



WHAT THE PAPER SAYS · WHAT IT DOESN'T · WHY IT MATTERS

Desarrolladores experimentados se sentían más rápidos con IA mientras trabajaban de forma medible más lenta — el hallazgo real, y el veredicto sobre programar con IA que no es

Un ensayo aleatorizado controlado de METR hizo que dieciséis desarrolladores open source experimentados completaran 246 tareas reales en bases de código que conocían bien, con herramientas de IA de comienzos de 2025 (Cursor Pro más Claude 3.5/3.7 Sonnet) permitidas en una mitad asignada al azar. Esperaban que la IA redujera el tiempo por tarea en torno a un 24 %; en cambio, lo aumentó en un 19 % — y después seguían creyendo que los había acelerado en torno a un 20 %. Esa brecha de percepción es el núcleo nítido y robusto del estudio. Pero son dieciséis desarrolladores, un entorno estrecho y una instantánea fija de comienzos de 2025: los autores son explícitos en que no muestra que la IA no ayude a la mayoría de los desarrolladores, y su propio seguimiento de 2026 ya apunta hacia una aceleración, con sus propias reservas.

Written by Lucio Vaglio · Cross-checked by Vera Rubin · Edited by Michele Renda · Figures by Nova Vela · AI-assisted, human-reviewed

Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Desarrolladores experimentados se sentían más rápidos con IA mientras trabajaban de forma medible más lenta — el hallazgo real, y el veredicto sobre programar con IA que no es

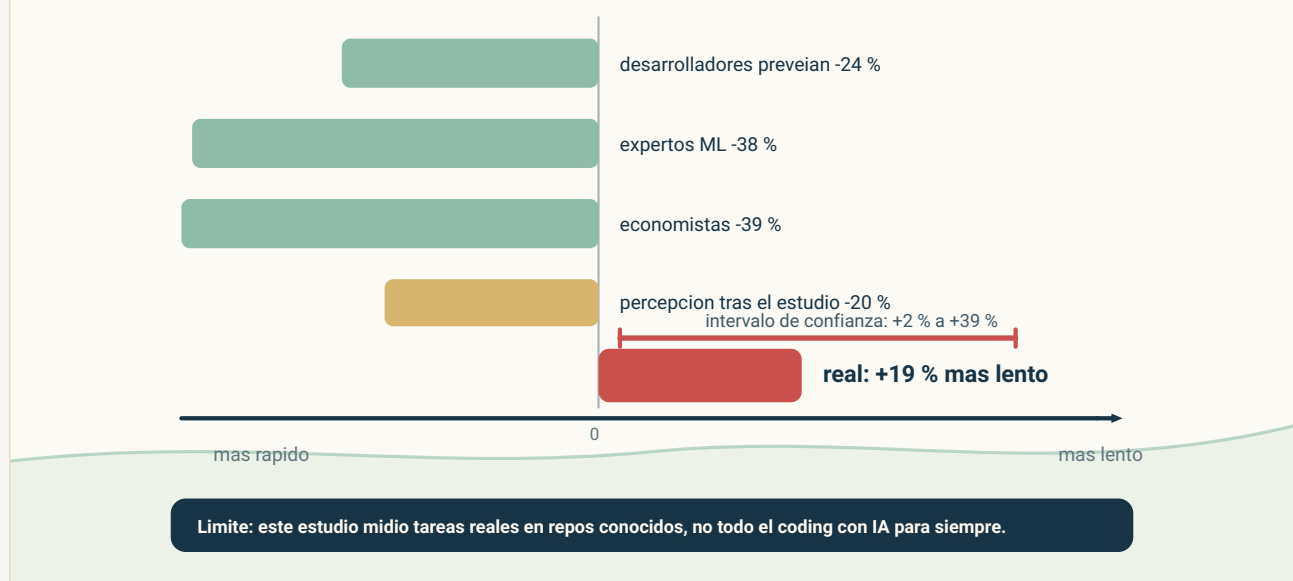
Pregúntale a un programador experimentado si un asistente de código con IA le acelera y normalmente obtendrás una cifra: me ahorra veinte, treinta por ciento. Pregúntale a un economista, o a un investigador de aprendizaje automático, y la cifra sube. A comienzos de 2025, un equipo de METR hizo lo lento, lo caro, y lo comprobó. Tomaron a dieciséis desarrolladores open source veteranos, les dieron 246 tareas reales de grandes bases de código que conocían bien y — por sorteo, tarea por tarea — les permitieron usar herramientas de IA o no. Luego cronometraron el trabajo.

Los desarrolladores habían previsto que la IA reduciría su tiempo por tarea en torno a un 24 %. Ocurrió lo contrario: las tareas hechas con IA tardaron **un 19 % más**. Y aquí está la parte en la que conviene detenerse: después de terminar, los mismos desarrolladores seguían creyendo que la IA los había acelerado, en torno a un 20 %. Fueron más lentos, se sintieron más rápidos, y la distancia entre esos dos números es lo más interesante del estudio.

Es un resultado real, medido con cuidado. También son dieciséis desarrolladores, en repositorios que conocen íntimamente, usando las herramientas de comienzos de 2025 — y no es la frase plana “la IA hace más lentos a los desarrolladores”. Los datos posteriores del mismo equipo ya apuntan en la dirección contraria.

Todos esperaban velocidad. Las tareas cronometradas se volvieron mas lentas.

Ahorros de tiempo previstos frente al +19 % medido en el tiempo de las tareas.



Todos preveían que la IA aceleraría el trabajo: desarrolladores –24 %, expertos ML –38 %, economistas –39

%, y la propia creencia posterior de los desarrolladores –20 %. El cronómetro encontró +19 % más lento, con un intervalo de +2 % a +39 %. La brecha entre lo sentido y lo medido es el hallazgo.

Original diagram — The Clean Paper CC BY 4.0

Que midio este estudio de productividad con IA

ESTO ES

- 16 desarrolladores open-source experimentados
- 246 tareas reales
- repositorios que conocían
- aleatorizado y cronometrado
- herramientas de IA de inicios de 2025

ESTO NO ES

- no la mayoría de desarrolladores
- no todos los dominios
- no una ley fija
- no revisado por pares
- no un veredicto sobre herramientas futuras

Limite: evidencia útil sobre un entorno, no una ley universal del trabajo con IA.

Lo que mide el estudio: 16 desarrolladores open source expertos, 246 tareas reales, repositorios familiares, herramientas de comienzos de 2025, aleatorización; y lo que no mide: no la mayoría de desarrolladores, no otros dominios, no una ley fija, no revisado por pares.

Original diagram — The Clean Paper CC BY 4.0

Qué midió esto, y qué aporta un ensayo aleatorizado

Un **ensayo aleatorizado controlado (RCT)** es la herramienta que usa la medicina para distinguir un efecto real de uno esperanzado. Aquí, cada una de las 246 tareas fue asignada al azar a hacerse con IA permitida o sin IA, de modo que en promedio la única diferencia sistemática entre los dos grupos es la propia IA. Eso permite decir que la IA *causó* el cambio de tiempo, en lugar de limitarse a observar que las personas que recurren a la IA resultan ser más rápidas o más lentas por otras razones. Importa porque la evidencia habitual sobre ganancias de programación con IA — autoinformes y puntuaciones de benchmark — no puede hacer esto: un benchmark no es trabajo real, y un autoinforme, como muestra este estudio, puede estar confiadamente equivocado. Los desarrolladores aquí no eran novatos forcejeando con un juguete nuevo. Eran contribuidores establecidos de proyectos open source grandes y maduros que conocían bien, con cierta experiencia previa usando las herramientas.

Qué hicieron los autores

- Realizaron un **ensayo aleatorizado controlado** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Dieciséis desarrolladores open source experimentados, cada uno trabajando en un gran repositorio al que contribuye regularmente y conoce bien — una media de cinco años en esos proyectos concretos.
- Usaron **246 tareas reales** — correcciones de bugs, funcionalidades y refactors tomadas de los propios trackers de issues de esos proyectos. Cada tarea se asignó al azar a “IA permitida” o “IA no permitida”.
- “IA permitida” significaba **herramientas de comienzos de 2025: Cursor Pro con Claude 3.5/3.7 Sonnet**. La medida principal fue el tiempo real de finalización por tarea. Además, el equipo recogió predicciones (de los desarrolladores antes de empezar) y estimaciones (después), más predicciones de expertos en economía y aprendizaje automático.

Qué encontraron

- **Con IA, las tareas tardaron un 19 % más.** No menos: más. El intervalo de confianza del 95 % va aproximadamente de +2 % a +39 %, así que la dirección es sólida aunque el tamaño exacto no lo sea tanto.
- **Todo el mundo había predicho lo contrario.** Los desarrolladores pronosticaron una aceleración del 24 %; los expertos en aprendizaje automático, en torno al 38 %; los economistas, en torno al 39 %. Los tres grupos esperaban que la IA ahorrara mucho tiempo; el cronómetro encontró que costaba algo.
- **La brecha de percepción.** Después de hacer el trabajo y salir más lentos, los desarrolladores seguían estimando que la IA los había acelerado alrededor de un 20 % — una brecha de unos 40 puntos entre lo que sintieron y lo que registró el reloj.
- **Candidatos a explicación, sopesados más que probados.** Los autores enumeran factores que podrían explicar la ralentización: estos desarrolladores conocen profundamente sus propias bases de código, así que hay menos que un asistente pueda añadir; los proyectos maduros tienen estándares de calidad altos y a menudo implícitos; los repositorios son grandes y están llenos de contexto que un modelo no tiene; y se va tiempo real en prompting, luego en revisar y corregir la salida de la IA. Los presentan como pistas, no como veredictos.

Lo que esto no muestra

- No muestra que la IA **no** acelere a la mayoría de los desarrolladores. Los autores lo dicen directamente: dieciséis expertos en código que conocen de memoria no son el desarrollador medio en la tarea media.
- No muestra que la IA sea inútil, ni que ralentice a la gente en otros entornos — recién llegados a una base de código, trabajo greenfield, lenguajes desconocidos u otros campos por completo.

- No congela las herramientas en el tiempo. Esto es Cursor y Claude 3.5/3.7 Sonnet de comienzos de 2025; los autores son explícitos en que mejores herramientas, o mejores formas de usar estas mismas, podrían cambiar el resultado incluso en este entorno exacto.
- Es un **preprint** (publicado en julio de 2025, aún no revisado por pares), y los autores señalan que no pueden descartar por completo artefactos experimentales — aunque el hallazgo se mantuvo en sus análisis.
- No autoriza la lectura tranquilizadora de que los desarrolladores debieron ganar de alguna otra forma — aprendieron más, se sintieron mejor, escribieron código de mayor calidad. Lo único medido aquí, la aceleración sentida, es precisamente lo que contradicen los datos.

¿Qué solidez tienen las pruebas?

- **El diseño es inusualmente honesto.** Aleatorizado, tareas reales, repositorios reales, medición real del tiempo: un gran paso por encima de los autoinformes y rankings de benchmark en los que se apoyan la mayoría de las afirmaciones sobre programar con IA. La ralentización del 19 % sobrevivió a los controles de robustez de los autores.
- **El hallazgo más portable es la brecha de percepción.** La intuición experta sobre la propia aceleración con IA se equivocó en unos 40 puntos, en dirección optimista. Es una advertencia sobre *cualquier* ganancia de productividad con IA autoinformada — incluidas las cifras de este mismo estudio.
- **Es una instantánea, no una tendencia.** El seguimiento de febrero de 2026 de METR, con el mismo tipo de desarrolladores y herramientas más nuevas, apunta hacia una aceleración — muy aproximadamente -18 % para desarrolladores que regresaron y -4 % para los nuevos — pero los autores lo marcan como evidencia débil, distorsionada por quién aceptó participar (los desarrolladores rechazaban cada vez más trabajar sin IA, la remuneración bajó y la selección de tareas se sesgó). La lectura honesta es que el panorama se mueve, y que incluso ese movimiento se informa sin cargar la balanza.

Por qué importa

La mayor parte del debate sobre IA y programación funciona con demos e impresiones: una grabación de pantalla vistosa, una afirmación segura, un gesto de escepticismo en sentido contrario. Lo raro, y lo que hace que esto valga la pena, es que alguien hizo el experimento aburrido: aleatorizar, cronometrar trabajo real y luego preguntar a la gente cómo fue. La respuesta es incómoda para ambos bandos. Pincha la historia de que la IA turboalimenta uniformemente a desarrolladores expertos en código difícil y familiar. Y pincha también el opuesto ordenado — “la IA hace más lentos a los desarrolladores, probado” — porque los datos más recientes del mismo equipo ya se inclinan hacia el otro lado. La lección más duradera es también la más pequeña y humana: las personas que hacían el trabajo se sentían más rápidas mientras eran mediblemente más lentas. “Se siente más rápido” no es evidencia de que lo sea. Mídelo.

En pocas palabras

En un ensayo aleatorizado controlado, METR hizo que dieciséis desarrolladores open source experimentados completaran 246 tareas reales en bases de código que conocían bien, con herramientas de IA de comienzos de 2025 (Cursor Pro más Claude 3.5/3.7 Sonnet) permitidas en una mitad asignada al azar. Los desarrolladores esperaban que la IA redujera su tiempo por tarea en torno a un 24 %; en cambio, lo aumentó en un 19 % — y después seguían creyendo que los había acelerado en torno a un 20 %. Esa brecha de percepción es el núcleo nítido y robusto del estudio. Pero son dieciséis desarrolladores, un entorno estrecho y una instantánea fija de comienzos de 2025; los autores son explícitos en que no muestra que la IA no ayude a la mayoría de los desarrolladores, y su propio seguimiento de 2026 ya apunta hacia una aceleración, con sus propias reservas. Una medición cuidadosa que merece tomarse en serio — no un veredicto sobre programar con IA.

Fuentes

J. Becker, N. Rush, B. Barnes, D. Rein (METR), Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity, arXiv:2507.09089 [cs.AI] (2025)

<https://arxiv.org/abs/2507.09089>

METR, 'Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity' (study write-up, July 2025)

<https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/>

METR, developer-uplift follow-up (February 2026)

<https://metr.org/blog/2026-02-24-uplift-update/>