipo de tarea, mantuvimos el mismo alcance general, código de producción o pruebas, y minimizamos las diferencias de fecha, líneas Python modificadas, archivos y commits. Antes de analizar el código, congelamos 200 pares principales. Limitamos la concentración a un máximo de 20% de pares por repositorio y 50% por agente. Conservamos 150 pares con criterios temporales y de tamaño más estrictos y 84 pares con atribución conservadora para análisis de sensibilidad.

Para evitar atribuir cambios ajenos, calculamos el ancestro común con git merge-base y analizamos únicamente los archivos Python que GitHub registró en la pull request. Conservamos los estados anterior y final de cada archivo, incluidos añadidos, eliminados y renombrados. La unión de la muestra principal y las sensibilidades comprendió 439 pull requests únicas. Ruff 0.16.0 identificó hallazgos nuevos de mantenibilidad y fiabilidad mediante un conjunto fijado de reglas sobre complejidad, ramas, argumentos, anidamiento, nombres no definidos, valores mutables y patrones

simplificables. Radon 6.0.1 calculó complejidad ciclomática, bloques con complejidad igual o superior a 11 e índice de mantenibilidad. Expresamos los hallazgos nuevos de Ruff por cada 100 líneas Python modificadas y calculamos el cambio entre el estado final y el anterior.

Semgrep Community Edition 1.172.0 evaluó alertas potenciales de seguridad con el paquete de reglas Python congelado antes del análisis. Revisamos contextualmente cada alerta nueva. Solo clasificamos un hallazgo como alerta válida cuando el patrón correspondía a un uso con implicaciones de seguridad y el contexto del código respaldaba esa interpretación. Clasificamos como humanas las cuentas de tipo User que no correspondían al autor, a un agente, a un bot de revisión, a integración continua, a seguridad, a dependencias o a una cuenta de servicio. La medida estricta incluyó revisiones formales y comentarios en línea sobre el diff. Se validaron manualmente 248 comentarios generales considerados candidatos. De este total, 81 correspondieron a intervenciones humanas técnicamente sustantivas, 132 procedieron de automatizaciones o agentes y 35 fueron clasificados como mensajes administrativos o no evaluativos. Los 81 comentarios sustantivos se incorporaron únicamente en una medida ampliada de sensibilidad.

Se distinguieron dos componentes del proceso de revisión. La recepción de revisión indicó si existió al menos una intervención humana observable, mientras que la intensidad se evaluó mediante el número de eventos, mensajes, revisores únicos, solicitudes de cambios, commits posteriores, rondas de revisión-corrección y tiempos asociados al proceso. La intensidad se analizó de manera condicional en aquellos pares en los que ambos miembros recibieron revisión, con el propósito de evitar que el exceso de valores cero confundiera la ausencia de supervisión con un bajo nivel de esfuerzo revisor.

Las variables continuas se describieron mediante medianas y rangos intercuartílicos. Para los resultados cuantitativos emparejados se aplicó la prueba de rangos con signo de Wilcoxon, mientras que para los resultados binarios se utilizó la prueba exacta de McNemar. Los intervalos de confianza del 95 % se estimaron mediante 10.000 remuestreos bootstrap y las comparaciones múltiples se controlaron mediante el procedimiento de Holm. Para evaluar la dependencia interna de los proyectos, se ejecutaron pruebas de permutación por repositorio.

Figura 1. *Flujo de selección, emparejamiento y conformación de muestras.*

Fuente: Elaboración propia.

Asimismo, se ajustaron dos modelos logísticos condicionales con estratos definidos por cada par: uno para la recepción de revisión humana y otro para la fusión de las contribuciones. En ambos modelos se incluyeron como covariables el origen de la contribución, los logaritmos del número de líneas modificadas, archivos y

commits, la densidad de hallazgos identificados mediante Ruff, el cambio en el índice de mantenibilidad y el cambio en la complejidad media. Las covariables continuas fueron estandarizadas. Las razones de momios se interpretaron como asociaciones de carácter observacional y no como efectos causales. Se adoptó un nivel de significación estadística de 0,05.

Resultados y Discusión

La muestra principal reunió 400 pull requests de 47 repositorios. Devin aportó 96 pares, OpenAI Codex 78 y GitHub Copilot 26. Predominaron las funcionalidades, con 110 pares, y las correcciones, con 78. La mediana de separación temporal dentro del par fue 40,24 días y la razón mediana entre tamaños humano/agente fue 1,00. Como se resume en la Tabla 1, las diferencias estandarizadas de las principales variables de emparejamiento permanecieron en magnitudes pequeñas, lo que respalda la comparabilidad inicial entre las contribuciones generadas por agentes y sus controles humanos. El flujo de selección muestra que la reducción más importante ocurrió al exigir código reconstruible y un control del mismo repositorio. Esta decisión sacrificó cobertura para evitar una comparación superficial entr proyectos con prácticas incompatibles. La muestra no representa todo GitHub, pero sí ofrece un contraste interno más defendible que una agregación masiva sin emparejamiento.

Tabla 1. *Composición y equilibrio de la muestra principal.*

Indicador	Resultado
Pares principales	200 (400 pull requests)
Repositorios	47
Agentes	Devin 96; Codex 78; Copilot 26
Separación temporal mediana	40,24 días
Razón mediana de tamaño humano/agente	1,00
Diferencia estandarizada: líneas	0,028
Diferencia estandarizada: archivos	-0,073
Diferencia estandarizada: commits	-0,133

Fuente: Elaboración propia.

No identificamos una desventaja técnica sistemática para las contribuciones de agentes. La densidad mediana de nuevos hallazgos Ruff fue cero en ambos grupos y la diferencia media fue -0,041 hallazgos por 100 líneas. Cuarenta y cinco contribuciones de agentes, 22,5%, introdujeron al menos un hallazgo, frente a 48 humanas, 24,0%; la diferencia de -1,5 puntos porcentuales no fue significativa.El cambio mediano del índice de mantenibilidad fue -0,470 en agentes y -0,438 en humanas. La complejidad ciclomática media aumentó 0,011 en ambos grupos y el cambio neto de bloques con complejidad alta permaneció en cero. Ninguna comparación resistió el ajuste de Holm ni las permutaciones por repositorio. Las sensibilidades reprodujeron el mismo patrón. En conjunto, los resultados de la Tabla 2 muestran diferencias reducidas e inestables entre grupos, sin evidencia estadística que permita atribuir a las contribuciones de agentes un deterioro técnico sistemático. Este resultado matiza las conclusiones derivadas de estudios de fragmentos o de conjuntos no emparejados. Pearce et al. (2022) y Perry et al. (2023) demostraron que un asistente puede proponer soluciones inseguras en tareas controladas, pero nuestro diseño observa cambios que atravesaron un flujo real de repositorio. Semgrep produjo una sola alerta nueva en la muestra principal. El patrón correspondió al uso de MD5 para construir una clave interna de caché a partir de

una versión y un texto. El código no utilizó el hash para autenticación, contraseñas, firmas, cifrado ni verificación de integridad; por ello, clasificamos la alerta como no aplicable a seguridad. Después de la validación, ambos grupos quedaron con cero alertas nuevas válidas.

Tabla 2. *Comparación emparejada de indicadores técnicos.*

Indicador	Agentes	Humanas	p
Hallazgos Ruff por 100 líneas, mediana	0,000	0,000	0,632
Con al menos un hallazgo nuevo	22,5%	24,0%	0,798
Cambio de mantenibilidad, mediana	-0,470	-0,438	0,523
Cambio de complejidad media, mediana	0,011	0,011	0,418
Cambio neto de bloques CC ≥ 11	0,000	0,000	0,598
Alertas de seguridad validadas	0	0	No aplicable

Fuente: Elaboración propia.

La ausencia de alertas validadas no prueba ausencia de vulnerabilidades. El análisis estático tiene cobertura limitada, depende de reglas y puede omitir fallos lógicos o contextuales. El resultado más prudente es que no observamos una diferencia de seguridad detectable mediante el conjunto de reglas aplicado. Este alcance evita transformar un resultado nulo en una afirmación de equivalencia. La dimensión de revisión mostró el contraste más marcado. Solo 73 contribuciones de agentes, 36,5%, recibieron revisión humana estricta, frente a 135 humanas, 67,5%.

La diferencia emparejada fue -31,0 puntos porcentuales y permaneció significativa en la permutación por repositorio. Al incorporar los 81 comentarios generales validados, las proporciones aumentaron a 41,0% y 71,0%, pero la brecha casi no cambió. La Figura 2 muestra que las diferencias más amplias entre los grupos se concentraron en la recepción de revisión humana y en la fusión de las contribuciones, no en los indicadores estáticos del código. Los recuentos globales también fueron menores en agentes: mediana de cero eventos frente a uno, cero revisores frente a uno y menos mensajes. Sin embargo, esos valores reflejan sobre todo que muchas contribuciones de agentes nunca entraron en una revisión humana observable. En los 58 pares donde ambos miembros sí fueron revisados, no encontramos diferencias en eventos, mensajes, revisores, solicitudes de cambios, commits posteriores, rondas ni tiempos. Por tanto, los datos contradicen la hipótesis de que una contribución de agente demanda necesariamente más trabajo cuando un revisor decide examinarla. Como se detalla en la Tabla 3, los recuentos globales fueron menores para los agentes, pero las diferencias desaparecieron en gran medida cuando el análisis se restringió a los pares en los que ambas contribuciones recibieron revisión humana.

La distinción entre recepción e intensidad es analíticamente decisiva. Duma et al. (2026) ya mostraron que la actividad registrada en contribuciones de agentes puede estar dominada por automatizaciones. Nuestro filtrado confirma ese riesgo: 132 de 248 comentarios generales candidatos no representaban juicio humano. Contar indiscriminadamente esos mensajes habría inflado la supervisión y habría ocultado la brecha real. Se fusionaron 104 contribuciones de agentes, 52,0%, y 167 humanas, 83,5%. La diferencia emparejada fue -31,5 puntos porcentuales. En el modelo condicional, el origen agente se asoció con menores

probabilidades de revisión humana, OR=0,140, IC 95% 0,064-0,306, y de fusión, OR=0,147, IC 95% 0,069-0,314. La Tabla 4 presenta los coeficientes completos de ambos modelos y permite observar que la asociación con el origen de la contribución persistió después de controlar el tamaño del cambio, los commits y los indicadores técnicos.

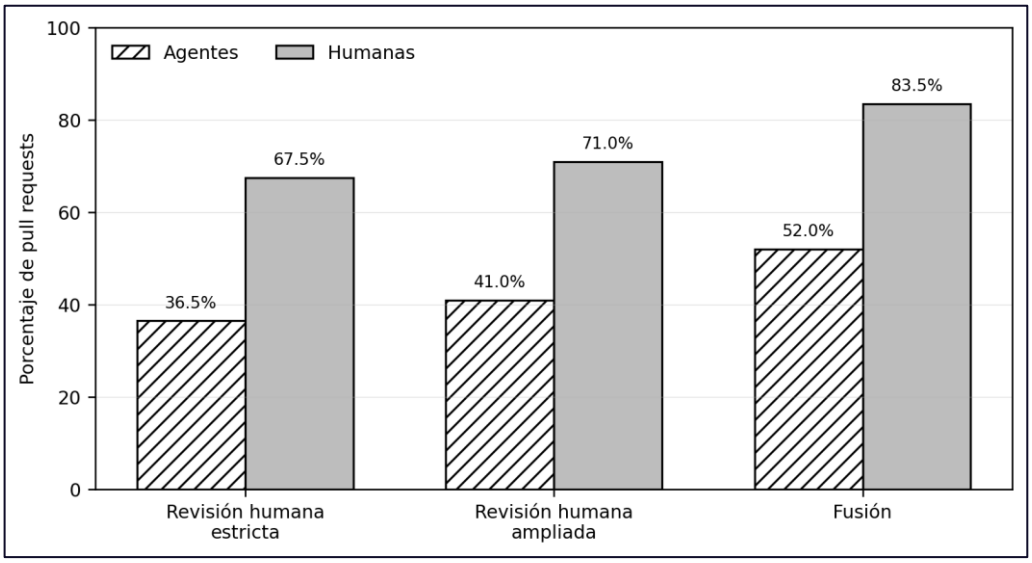

Figura 2. *Recepción de revisión humana y fusión en la muestra principal.*

Fuente: Elaboración propia a partir de eventos de GitHub validados.

Tabla 3. *Revisión e integración de las contribuciones*

Resultado	Agentes	Humanas	Diferencia	p
Revisión humana estricta	36,5%	67,5%	-31,0 pp	<0,001
Revisión humana ampliada	41,0%	71,0%	-30,0 pp	<0,001
Pull request fusionada	52,0%	83,5%	-31,5 pp	<0,001
Eventos humanos, mediana	0	1	-1,385 media	<0,001
Revisores únicos, mediana	0	1	-0,385 media	<0,001
Eventos automáticos, mediana	2	0	+1,315 media	0,082*

Fuente: Elaboración propia.

Las métricas técnicas no explicaron la recepción de revisión: los intervalos de los hallazgos Ruff, mantenibilidad y complejidad incluyeron el valor nulo. En el modelo de fusión, el número estandarizado de archivos mostró una asociación positiva, OR=3,498, pero no la interpretamos como una ventaja causal. Dentro de pares y tareas parecidas, esa señal puede capturar estructura residual del trabajo o prácticas de proyectos concretos. El resultado central permanece: la diferencia de integración persiste sin un deterioro estático detectable. Nachuma y Zibran (2026) sostienen que la integración de agentes depende de señales de colaboración y convergencia durante la revisión. Nuestros resultados son compatibles con esa lectura, pero agregan una advertencia: antes de evaluar la calidad de la interacción, muchas contribuciones de agentes ni siquiera reciben revisión humana. La gobernanza del repositorio, la forma de presentación, la confianza en la cuenta y la selección temprana

de tareas pueden pesar tanto como el código. La Figura 3 representa las razones de momios ajustadas del origen agente y evidencia que sus intervalos de confianza permanecieron por debajo de uno tanto para la recepción de revisión humana como para la fusión.

Tabla 4. *Modelos logísticos condicionales ajustados por par.*

Modelo	Predictor	OR	IC 95%	p
Recepción de revisión	Contribución de agente	0,140	0,064-0,306	<0,001
Recepción de revisión	Hallazgos Ruff	0,534	0,276-1,036	0,063
Recepción de revisión	Cambio de mantenibilidad	0,947	0,539-1,661	0,848
Recepción de revisión	Cambio de complejidad	0,853	0,550-1,322	0,476
Fusión	Contribución de agente	0,147	0,069-0,314	<0,001
Fusión	Archivos modificados	3,498	1,506-8,120	0,004
Fusión	Hallazgos Ruff	0,792	0,534-1,174	0,245

Fuente: Elaboración propia.

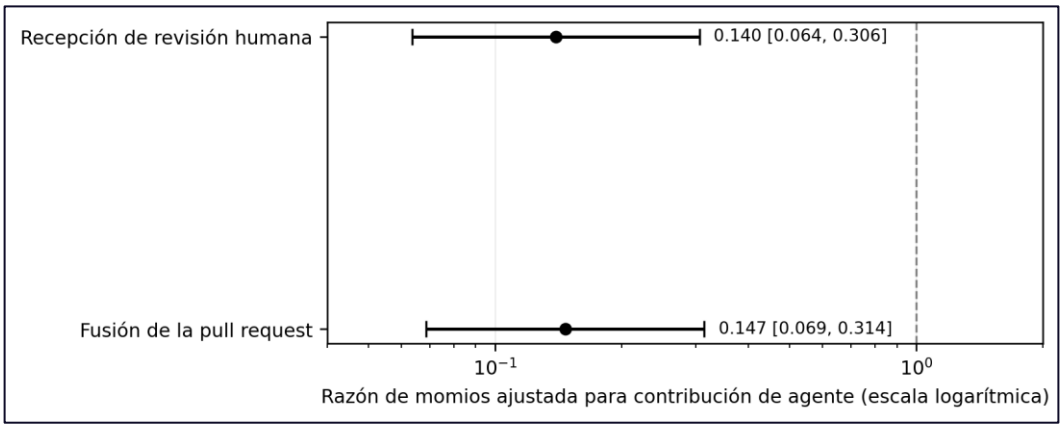

Figura 3. *Asociación ajustada del origen agente con revisión humana y fusión.*

Fuente: Elaboración propia

El patrón agrupado no fue uniforme. Devin recibió revisión en 29,2% de sus contribuciones frente a 76,0% de sus controles; Codex, 25,6% frente a 50,0%. GitHub Copilot presentó 96,2% frente a 88,5%, sin diferencia concluyente. Las tasas de fusión fueron 31,3% para Devin, 64,1% para Codex y 92,3% para Copilot; sus controles alcanzaron 82,3%, 82,1% y 92,3%, respectivamente. La submuestra de Copilot contiene solo 26 pares y opera en flujos distintos, por lo que no permite afirmar superioridad. Sí demuestra que "contribución de agente" no constituye una categoría homogénea. Cada agente se inserta mediante cuentas, permisos, tareas y convenciones diferentes. Una política uniforme puede confundir problemas del flujo de trabajo con limitaciones del modelo. La interpretación también enfrenta amenazas de validez. El grupo humano puede incluir asistencia privada no declarada; la autoría pública no captura toda edición híbrida; GitHub no registra revisiones externas; el análisis se restringe a Python y proyectos populares; y el emparejamiento solo controla variables observables. Además, la ausencia de comentarios puede representar rechazo temprano, aceptación directa o evaluación invisible. Por eso, describimos asociaciones y patrones, no causalidad ni esfuerzo laboral total.

Conclusiones

Del análisis emparejado y de las comprobaciones de sensibilidad se concluyó

que las contribuciones generadas por agentes no mostraron un incremento detectable de la deuda técnica, la complejidad del código ni las alertas potenciales de seguridad en comparación con contribuciones humanas de características similares. Estos resultados indican que, dentro de las condiciones evaluadas, el origen de la contribución no estuvo asociado con un deterioro significativo de las métricas estáticas de calidad analizadas. La principal diferencia se identificó en el proceso de supervisión e integración. Las contribuciones de agentes ingresaron con menor frecuencia en procesos de revisión humana observable y presentaron una menor proporción de fusión. Este hallazgo sugiere que las diferencias entre contribuciones humanas y generadas por agentes se manifestaron principalmente en las prácticas de revisión y aceptación, más que en las características técnicas medidas mediante herramientas de análisis estático. Cuando ambos miembros de un par recibieron revisión humana, la intensidad del proceso y los tiempos empleados fueron semejantes. Por consiguiente, los resultados no respaldan la afirmación de que el código generado por agentes requiera un mayor esfuerzo humano una vez iniciada su evaluación. La diferencia observada se relacionó más con la probabilidad de recibir revisión que con la carga de trabajo necesaria durante el proceso revisor.

Asimismo, las métricas estáticas consideradas no permitieron explicar completamente la brecha encontrada en la revisión y la fusión de las contribuciones. En consecuencia, los proyectos de software deberían complementar los controles automáticos con reglas explícitas de trazabilidad, revisión humana y presentación de evidencia de pruebas. Estas medidas deben adaptarse tanto al tipo de agente utilizado como a las características del flujo de integración de cada repositorio. La heterogeneidad observada entre Devin, Codex y Copilot impide establecer un único perfil de riesgo aplicable a todos los agentes de programación. Las investigaciones futuras deberán analizar por separado cada agente, los distintos tipos de tareas y las prácticas específicas de los repositorios. También será necesario incorporar pruebas dinámicas y seguimiento posterior a la fusión para evaluar con mayor amplitud la calidad, seguridad y mantenibilidad de las contribuciones generadas mediante inteligencia artificial.

Referencias Bibliográficas

Barke, S., James, M., & Polikarpova, N. (2023). Grounded Copilot: How programmers interact with code-generating models. *Proceedings of the ACM on Programming Languages, 7*(OOPSLA1), Article 78, 85–111. https://doi.org/10.1145/3586030

Duma, K., Wróblewski, P., Bobińska, J., Winiarska, J., & Przymus, P. (2026). These aren't the reviews you're looking for: How humans review AI-generated pull requests. *arXiv*. https://doi.org/10.48550/arXiv.2605.02273

Gousios, G., Pinzger, M., & Deursen, A. (2014). An exploratory study of the pull-based software development model. *Proceedings of the 36th International Conference on Software Engineering*, 345–355. https://doi.org/10.1145/2568225.2568260

Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., German, D., & Damian, D. (2016). An in-depth study of the promises and perils of mining GitHub. *Empirical Software Engineering, 21*, 2035–2071. https://doi.org/10.1007/s10664-015-9393-5

Li, H., Zhang, H., & Hassan, A. (2026). AIDev: Studying AI coding agents on GitHub. *Proceedings of the 23rd International Conference on Mining Software Repositories*. https://doi.org/10.1145/3793302.3797249

MacLeod, L., Greiler, M., Storey, M., Bird, C., & Czerwonka, J. (2018). Code reviewing in the trenches: Challenges and best practices.

IEEE Software, 35(4), 34–42. https://doi.org/10.1109/MS.2017.265100500

McIntosh, S., Kamei, Y., Adams, B., & Hassan, A. (2016). An empirical study of the impact of modern code review practices on software quality. *Empirical Software Engineering, 21*, 2146–2189. https://doi.org/10.1007/s10664-015-9381-9

Nachuma, C., & Zibran, M. (2026). When AI teammates meet code review: Collaboration signals shaping the integration of agent-authored pull requests. *arXiv*. https://doi.org/10.48550/arXiv.2602.19441

Ogenrwot, D., & Businge, J. (2026). How AI coding agents modify code: A large-scale study of GitHub pull requests. *arXiv*. https://doi.org/10.48550/arXiv.2601.17581

Pansuriya, K., Ghorbani, E., Singh, D., & AlOmar, E. (2026). Predicting acceptance and review effort in human and agent pull requests. *arXiv*. https://doi.org/10.48550/arXiv.2607.12057

Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2022). Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. *2022 IEEE Symposium on Security and Privacy*, 754–768. https://doi.org/10.1109/SP46214.2022.9833571

Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do users write more insecure code with AI assistants? *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2785–2799. https://doi.org/10.1145/3576915.3623157

Rigby, P., & Bird, C. (2013). Convergent contemporary software peer review practices. *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, 202–212.https://doi.org/10.1145/2491411.2491444

Sadowski, C., Söderberg, E., Church, L., Sipko, M., & Bacchelli, A. (2018). Modern code review: A case study at Google. *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, 181–190. https://doi.org/10.1145/3183519.3183525

Sawada, S., Shirai, T., Kashiwa, Y., Yamaguchi, K., Iwata, H., & Iida, H. (2026). To what extent does agent-generated code require maintenance? An empirical study. *arXiv*. https://doi.org/10.48550/arXiv.2605.06464

Vaithilingam, P., Zhang, T., & Glassman, E. (2022). Expectation versus experience: Evaluating the usability of code generation tools powered by large language models. *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, Article 332. https://doi.org/10.1145/3491101.3519665

Watanabe, K., Tsuchida, R., Monno, T., Huang, B., Yamasaki, K., Fan, Y., Shimari, K., & Matsumoto, K. (2026). How AI coding agents communicate: A study of pull request description characteristics and human review responses. *arXiv*. https://doi.org/10.48550/arXiv.2602.17084

Watanabe, M., Li, H., Kashiwa, Y., Reid, B., Iida, H., & Hassan, A. (2025). On the use of agentic coding: An empirical study of pull requests on GitHub. *arXiv*. https://doi.org/10.48550/arXiv.2509.14745

Xu, G., Subramanian, A., & Karthik, N. (2026). AI agent pull requests on GitHub: Frequency, structure, and merge conflict rates. *arXiv*. https://doi.org/10.48550/arXiv.2607.04697

Declaraciones éticas y editoriales del artículo
Contribución de los autores (Taxonomía CRediT) Diego Marcelo Reina Haro: conceptualización, formulación de los objetivos de investigación, diseño metodológico, análisis formal, interpretación de los resultados, redacción del borrador original, revisión crítica del contenido científico y supervisión general del estudio. Andrea Fernanda Criollo Cantos: curación y organización de los datos, desarrollo y ejecución de los procedimientos computacionales, aplicación de las herramientas de análisis estático, validación de los resultados y elaboración de tablas y figuras. Carlos Miguel Campoverde Santos: investigación, verificación de los criterios de selección y emparejamiento, validación de la clasificación de las alertas de seguridad y de los eventos de revisión, interpretación crítica de los resultados y revisión y edición del manuscrito. Estalin Fabián Mejía Hidalgo: provisión de recursos, supervisión metodológica, revisión de la reproducibilidad del estudio, revisión crítica del manuscrito y aprobación de la versión final.
Declaración de conflicto de intereses Los autores declaran que no existe conflicto de intereses en relación con la investigación presentada, la autoría del manuscrito ni la publicación del presente artículo.
Declaración de financiamiento La presente investigación no recibió financiamiento específico de agencias públicas, comerciales o de organizaciones sin fines de lucro. En caso de existir financiamiento institucional o externo, este deberá ser declarado explícitamente por los autores en esta sección.
Declaración del editor El editor responsable certifica que el proceso editorial del presente artículo se desarrolló conforme a los principios de integridad científica, transparencia y buenas prácticas editoriales. El manuscrito fue sometido a un proceso de evaluación mediante revisión por pares doble ciego, garantizando la confidencialidad de la identidad de los autores y revisores durante todo el proceso de dictamen académico. Asimismo, el editor declara que el artículo cumple con los criterios científicos, metodológicos y éticos establecidos por la revista.
Declaración de los revisores Los revisores externos que participaron en la evaluación del presente manuscrito declaran haber realizado el proceso de revisión de manera objetiva, independiente y confidencial. Asimismo, manifiestan que no mantienen conflictos de interés con los autores ni con la investigación evaluada, y que sus observaciones y recomendaciones se fundamentan exclusivamente en criterios científicos, metodológicos y académicos.
Declaración ética de la investigación Los autores declaran que la investigación se desarrolló respetando los principios éticos de la investigación científica, garantizando la confidencialidad de los datos y el respeto a los participantes del estudio. En los casos en que la investigación involucre seres humanos, los procedimientos deben ajustarse a los principios éticos establecidos en la Declaración de Helsinki y a las normativas institucionales correspondientes.
Declaración sobre el uso de inteligencia artificial Los autores declaran que el uso de herramientas de inteligencia artificial, en caso de haberse utilizado durante el proceso de investigación o redacción del manuscrito, se realizó únicamente como apoyo técnico para mejorar la claridad del lenguaje o el análisis de información, manteniendo siempre la responsabilidad intelectual sobre el contenido del artículo. Las herramientas de inteligencia artificial no fueron utilizadas como autoras del manuscrito ni sustituyen la responsabilidad académica de los investigadores.
Disponibilidad de datos Los datos que respaldan los resultados de esta investigación estarán disponibles previa solicitud razonable al autor de correspondencia, respetando las normas éticas y de confidencialidad establecidas por la investigación.