

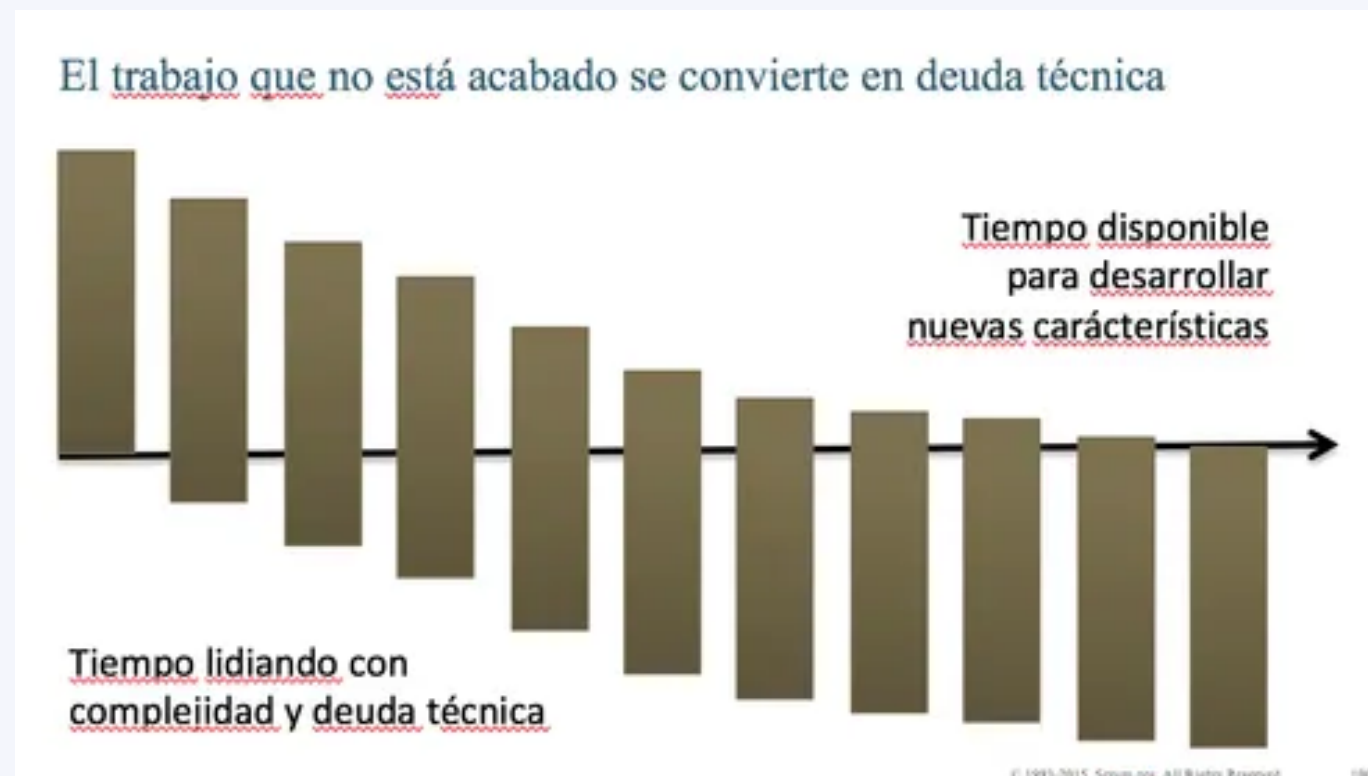
Deuda Técnica en la Era de la IA: El Costo Oculto de la Automatización

Garcia Obeso Carlos Mario

Play

Introduccion

La mayoría conocemos el concepto de Deuda Técnica acuñado por Ward Cunningham en 1992, El cual significa es el costo implícito de elegir una solución fácil y rápida ahora, en lugar de un enfoque mejor que tomaría más tiempo. Es como usar una tarjeta de crédito; si no pagas el saldo (refactorización), los intereses (la dificultad de mantener el código) te comen.



Antecedentes y hechos factuales

Un informe reciente de Ox Security analizó 300 proyectos open-source (50 parcialmente generados por IA) y halló que el código generado por IA suele carecer de “juicio arquitectónico” y presenta anti-patrones recurrentes: duplicación, evitación de refactorizaciones, sobre-especificación, y bugs repetidos

Según un estudio de GitClear analizando 211 millones de líneas de código en múltiples repositorios, tras la adopción generalizada de asistentes de IA:

- La duplicación de bloques de código aumentó hasta 8 veces.
- El “code churn” se duplicó entre 2021 y 2024.
- GitClear midió que los “bloques copiados/pegados” superaron por primera vez a las líneas “movidas” (refactorizadas), lo que indica un fuerte descenso en la reutilización de código y un aumento de código redundante.

“Security Weaknesses of Copilot-Generated Code in GitHub Projects” analizó fragmentos generados por IA (herramientas como GitHub Copilot y otras) y halló que un porcentaje significativo contenía debilidades de seguridad: por ejemplo ~29.5 % de los fragmentos en Python y ~24.2 % en JavaScript tenían vulnerabilidades catalogadas bajo estándares comunes de seguridad.

Un estudio reciente publicado en 2025 “Human-Written vs. AI-Generated Code: A Large-Scale Study of Defects, Vulnerabilities, and Complexity” comparando código escrito por humanos vs código generado por IA encontró que el código asistido por IA tendía a ser más repetitivo, contener más construcciones innecesarias o debug temporario, y mostrar un mayor riesgo de vulnerabilidades de seguridad.

Pan para hoy, Bugs para mañana: La Paradoja de la IA

2010 - 2015

La deuda era **lineal**. Si un desarrollador copiaba código de un foro, tenía que adaptarlo manualmente. Ese proceso de adaptación obligaba a entender (aunque fuera mínimamente) lo que se estaba pegando.

Tipo de Deuda: Código espagueti, falta de documentación, o malas decisiones de arquitectura tomadas por falta de experiencia.

2015 - 2030

La deuda ahora es **exponencial**. Un dev junior puede generar en 10 segundos un microservicio complejo que tardaría días en entender completamente.

Tipo de Deuda: El código parece correcto sintácticamente y pasa los tests básicos, pero contiene errores lógicos sutiles o alucinaciones de seguridad.

Problemas

Problema 1

Hemos eliminado el freno natural. Podemos generar deuda técnica a velocidad de máquina, pero nuestra capacidad humana para revisar el código sigue siendo la misma.

Problema 2

La IA reduce el tiempo de escritura, pero aumenta drásticamente el tiempo de lectura y depuración.



Cuando Copilot conduce ¿Quién paga la gasolina?

con tiempo de revisión humana.

Leer código es mucho más difícil que escribirlo. La IA escribe código con la lógica de una máquina, no con la narrativa de un humano. El desarrollador Senior o el líder técnico debe gastar horas descifrando por qué la IA eligió esa solución específica

con recursos de computación

La IA tiende a sugerir soluciones genéricas ("fuerza bruta") en lugar de optimizadas. Copilot a menudo prioriza que el código funcione sobre que sea eficiente. En la nube (AWS/Azure), código ineficiente = facturas más altas a fin de mes.

con aprendizaje perdido.

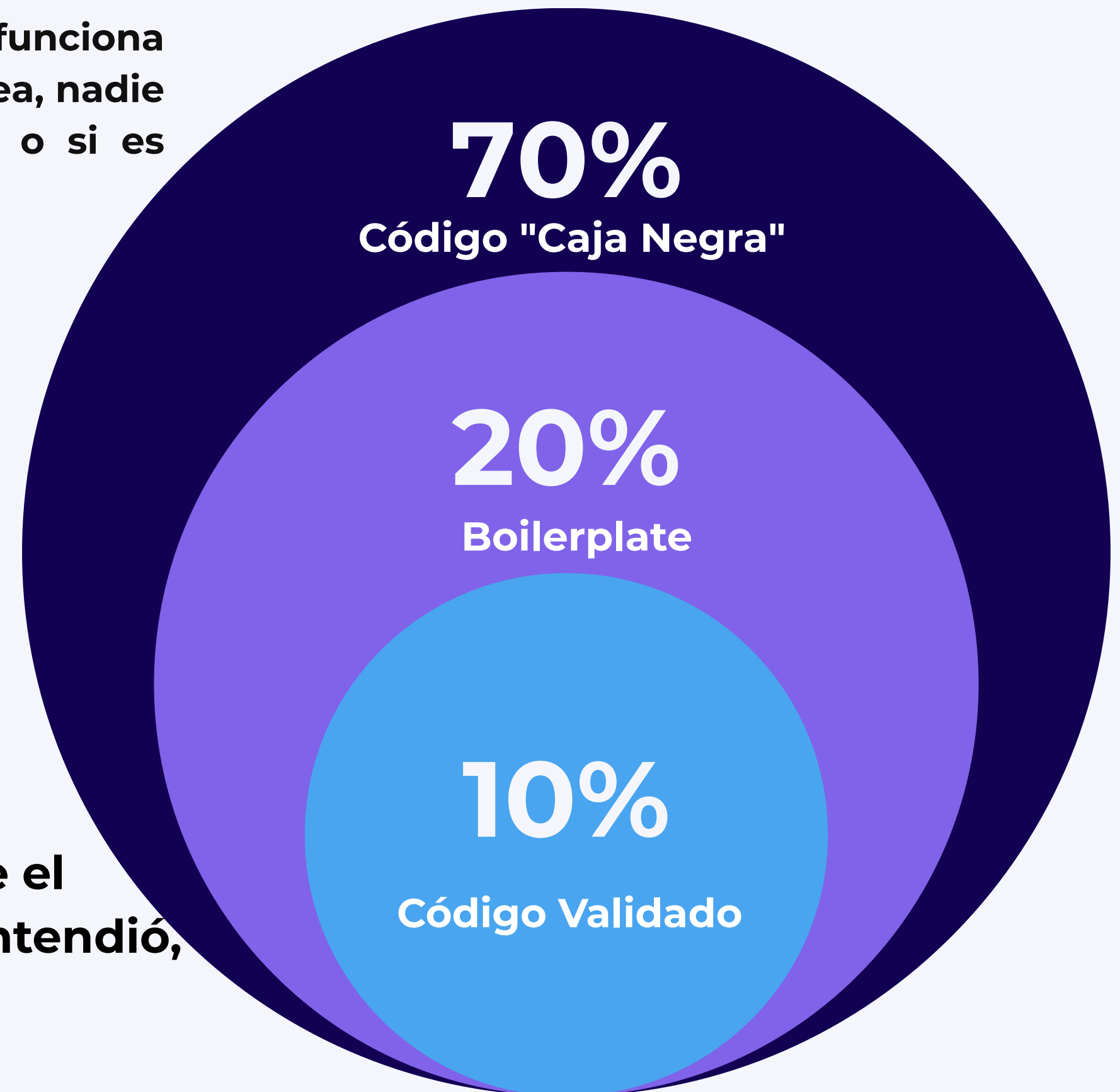
Si dejamos que Copilot conduzca siempre, los nuevos desarrolladores se convierten en "pasajeros" de su propio código. Atrofia de habilidades. Cuando el GPS falla (o la IA alucina), el conductor no sabe leer un mapa.

La Proporción del Código Zombi

Es el código generado que funciona "mágicamente". Nadie lo leyó línea por línea, nadie sabe exactamente qué librerías importó o si es seguro. Es deuda técnica.

Código repetitivo, verboso o innecesario que la IA generó para "rellenar" la solución. No rompe nada, pero ensucia la base de código y dificulta la lectura.

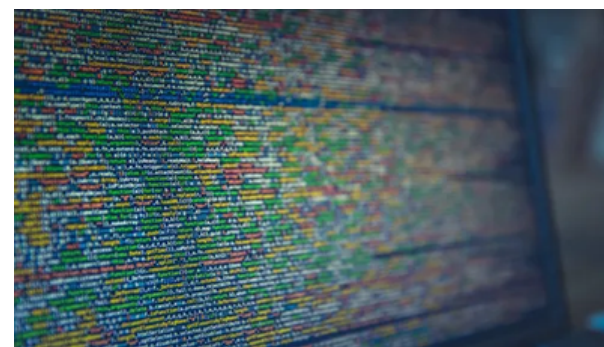
La única parte del sistema que el desarrollador humano realmente entendió, revisó y optimizó.



¿Cómo la IA Está Acelerando la Deuda Técnica?



La razón principal por la que la deuda aumentará es física. La IA puede generar 1,000 líneas de código en segundos (Costo casi cero). Un humano tarda horas en revisar, comprender y validar esas mismas líneas (Costo alto).



Estamos entrando en una era de 'Cajas Negras'. Antes, la deuda técnica era código feo que sabíamos cómo arreglar pero no teníamos tiempo. Ahora, la deuda técnica es código complejo que la IA escribió



A medida que confiamos más en Copilot, estamos entrenando a una generación de desarrolladores que son expertos en prompts, pero novatos en arquitectura.



Estudios en GitHub Copilot muestran que entre 24 % y 30 % del código generado contiene vulnerabilidades de seguridad o errores potenciales

Conclusion

La deuda técnica generada por inteligencia Artificial seguirá creciendo solo si nosotros seguimos actuando como simples operadores que copian y pegan lo que dice la máquina.

Pero si cambiamos nuestra mentalidad, si dejamos de obsesionarnos con la velocidad y empezamos a valorar la comprensión, entonces la IA deja de ser como quien dice una tarjeta de crédito con intereses altos y se convierte en lo que realmente debe ser: una herramienta para potenciar nuestra creatividad, no para reemplazar nuestro juicio.

“el código más valioso no es el que se escribe más rápido. El código más valioso es el que, dentro de 5 años, otro ser humano podrá leer, entender y agradecer que nos tomamos el tiempo de revisar”

**GRACIAS POR SU
ATENCIÓN**

