

INFORME

El impuesto de los tokens: Por qué los agentes de IA autónomos están erosionando los márgenes que debían proteger

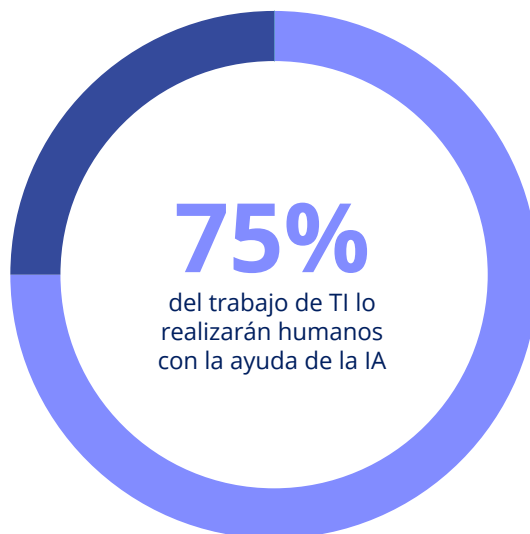
Una guía para directores
de operaciones (COO),
directores ejecutivos (CXO)
y vicepresidentes de
operaciones



La oportunidad es real

La IA ha transformado el desarrollo de software tal como lo conocemos. Es más rápido, más accesible y con más capacidades que nunca.

Los desarrolladores pueden crear y lanzar productos a velocidades que parecían imposibles incluso hace un año. El 84% de los desarrolladores utilizan actualmente la IA para acelerar su trabajo ([Stack Overflow Developer Survey, 2025](#)). Los prototipos de aplicaciones completas que antes tardaban semanas en desarrollarse, ahora se crean en cuestión de días. La democratización del desarrollo de software es una realidad y se está acelerando. Los CIO ya ven hacia dónde se dirige esto.



Para 2030, Gartner espera que el 75% del trabajo de TI lo realicen humanos aumentados con IA, y el 25% lo realice solo IA, sin que haya ninguna tarea que no se vea afectada por la IA ([Gartner, noviembre de 2025](#)). No se trata de una predicción a largo plazo. Es el horizonte de planificación con el que trabajan la mayoría de los departamentos de TI de las empresas.

La inversión está justificada. La dirección es la correcta. Pero más lanzamientos aún no se ha traducido en los resultados de negocio transformadores que espera el consejo de administración. Y los responsables de TI tienen que explicar el porqué.

Esta brecha no es un problema tecnológico. Es un problema de arquitectura. Tres preguntas se encuentran en el centro de todo esto, y la mayoría de las empresas aún no tienen respuestas claras para ninguna de ellas.

1. *¿Se ha vuelto más costoso el desarrollo de la IA que contratar más ingenieros?*
2. *¿Se puede confiar en el código generado por IA para operar procesos empresariales a gran escala?*
3. *¿Cómo se mantiene la visibilidad y el control sobre lo que la IA está creando e implementando?*



La cuestión del costo: ¿Está pagando más para acumular más deuda?

La promesa del desarrollo asistido por IA era clara: reducir el costo y el tiempo de creación de software. La realidad, para muchas empresas es que la estructura de costos ha cambiado de maneras que no estaban contempladas en el caso de negocio original.

Escribir código no es la parte más cara del desarrollo de software. Representa aproximadamente el 10% del ciclo de vida total de la aplicación. El 90% restante corresponde a revisión, pruebas, aplicación de parches de seguridad, depuración, refactorización y mantenimiento a medida que cambian las necesidades del negocio. Los asistentes de programación basados en IA han acelerado ese 10 %. Lo que los datos muestran cada vez con mayor claridad es que también han encarecido el 90% restante.

[El análisis realizado por GitClear en 2025](#) sobre 211 millones de líneas de código en repositorios de Google, Microsoft, Meta y organizaciones empresariales reveló un aumento de ocho veces en los bloques de código duplicados en proyectos asistidos por IA respecto a dos años antes. La rotación de código, es decir las líneas revisadas o descartadas en las dos semanas posteriores a su escritura, se duplicó entre 2020 y 2024. Como señaló Bill Harding, fundador de GitClear, el código generado por IA funciona como "una tarjeta de crédito nueva" para acumular deuda técnica: el gasto parece productivo en el momento, la factura llega después. [El informe State of Software Delivery de Harness de 2025](#) reveló que la mayoría de los desarrolladores ahora dedican más tiempo a depurar código generado por IA que a crear nuevas funcionalidades.

El impuesto de los tokens empeora aún más la situación. Cada línea de código que escribe una IA supone un costo en tokens. A diferencia de un token gastado en una interacción con un cliente, un token gastado en generar código de aplicaciones produce un activo que requiere una inversión continua para su mantenimiento. Si ese activo acumula deuda más rápido de lo que genera valor, la situación económica se torna negativa desde el primer día.

Además, la estructura de costo de las herramientas de desarrollo de IA ya no es predecible. GitHub Copilot adoptó una facturación basada en el consumo en junio de 2026, reemplazando su modelo de solicitudes premium de tarifa plana por créditos de IA basados en tokens. GitHub reconoció que "una pregunta rápida en el chat y una sesión de codificación autónoma de varias horas podían costar al usuario lo mismo" bajo el modelo anterior, y que absorber los crecientes costos de inferencia "ya no es sostenible" ([GitHub Blog, abril de 2026](#)). Las empresas que habían presupuestado el desarrollo de IA con tarifas planas ahora operan en un entorno de costos variables, con menos visibilidad sobre lo que gastan por cada flujo de trabajo.

La verdadera cuestión no es si merece la pena utilizar herramientas de desarrollo de IA. La cuestión es si el enfoque actual, utilizar la IA para generar más código con mayor rapidez, realmente reduce el costo total de desarrollo o simplemente adelanta la entrega, mientras acumula costes de mantenimiento que aparecerán en el presupuesto del próximo año.

La cuestión del rendimiento y la seguridad: ¿Se puede confiar en lo que construye la IA ?

Una entrega más rápida solo crea valor si se puede confiar en que lo que se entrega funcionará a escala empresarial. En este sentido, las pruebas están dando a los líderes de TI motivos legítimos para ser precavidos.

El análisis de CodeRabbit de 470 solicitudes de incorporación de cambios reales reveló que el código generado por IA introduce 1,7 veces más problemas totales que el código escrito por humanos, y específicamente 2,74 veces más vulnerabilidades de seguridad ([CodeRabbit State of AI vs Human Code Generation, diciembre de 2025](#)). [Una investigación del IEEE presentada en ISSRE 2025](#) descubrió que el código generado por IA presenta mayores índices de vulnerabilidades de seguridad de alto riesgo a pesar de producir resultados estructuralmente más simples, lo que sugiere que el riesgo reside en cómo se construye la lógica, no solo en su complejidad.

Las herramientas de IA generan soluciones a problemas locales e inmediatos. No tienen pleno conocimiento de la arquitectura general de su sistema, sus abstracciones internas, sus utilidades compartidas ni las restricciones de seguridad que rigen cómo se mueven los datos a través de su entorno. El resultado es un código que supera las pruebas unitarias y la revisión inicial, pero introduce vulnerabilidades que solo se manifiestan en casos extremos o bajo carga de producción.

El 76% de los desarrolladores que utilizan herramientas de programación basadas en IA reconocen haber generado código que no entendieron completamente al menos en algunas ocasiones ([Stack Overflow, 2026](#)). Esa estadística tiene un peso operativo real. Cuando un regulador, un equipo legal o una junta directiva le pregunta a un CIO por qué se produjo un comportamiento específico del sistema, "la IA lo generó y el desarrollador no estaba seguro de cómo funcionaba" no es una respuesta aceptable. Y sin embargo, esa es la postura implícita en materia de responsabilidad de cualquier organización que implemente código generado por IA en producción sin una capa de gobernanza estructural.

Las principales predicciones de Gartner para 2026 incluyen una previsión de que las demandas legales por "death by AI" (muerte por IA) superarán las 2000 a nivel mundial, impulsadas por una gobernanza insuficiente sobre los resultados de las aplicaciones generadas por IA ([Gartner, octubre de 2025](#)). Estos no son hipotéticos riesgos futuros. Representan un riesgo real para las empresas que hoy en día han otorgado al código generado por IA una autoridad significativa sobre los flujos de trabajo regulados o de cara al cliente.

La cuestión de la fiabilidad: ¿Sabe qué está construyendo la IA ?

La velocidad y la seguridad son aspectos operativos importantes. La auditabilidad es una cuestión estructural, y para los CIO y los arquitectos empresariales, puede ser la más difícil de implementar a posteriori.

Cuando una aplicación toma una decisión, ya sea enrutar un caso, determinar la elegibilidad, otorgar acceso o generar un resultado para el cliente, el departamento de TI es responsable de esa decisión, no la IA que escribió el código. El departamento de TI implementó la aplicación y los responsables la aprobaron.

En un entorno de desarrollo tradicional, esa responsabilidad es gestionable porque la lógica es inspeccionable. Cuando un auditor pregunta por qué se tomó una decisión, un desarrollador rastrea la ruta del código relevante y encuentra la respuesta. En un entorno donde la lógica de la aplicación fue generada por IA, esa trazabilidad no está garantizada. El código generado por IA produce resultados funcionales, pero el razonamiento detrás de las decisiones de implementación específicas no está documentado de la misma manera que lo haría un desarrollo realizado deliberadamente por personas. Cuando la pregunta de la auditoría es "¿por qué este sistema tomó esta decisión en este caso excepcional?", la respuesta "porque la IA lo programó así" no es una respuesta defendible.

Este es el problema fundamental de fiabilidad a nivel empresarial. No se trata de si el código se ejecuta correctamente en promedio. Se trata de si puedes explicar un resultado específico a cualquier persona que pregunte, en cualquier momento en el futuro, en un lenguaje que le permita actuar en consecuencia.

La lógica de aplicación visual y declarativa resuelve este problema estructuralmente. Cuando las reglas de negocio, las tablas de decisión y el enrutamiento del flujo de trabajo se expresan como una configuración inspeccionable en lugar de como código generado, cada ruta de decisión es rastreable sin necesidad de análisis forense. La lógica reside en un nivel que pueden comprender los equipos de cumplimiento normativo, los equipos legales y las partes interesadas de la empresa, no solo los ingenieros. La auditoría se convierte en una revisión de la configuración, no en un ejercicio de arqueología del código.

La arquitectura que responde a las tres

El cambio arquitectónico que resuelve simultáneamente los costos, la seguridad y la fiabilidad no supone una limitación para el uso de la IA . Se trata de una reorientación de donde se aplica la IA dentro del proceso de desarrollo.

La IA es realmente eficaz para acelerar la generación de estructuras de alto nivel de las aplicaciones: modelos de datos, definiciones de flujo de trabajo, tablas de decisión y lógica de enrutamiento expresada de forma declarativa. Se trata de artefactos que describen lo que hace una aplicación sin codificar su implementación en código imperativo. Son legibles, auditables y fáciles de mantener, algo que no ocurre con el código generado. Cuando cambian los requisitos del negocio, el modelo cambia, y el comportamiento de la aplicación cambia con él, sin las modificaciones en cascada del código ni el riesgo de regresión que esos cambios conllevan.

Este es el enfoque de desarrollo basado en modelos llevado a la práctica La IA realiza el trabajo para el que es realmente buena: traducir la intención en estructura, generar configuraciones a partir del lenguaje natural y acelerar la definición arquitectónica. El resultado es un modelo, no código. La generación de código, cuando es necesaria, se produce de forma subyacente como un detalle de implementación, no como el artefacto principal.

Las implicaciones operativas son concretas. Los equipos de desarrollo dedican menos tiempo al mantenimiento porque la superficie de mantenimiento es el modelo, no el código fuente. La revisión de seguridad es más manejable porque la lógica de negocio se expresa de forma declarativa. Las auditorías de cumplimiento normativo son más rápidas porque la capa de decisión se puede inspeccionar sin necesidad de asistencia técnica. Y el gasto en tokens se concentra en la parte de mayor valor del desarrollo: definir qué debe hacer la aplicación, no generar la mecánica de cómo lo hace.

Según [la encuesta Gartner I&O de finales de 2025](#), solo el 28 % de los casos de uso de infraestructura de IA cumplen plenamente las expectativas de retorno de la inversión . La brecha entre las expectativas y la realidad no se debe a una falta de capacidad. Es un fallo de la arquitectura. Las empresas que obtienen retorno de la inversión gracias a la IA en la fase de desarrollo no utilizan necesariamente modelos más avanzados. Están utilizando IA para crear artefactos que no se deprecian al mismo ritmo que el código generado automáticamente.



La decisión sobre la arquitectura

La decisión que deben tomar los líderes de TI no es entre velocidad y seguridad. Esa forma de plantearlo es lo que hace que la situación actual parezca de una dificultad insuperable. La elección real se plantea entre dos arquitecturas diferentes con perfiles de riesgo y costo a largo plazo muy distintos.

Una arquitectura centrada en código generado por IA ofrece una producción inicial rápida, pero con un mantenimiento progresivo, una capacidad de auditoría limitada y una exposición a la seguridad que aumenta con el tamaño del código base. Una arquitectura centrada en el desarrollo basado en modelos con IA integrada ofrece resultados iniciales que son auditables por diseño, fáciles de mantener en la capa del modelo y generan un gasto de tokens por parte del desarrollo que realmente crea valor duradero.

Los responsables de TI no tienen por qué negarse a la demanda del área de negocio de una entrega acelerada con IA. La respuesta no es más despacio. La respuesta reside en una filosofía de desarrollo diferente que el departamento de negocio pueda sostener y que el departamento de TI pueda defender.

La inversión adecuada en IA no consiste en escribir código más rápido. Se trata de crear modelos de aplicaciones que no necesiten ser reescritos.





Gracias

Acerca de Pega

Pega proporciona la plataforma líder impulsada por IA para la transformación empresarial. Las organizaciones más influyentes del mundo confían en nuestra tecnología para reimaginar cómo se hace el trabajo automatizando los flujos de trabajo, personalizando la experiencia del cliente y modernizando los sistemas heredados. Desde 1983, nuestra arquitectura escalable y flexible ha impulsado la innovación continua, ayudando a los clientes a acelerar su camino hacia la empresa autónoma.

[PEGA.COM/ES](https://www.pega.com/es)