



# El surgimiento de los codebots: Cómo la inteligencia artificial está cambiando el desarrollo de software

Ethan Mollick no es programador profesional ni pretende serlo.

No obstante, hace poco, pudo crear un programa para un videojuego sencillo pero jugable para celebrar el aniversario 30 de *Doom*.

¿Cómo? Mediante el uso de un chatbot (o bot conversacional).

La interacción fue tan sencilla como esta:

- chatgpt, crea para mí un juego interactivo que rinda homenaje a Doom, debe estar completo y ser jugable, pero dentro de tus habilidades. Proporcióname un archivo zip con html, css y Javascript (o adopta otro enfoque). Puedes llegar a ser sumamente creativo, además de muy curioso y genial.
- [La IA explica cómo se hace]
- no, tienes que hacerlo tú, no seas perezoso. Créame un juego que funcione. Puedes crear archivos descargables y puedes hacer todas las cosas que has dicho.
- [La IA crea un archivo zip]
- haz que sea más parecido a Doom.
- [La IA crea un archivo zip]
- esto está bien, ¿pero puedes hacerlo más parecido a Doom y con un diseño más lindo?
- [La IA crea un archivo zip] ...



**a inteligencia artificial te permite programar solo con la intención»,** afirma Mollick Mollick, profesor adjunto de Administración en la Wharton School de la Universidad de Pennsylvania y autor de [Cointeligencia: vivir y trabajar con la IA.](#)

El chatbot que usó Mollick es ChatGPT, que se basa en inteligencia artificial (IA) generativa. Se trata de una nueva y potente clase de algoritmos que pueden generar contenido nuevo (texto, música, vídeo o imágenes) y a la que, como demuestra la experiencia de Mollick, se le da bien programar software. Los «modelos de lenguaje de gran tamaño» (LLM), una categoría de modelos de aprendizaje profundo sobre la que se basan muchas herramientas de IA generativa, poseen una capacidad increíble para programar y completar código informático.

Gracias a una nueva gama de asistentes de programación de IA generativa, esta tecnología ya está convirtiendo la programación en más rápida y más productiva. «La IA supone un gran salto para la ingeniería de software», afirma Mike Mason, Director de IA en la consultora tecnológica global Thoughtworks. «La rápida adopción y el avance constante de las herramientas de IA para los desarrolladores de software pone de manifiesto el impacto transformador que ya está teniendo, y seguirá teniendo, en nuestro sector».

¿Pero cómo serán exactamente estos cambios y hasta dónde llegarán? ¿Traerán consigo alguna contrapartida que debamos intentar minimizar? ¿Y qué significará todo esto en última instancia para el papel de los desarrolladores y el tipo de software que crean?

A fin de cuentas, algunas personas creen que la IA

**“LA IA SUPONE UN GRAN SALTO PARA LA INGENIERÍA DE SOFTWARE”**

Mike Mason, Director de IA en la consultora tecnológica global Thoughtworks



generativa representa un punto de inflexión tecnológico tan importante como lo fue el paso de la interfaz de línea de comandos a la interfaz gráfica o incluso el nacimiento de Internet. Es posible que, como sugieren, cambie la informática tal y como la conocemos.

Pero, en primer lugar, vamos a analizar la situación actual...

**¡Hola, (valiente nuevo) mundo!**

**Hay un motivo por el que la IA generativa ha demostrado tener un talento natural para la programación.** La programación de software es una tarea que depende del lenguaje, aunque sea uno muy estructurado y basado en reglas, y ya existen repositorios enormes de código viable sobre los que pueden basarse los modelos.

En los últimos dos años, hemos visto el auge de herramientas especializadas que pueden completar código a partir de datos parciales en tiempo real o incluso generar secuencias de comandos desde cero. Son lo suficientemente inteligentes como para entender el contexto, con lo que pueden programar código que se ajuste a los requisitos del desarrollador y suelen incluir una interfaz de chat para hacer preguntas relativas a la programación.

«Como la función Autocompletar, pero elevada a la enésima potencia», así es como describe Mason a los asistentes de programación basados en IA. Eliminan gran parte del trabajo monótono de la programación, agilizando los procesos rutinarios, detectando errores y proporcionando recursos que permiten resolver problemas y localizar fallos sobre la marcha. «Hay veces que literalmente parece que me ha leído el pensamiento con respecto al bloque de código que quería programar», afirma Mason, que empezó a programar a los seis años.

Una ventaja esencial de estas herramientas es que pueden eliminar la necesidad de salir de la burbuja creativa que representa el entorno de desarrollo integrado (IDE) —el conjunto de aplicaciones de programación que usan los programadores— para buscar solución a problemas en los motores de búsqueda u otros sitios web. Salir del IDE puede hacer que un programador pierda ese «ritmo» tan importante; ese estado de inmersión completa en la tarea que tiene entre manos, que se asocia con el alto rendimiento.

**“ES UN ENORME POTENCIADOR DE LA EFICIENCIA”**

Andreas Nauertz, Director de tecnología y vicepresidente ejecutivo de Bosch Digital

Y están demostrando tener un gran éxito. En enero de 2024, una de las principales herramientas cuenta ya con 1,3 millones de suscriptores de pago en todo el mundo — un aumento del 30 % con respecto al trimestre anterior— y tiene como clientes a más de 50 000 empresas. Según una encuesta reciente, el 95 % de los desarrolladores ya está usando herramientas de IA para programar.

Los programadores están viéndolas cada vez menos como un extra opcional y más como algo necesario. «No es que nuestros desarrolladores me lo estuvieran pidiendo... ¡es que me lo estaban suplicando!», dice Andreas Nauertz, Director de tecnología y Vicepresidente ejecutivo de Bosch Digital. Su división pone en práctica soluciones digitales en todo el grupo de ingeniería y tecnología de Bosch. Tras evaluar los posibles riesgos y ventajas, la empresa contrató en diciembre del año pasado una importante herramienta de codificación de IA y la está utilizando como asistente de programación en muchos de sus proyectos. «Es un impulsor de la eficiencia enorme», comenta. «Y hace que los empleados estén más contentos porque pueden centrarse en problemas reales y no solo en las tareas cotidianas aburridas que hay que hacer».

Los datos sugieren que las herramientas de finalización de código se consideran más bien proveedoras de sugerencias útiles que hay que mejorar, en vez de programadoras de código definitivo. Aun así, las ventajas son cuantificables. Un estudio llevado a cabo por una consultora de gestión determinó que el 75 % de los programadores que habían adoptado asistentes de programación basados en IA afirmaba que las herramientas habían cumplido o superado sus expectativas y consideraba la «rapidez de comercialización» como su mayor ventaja. En una encuesta realizada por un proveedor líder en el mercado a más de 2000 desarrolladores, el 88 % respondió que su productividad era mayor al usar el asistente de programación basado en IA. Un artículo exhaustivo en el que se analizaba cómo medir el modo en que la IA que completa código afecta a la productividad llegó a la conclusión de que, aunque «la productividad de los desarrolladores es una cuestión pluridimensional que no puede resumirse con una sola métrica», las ventajas notificadas de la sugerencias de programación basadas en IA «abarcan toda la gama de aspectos que suelen investigarse en relación con la productividad, como el tiempo dedicado

a la tarea, la calidad del producto, la carga cognitiva, el disfrute y el aprendizaje» y destacó que «los incrementos de productividad percibidos se reflejan en métricas objetivas de la actividad de los desarrolladores».

## Quando la IA no es tan inteligente

**La velocidad constituye sólo una pequeña parte de lo** que significa ser un buen ingeniero de software y aquí es donde se complica la cosa. Una investigación reciente llevada a cabo por una empresa de análisis de código llegó a la conclusión de que la calidad del código ha descendido desde la introducción de los asistentes basados en IA. De hecho, un estudio de referencia realizado por otra empresa de análisis de código analizó la refactorización, es decir, el proceso de mejorar el diseño de un código previo sin cambiar su comportamiento. Constató que las soluciones de IA existentes solo producen refactorizaciones funcionalmente correctas en el 37 % de los casos.

«Una de las contrapartidas a la utilidad de los asistentes de programación es su falta de fiabilidad», afirma Birgitta Böckeler, responsable global de entrega de software basada en IA de Thoughtworks. «Los modelos subyacentes que usan son bastante genéricos y se basan en una cantidad enorme de datos de entrenamiento que no son siempre relevantes para la tarea en cuestión. Y los modelos de lenguaje de gran tamaño también se inventan cosas, es decir, tienen «alucinaciones».

Lo preocupante es que los errores que comete la IA son sutiles y, por lo tanto, difíciles de detectar. Aunque el código que genera debería considerarse un punto de partida, puede parecer correcto y convincente aunque contenga errores. Por lo tanto, si un desarrollador no está atento, puede pasar por alto errores. Esto explica la paradoja de estas herramientas. Aunque pueden hacer que la programación sea más accesible para las personas con menos experiencia, precisamente los que tienen menos experiencia son los que pueden convertirse en víctimas de sus defectos.

«Me preocupa darle asistentes de programación a los desarrolladores sin experiencia», dice Rebecca Parsons, Directora emérita de tecnología de Thoughtworks. «Alguien que lleve programando tantos años como yo, puede detectar los pequeños errores de lógica y cosas tan sutiles como

esa, pero no es tan fácil si no se tiene experiencia. Estos modelos de programación realmente no saben lo que es un buen código, así que existe el riesgo de que terminemos teniendo una gran cantidad de código de mala cantidad con más errores».

Los problemas de seguridad son evidentes. La investigación académica ha constatado que los programadores que usan asistentes de IA no solo generan código bastante menos seguro que los que no los usan, sino que también es mucho más probable que crean que están generando código seguro. La IA puede producir alucinaciones a la hora de corregir vulnerabilidades comunes y omitir comprobaciones de validación esenciales, lo que expone los sistemas a los piratas informáticos. Y puesto que los programadores más jóvenes dependen de estas herramientas cada vez más, esto también podría estar gestando un problema aún más importante de seguridad de cara a los próximos años. «Creo que hay un enorme riesgo de que se degraden las competencias de programación en el futuro», comenta Nauerz. «Cuantas más personas dejen de tener un conocimiento profundo sobre el funcionamiento de los sistemas, más perderemos el control sobre estos y más expuestos estaremos a nuevos vectores de ataque».

Además, hay que tener en cuenta una cuestión entre ética y jurídica. En los últimos meses, se ha hablado en los medios de ejemplos de IA entrenada con datos protegidos por derechos de autor. Y los asistentes de programación ahora también están en tela de juicio. Uno de los proveedores líderes en el mercado ha sido objeto de una demanda colectiva, en la que se describen sus acciones como «piratería de software a un nivel nunca visto antes». Se refiere a las largas líneas de código con licencia que estos asistentes pueden repetir sin dar el crédito correspondiente. No obstante, esta causa se encuentra en una fase inicial.

Es evidente que, a medida que los programadores se aventuren más en este nuevo mundo, será indispensable mitigar los riesgos. Las empresas tendrán que tener políticas claras respecto al uso de la IA -manteniendo informadas a las personas y sin usar modelos que incorporen datos de instrucciones- y los desarrolladores deberán adoptar una mentalidad de sano escepticismo en lo que respecta a la revisión de los resultados de la IA.





Cuando Böckeler empezó a trabajar con un asistente de IA, decidió darle una personalidad, como si se tratara de un compañero nuevo. Esto le ayudó a concretar el enfoque que debía adoptar al trabajar con él. Lo definió como: deseoso de ayudar, obstinado, erudito pero sin experiencia y poco dispuesto a admitir que no sabía algo. A continuación, introdujo esta descripción junto con distintos personajes animales en el generador de imágenes de IA Midjourney, que creó una imagen de un burro sonriente y algo frenético, rodeado de libros.

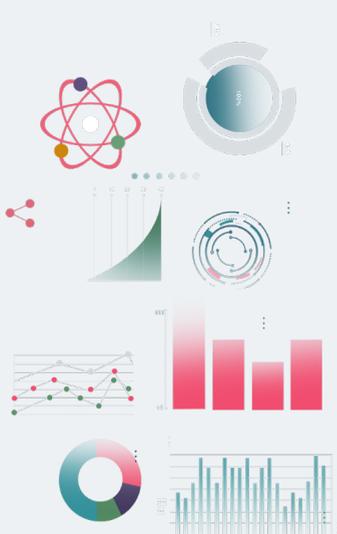
## ¿Qué será lo siguiente? De secuencias de comandos a sistemas

### Los programadores hacen mucho más que crear código.

En promedio, la programación ocupa menos de un tercio de su tiempo. El resto lo dedican a una variedad de tareas como la realización de pruebas, la depuración de errores, hablar con los usuarios finales, diseñar nuevas soluciones o adquirir nuevas competencias. Y, luego, están las tareas creativas de nivel superior: ¿qué se debe conseguir realmente con el software ante todo? Y, ¿está enfocando esto el cliente de la manera adecuada?

«La IA funciona muy bien en los asistentes de programación y actualmente va camino de proporcionar sugerencias cada vez más relevantes», comenta Amjad Masad, fundador y consejero delegado de Replit, una plataforma de programación asistida por IA. «Pero la IA todavía está en una fase muy temprana en lo que respecta a qué funciones queremos crear, la estructura del producto, las pruebas de integración y los componentes y cosas similares».

En el futuro, es probable que esto se convierta en un área de desarrollo. Thoughtworks ha llevado a cabo varios experimentos recientemente a lo largo del ciclo de vida del desarrollo de software para ver dónde podría la IA generativa marcar una diferencia significativa. Los resultados iniciales han mostrado un aumento en la productividad del 10 % al 30 %, y no solo en la programación. En China, han usado la IA generativa como mecanismo para poner en común los conocimientos, permitiendo sintetizar todo el contexto de un proyecto para determinar la inteligibilidad. La empresa también ha usado la IA generativa como herramienta de conceptualización, creándola de tal forma que pueda proporcionar ideas en un espectro que va de conservador





# “TENEMOS QUE REINVENTAR EN GRAN MEDIDA EL MODO EN QUE CONCEBIMOS LOS SISTEMAS”

Ethan Mollick, Profesor adjunto de administración de la Wharton School

a pragmático y a completamente creativo.

Bosch Digital también está analizando esta cuestión. Nauertz cree que la IA no solo mejorará la gama completa de procesos existentes, sino que también será la base de los procesos que necesitaremos en el futuro, como la detección de sesgos. «Es un reto específico de la IA», comenta, «y necesitaremos una herramienta nueva para ello».

A medida que la IA generativa penetre en el desarrollo de software, tendrá diversas consecuencias para los departamentos de TI y, en última instancia, el consejo de administración. La más inmediata es: «¿qué pasa con todo el tiempo que se ahorra?». Quizá se reasigne a esas tareas creativas o se dedique a resolver los problemas relativos a seguridad o es posible que se deje sencillamente que los equipos hagan más cosas de forma más rápida. Todas las organizaciones tendrán distintas prioridades pero es evidente que será importante meditarlo bien.

La dirección también tendrá que resistirse a verlo como la panacea en términos de productividad. Aunque parezca contrario al sentido común, una programación más rápida puede crear cuellos de botella en los procesos de desarrollo. «La mayoría de las organizaciones tiene un ciclo bastante lento de realización de pruebas y lanzamiento a fin de verificar que el software es correcto y pasarlo finalmente a producción», afirma Mason. «Por lo tanto, si se tiene un ciclo lento de realización de pruebas y lanzamiento pero se está generando código rápidamente, en realidad, van a empeorar las cosas porque habrá más componentes que publicar en cada versión y, por tanto, las versiones serán más grandes y el riesgo será mayor».

Además, se plantean preguntas más amplias sobre la organización. El desarrollo de software actualmente se basa en la metodología ágil (o Agile), un término que hace referencia tanto a un conjunto de valores y principios como a metodologías asentadas de gestión de proyectos e ingeniería. Estas metodologías pueden adoptar distintas formas pero tienden a hacer hincapié en la iteración rápida por parte de equipos interdisciplinarios, con una evaluación continua por parte del cliente. La introducción de la IA podría cambiar estas metodologías. «Todos los métodos que usamos para organizar el trabajo solo tienen en cuenta un tipo de inteligencia, que es la inteligencia humana, pero

esto ya no es así», afirma Mollick. «Por lo tanto, tenemos que reinventar en gran medida el modo en que concebimos los sistemas». Esto significa que es probable que los procesos ágiles cambien si, por ejemplo, la IA generativa puede recopilar los requisitos del cliente o si podemos hablarle de un producto y preguntarle qué pensaría de él si fuera un tipo específico de persona. Dentro de este paradigma, es posible que las metodologías ágiles de repente no nos parezcan nada ágiles.

## Visión de futuro

**¿Llegará algún día un punto en el que cualquiera pueda crear software con la IA generativa, independientemente de si conocen o no algún lenguaje de programación? Y, si ese es el objetivo final, ¿cómo sería el camino hasta llegar a ese punto?**

Un paso intermedio sería tomar el concepto de asistente de programación basado en IA y darle autonomía para actuar. Los «agentes de IA» -chatbots que pueden usar aplicaciones, completar tareas y llevar a cabo procesos de varias fases de forma autónoma- son un tema candente actualmente en el sector, y los principales actores están invirtiendo grandes sumas para convertir el concepto en realidad. Amjad Masad, de Replit, cree que tienen un potencial enorme para los desarrolladores de software, dado que empresas incipientes ya están creando agentes diseñados para gestionar los procesos de ingeniería, desde la creación del código hasta la depuración de errores y el lanzamiento. Masad prevé que llegará un momento a relativo corto plazo en el que los programadores se convertirán en gestores de proyectos «cuasidiosos» que operarán a un nivel de arquitectura y diseño y «dirigirán ejércitos de agentes de software para crear software cada vez más complejo en plazos demencialmente cortos».

A más largo plazo, es posible que estos agentes también puedan gestionar metas más elevadas: en vez de darle a la IA una imagen definida de lo que queremos que haga el software, podríamos decirle qué función queremos que desempeñe en última instancia y todo lo demás se generaría automáticamente a partir de eso. Si una herramienta como esta pudiera generar incluso el software más complejo a nivel de producción, el concepto de «programación» tal y como

| lo entendemos actualmente acabaría por dejar de existir.

## Las preguntas importantes

### 1. ¿Llegará la IA a ser realmente tan buena?

**Este debate ha entrado en el ámbito de lo desconocido** y plantea varias preguntas profundas. Aquí tenemos tres de ellas:

La posibilidad de que los modelos de lenguaje de gran tamaño transformen la naturaleza del desarrollo de software en la máxima medida descrita aquí depende de si dichos modelos pueden seguir dando pasos agigantados o no en el desarrollo de sus habilidades. Probablemente, para que las herramientas funcionen a este nivel hipotético, tendrían que aproximarse a la inteligencia de nivel humano. Hay quienes creen que esto se producirá de forma espontánea a medida que los modelos de lenguaje de gran tamaño adquieran más datos y más capacidad de computación. Pero hay otros que son más escépticos. Por ahora, es imposible decir si será lo uno o lo otro. «Llegará un momento en que la IA llegue a un límite y cuando alcance dicho límite, la pregunta será: ¿es este el final?», afirma Mollick. «¿O todavía queda mucho más margen que nadie se está molestando en desarrollar porque todos se están apresurando a crear modelos más grandes? Todavía no tenemos las respuestas».

### 2. ¿Llenará esto el mundo de software de baja calidad?

Existe la preocupación de que terminemos en un círculo vicioso en el que la IA genera una gran cantidad de código no seguro y lleno de errores que pasa a formar parte del acervo de código en el que se basan otros modelos de lenguaje. Böckeler no le quita importancia a esta preocupación. «Dependemos mucho del software y esto es algo que a veces me inquieta», comenta. «Lo ideal es que la IA nos ayude a producir software mejor y más fiable, ¿pero qué pasaría si empeorara las cosas porque en teoría cualquiera podría crear código? Siempre les decimos a nuestros clientes que la IA lo amplifica todo de forma indiscriminada, así que hay que tener cuidado con dónde se aplica».





### 3. ¿Qué significa la IA generativa para la propia esencia del software?

El concepto básico del software se ha mantenido invariable desde los albores de la informática. Si se quiere realizar una tarea, hay que comprar o crear una herramienta específica que realice dicha tarea. La diferencia respecto a la IA generativa es que esta puede realizar una amplia gama de tareas con distintas modalidades de datos de hecho, algunos piensan que podría llegar a realizar cualquier tipo de tarea. ¿Significa esto que puede acabar siendo lo suficientemente inteligente como para funcionar como una «superaplicación» universal es decir, una única interfaz que utilicemos para hacerlo todo? De ser así, cabría imaginar que tendríamos menos software, en lugar de más, y la cuestión clave para los desarrolladores sería determinar cómo debería ser la experiencia del usuario y cuál sería la mejor manera de crearla. De nuevo, ¿sugiere la historia que este sea un resultado realista? ¿No se desagregarán los modelos de lenguaje de gran tamaño en herramientas para realizar tareas específicas tal y como se hizo con las hojas de cálculo en aplicaciones especializadas como el software de contabilidad y CRM?

«Es posible que el grado de cambio dependa de si la IA generativa hace que las interfaces por voz despeguen o no», sugiere Mike Mason. «Los “asistentes inteligentes” digitales que tenemos hoy en día en nuestros móviles y hogares son inteligentes en cierto sentido, pero muy limitados en otro. Con los avances en la IA es posible que se conviertan en nuestro principal mecanismo para interactuar con el mundo», comenta Mason. «En vez de usar la aplicación de una línea aérea, podríamos simplemente pedir a nuestra IA personal que reserve vuelos, haga la facturación o encuentre nuestro equipaje perdido. Esto, sin duda, sienta las bases para una revolución en cómo será el software y cómo interactuaremos con él».

## El camino por delante...

**La IA generativa es el centro de gravedad de la tecnología** en la actualidad pero no ha ocupado este puesto durante mucho tiempo y está evolucionando muy rápidamente. Es un ámbito que todavía se define más por preguntas que por respuestas. Parece seguro decir que el puesto de desarrollador de software cambiará y que sus competencias se usarán cada vez más a un nivel superior de abstracción

# “VA A SEGUIR HABIENDO UNA FUNCIÓN PARA LOS SERES HUMANOS. AL MENOS, DURANTE MUCHO TIEMPO”

Rebecca Parsons, Directora emérita de tecnología de Thoughtworks

en el proceso de desarrollo, ¿pero de qué maneras y en qué medida? El tiempo lo dirá.

Por supuesto, lo que más preocupa a estos profesionales es, en esencia, si seguirá existiendo el cargo de «desarrollador de software». Rebecca Parsons es optimista.

«El mundo va a seguir cambiando y nos vamos a enfrentar a nuevos retos», comenta. «Pero lo que tenemos que entender sobre estos modelos de IA generativa es que trabajan efectivamente sobre lo que ha ocurrido en el pasado».

Por lo tanto, cuando surjan situaciones nuevas en el futuro y haya que representar en código algo que nadie ha visto antes, ¿a quién vamos a llamar?

«La capacidad que tenemos como desarrolladores de software es imaginar cómo trazar lo que está en el mundo real y las abstracciones que se necesitan», afirma Parsons. «Va a seguir habiendo una función para los seres humanos. Al menos, durante mucho tiempo».

## WIRED Consulting.

WIRED es donde se hace realidad el mañana. Es la fuente esencial de pensamiento libre y conocimientos avanzados sobre las tendencias tecnológicas, científicas y sociales que están cambiando el mundo. Consulting es una división de WIRED que acerca la red, los conocimientos y la marca exclusivos de WIRED a las organizaciones comerciales, ayudándolas a generar conocimientos internos, a desarrollar estrategias y a crear contenidos especializados que los posicionen a la vanguardia.

## /thoughtworks

Thoughtworks es una consultora tecnológica global que integra estrategia, diseño e ingeniería para impulsar la innovación digital. En Thoughtworks somos más de 10 500 thoughtworkers en 48 oficinas de 19 países. Durante 30 años, hemos logrado un impacto extraordinario junto con nuestros clientes, ayudándoles a resolver