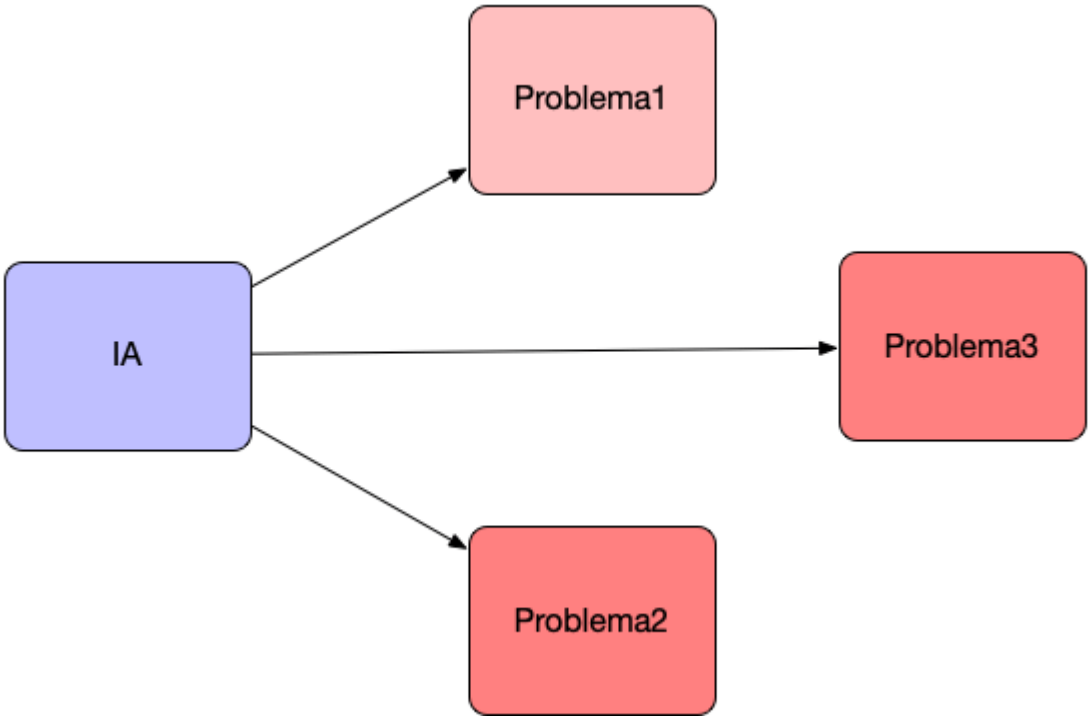
La mayoría de nosotros cada día usamos más las herramientas de generación de código por parte de la IA ya sea Copilot , Claude, Cursor, Codex o la que corresponda a cada uno . Son herramientas muy útiles y la velocidad de desarrollo se multiplica por X siendo X algo relativo a tu contexto.



Mis Experiencias y mis problemas

Mis experiencias son principalmente positivas ya que hoy por hoy construyo muchísimo más rápido. Sin embargo eso no quiere decir que todo sea perfecto . En muchas ocasiones me he encontrado en situaciones con problemas , con implementaciones que se alejaban de lo que a mi me gustaría o con soluciones que no son realmente lo que necesito.

Es curioso porque mucha gente comenta que usa la IA funciona perfectamente y lo hace todo muy bien . Mi opinión es más bien que la IA funciona y lo hace todo. Es parecido pero no es lo mismo lamentablemente .

Lo primero que quiero destacar es que es una herramienta con la que no hay vuelta atrás desde mi punto de vista.¿Ha llegado para quedarse? . Mas bien .. simplemente “se queda”

no me planteo otra cosa . ¿Cuales han sido mis experiencias y mis problemas ?Han sido varios y muy diversos vamos a categorizarlos.

Cosas Sencillas : Hay a veces que me he encontrado con temas como por ejemplo que no ha usado Lombok muy bien o que quizás un test se puede construir de una forma más clara o ha usado con Spring Data muchos query methods. Son problemas que a veces nos pueden llevar a desarrollos con problemas puntuales pero eso NO quiere decir que el código no funcione. Mas bien al contrario las soluciones funcionan y eso es quizás una de las cosas más preocupantes … todo parece funcionar.

Situaciones complejas : Son otros casos , no son muchos pero si me he encontrado con cosas como que la implementación tenía varios Ids en varias entidades y a nivel de REST se lio con ellos cuando se construía un api REST con relaciones. Fue curioso porque le costo mucho solventarlo … a la propia IA ya que el error que salía no ayudó al ser un error transaccional. En esta situación hubo que arremangarse bastante y me volvió a dejar claro "que el conocimiento es clave para gestionar la IA" cuando las cosas se ponen feas.

Situaciones absurdas: A veces en la construcción de código me he encontrado cosas que funcionan y que la implementación es no se si decir terriblemente mala o simplemente no lo he explicado bien . Me paso hace poco con una situación en las que podíamos tener varios niveles de complejidad en un campo de una entidad y ni se molesto en considerar una tabla parametrica en la base de datos directamente genero un enum cerrado.

Aumentar la complejidad : En otras situacion me he encontrado con código demasiado complejo para solventar un tema que no era para tanto por ejemplo usar rápidamente EntityGraphs en unas queries.

Mi experiencia , mis conclusiones

Después de todo lo que he probado … tengo varias conclusiones interesantes.

- En primer lugar que probablemente una parte de los errores son culpa mía ya que quizás los

prompts fueron poco concretos. Es algo que a veces a uno se le olvida y dependiendo de lo que tenga configurado en su arnes pueden quedar flecos por cubrir.

- En segundo lugar que el conocimiento “importa” ¿cuanto importa? . Mucho y mucho más que antes ¿Por que? porque generamos un volumen mucho más grande de código y tenemos que orientar a la IA y revisar lo que produce.
- En tercer lugar pero la más importante … con la IA … todo parece funcionar en un primer momento… los problemas vienen más adelante ya que los fallos son muy muy sutiles y sino tienes experiencia suficiente los errores se pueden colar a todos los niveles desde rendimiento hasta seguridad.
- La revisión pasa a ser una tarea fundamental y agotadora en muchos casos.
- ¿Cuanto vamos a necesitar a los desarrolladores Senior? . Infinito y el tiempo lo dirá ¿Que experiencias has tenido tu?

- Certificaciones Java
- ¿JPA vs Hibernate?
- ¿Como aprender en el mundo de la IA?
- ¿Y si necesitáramos más programadores?