

Tabla de Contenidos



- Prompts y tiempos perdidos
- Las vueltas hacia atrás
- Los reviews de código
- Asumir más carga
- Codificación y desarrollo
- Nuestro grave problema
- Conclusiones

IA y Productividad . En estos días todos nos vemos afectados por la llegada de la IA a las tareas de codificación. En muchos casos, cuando programamos, tenemos la sensación de que la IA es de gran ayuda, sobre todo en un momento inicial, al construir componentes sencillos. Por eso, mucha gente opina que es razonable mejorar entre un 30 % y un 35 % los tiempos de codificación.



Prompts y tiempos perdidos

Sin embargo, a medida que vamos profundizando, nos damos cuenta de que aparecen más herramientas, cada una con sus propios comportamientos y formas de generar código eso en principio es positivo. Sin embargo en más de una ocasión me he encontrado intentando obligar a la IA a que me devuelva algo concreto, y simplemente no es capaz de entender lo que solicito.

Poco a poco uno va dando tumbos de un lado a otro y se da cuenta de que alcanzar ese 35 % depende de muchas variables:

- primero, de lo que preguntes.
- segundo, de la tecnología que uses.

- tercero, de lo que ya tengas previamente construido.
- cuarto del modelo de IA que uses.
- quinto del conocimiento que tengas

Sin lugar a dudas, es un tema complejo. ¿Podemos llegar a ese 35 %? 40% Está claro que dependerá del contexto, y el contexto tiene un número de variables que parecen infinitas. No solo influye lo anterior, sino también el tipo de trabajo: no es lo mismo construir una clase básica o un test sencillo que algo más complejo. También cuenta nuestra experiencia como programadores y nuestra experiencia en prompting que va a convertirse en fundamental.

Las vueltas hacia atrás

Con el paso del tiempo, uno ve claramente que, en más de una ocasión, puede tocar volver hacia atrás. Y empieza a intuir que quizá alcanzar esa ansiada mejora del 35 % en productividad es cuestionable, especialmente cuando dejamos de trabajar con tareas muy básicas dentro de nuestro negocio y abordamos temas complejos. Ese es el primer aviso a navegantes y construir buenos prompts se hace cada día mas importante y tener un conocimiento “conceptual” empieza a ser critico.

Los reviews de código

Ojalá los prompts fueran nuestro mayor problema... pero no lo creo. El verdadero problema aparece cuando asumimos una gran cantidad de código generado por IA y, a medio plazo, descubrimos que resulta problemático y no encaja con nuevas piezas que vamos construyendo.

Ya no solo se trata solo de dar marcha atrás —que ya sería complicado—, sino de enfrentarse a una situación en la que toca modificar el código de una “máquina”. Un código que, curiosamente, puede ser fácil o instantáneo de generar, pero complejo de analizar y mantener algo que a veces sorprende.

Así que toca arremangarse y decidir, como *Senior Developer*, ¿qué hacer con el lío que tenemos?. ¿Puede ayudarnos algún junior? Probablemente no: primero, porque todavía no tiene la experiencia suficiente; y segundo, porque en algunas empresas ya ni siquiera existen todo el mundo quiere seniors. Las herramientas de IA nos ayudarán pero el conocimiento de como enfocar será fundamental.

Asumir más carga

Asumiremos más carga, no nos quepa duda. Y más carga implica más cansancio. A veces resulta paradójico que dispongamos de una herramienta que hace trabajo por nosotros y, aun así, terminemos más exhaustos que antes para cumplir con los nuevos tiempos de desarrollo aceptados. . Estos nuevos tiempos pueden traer un mayor porcentaje de Burnouts.

Si algo he aprendido en consultoría y en innovación es que, al principio, todo el mundo es muy optimista ya quieren ser los primeros en decir que el ahorro será colosal. El problema es que, en este caso, el optimismo puede salir caro ya que asumir un ahorro fuerte puede implicar pérdidas en el futuro.

Codificación y desarrollo

Me sorprende cómo muchas personas confunden el mundo del desarrollo con el mundo del código. Una cosa es codificar —que es solo una parte del desarrollo— y otra muy distinta es el desarrollo completo.

¿Te va a acompañar la IA a las reuniones de seguimiento?

¿Va a hacer la toma de requerimientos por ti?

¿Va a realizar el análisis por ti? ¿ Va a solventar un bug complejo?

Me temo que no.

¿Cuánto representa realmente la codificación dentro del esfuerzo total de un proyecto?

Probablemente entre el 40 % y el 50 %. El resto corresponde a otro tipo de tareas, dependerá de tu contexto.

Nuestro grave problema

Estamos empezando a ver que, incluso en tareas de codificación, alcanzar ese 30- 40% % de mejora no es tan trivial. Dependiendo de tu *stack* tecnológico, quizá una mejora realista esté más cerca del 15 % o el 25 % que del 35 % que a veces se anuncia. A veces es diferente que el tiempo de construcción del código del porcentaje de código construido.

El problema surge cuando hacemos números. Si en un producto gastamos 1.000 horas y queremos decirle al cliente que reduciremos el tiempo total del proyecto un 30 %, debemos hacer algunos cálculos sencillos pero que a veces asustan:

1.000 horas antes → 700 horas ahora

Horas de codificación: 400 (40%)

Si queremos entregar el proyecto en 700 horas, esas 400 horas de codificación tendrían que reducirse prácticamente a 100. Eso implicaría una mejora del 75 % en codificación. Es una reducción muy importantemucho.... muy alta.

¿Es realista?

¿Y las otras 600 horas? ¿Son también mejorables por la IA? No lo sé. No tengo números tan exactos pero es claro que si lo son sus mejoras son menores que en la propia área de codificación.

Conclusiones

Estos números son generales; cada caso es un mundo. Quizá la conclusión más sensata sea esta: no estaría mal ser muy prudentes en la estimación de nuestros desarrollos cuando hablamos de mejoras basadas en IA ya que las herramientas avanzan mes a mes. Habra

cosas que nos quedemos con un 20-25% y habra cosas que puedan llegar mucho más lejos.

¿Y tú? ¿Cuál ha sido tu experiencia?