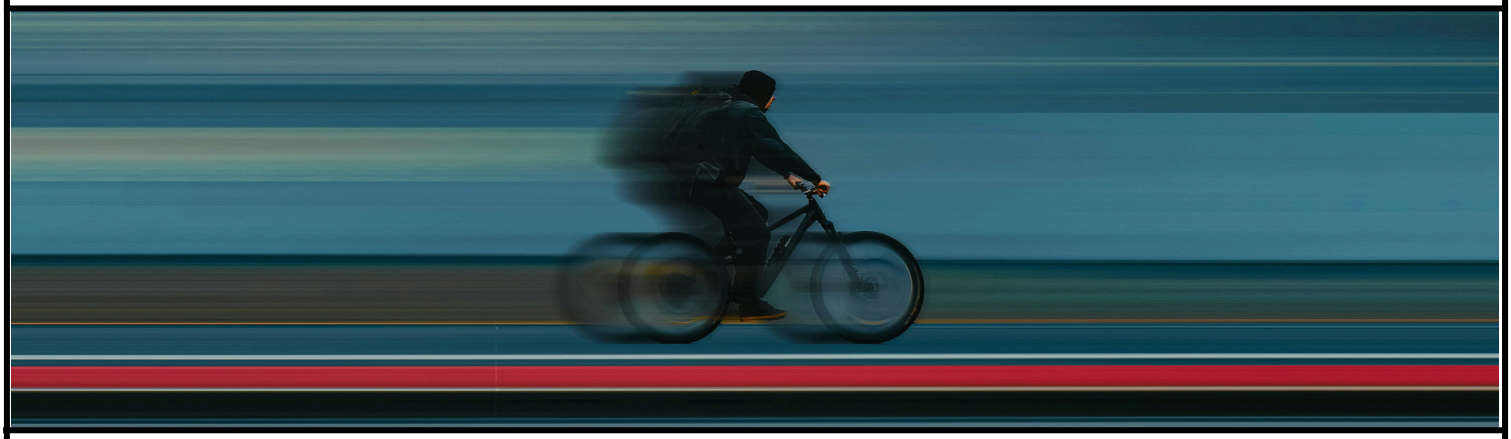
Incubator



¿QUÉ PASA CUANDO GANAMOS VELOCIDAD?

IMPACTO REAL DE LA IA EN EQUIPOS DE DESARROLLO DE SOFTWARE

RESUMEN EJECUTIVO

Impacto real de la IA en equipos de desarrollo de software
La IA está transformando el desarrollo de software. En 2025 en Incu tomamos la decisión consciente de integrar inteligencia artificial en nuestro flujo de trabajo técnico. La hipótesis era clara: automatizar tareas, acelerar entregas y elevar la productividad con IA.
Los resultados iniciales fueron prometedores: menos tiempo para escribir código, menos bloqueos y un flujo de trabajo más ágil. Pero cuando abrimos esta conversación a la comunidad, muchos mencionaron la afectación a su motivación como una externalidad inesperada de la automatización y el tener una computadora "pensando" por ellos.
Este artículo explora ese dilema y lo que aprendimos desde la experiencia. Si estás liderando un equipo Dev, trabajando en una implementación de IA o pensando cómo escalar tu organización sin comprometer calidad y talento, quizás esto te puede ayudar a orientar algunas elecciones.

IA PARA ESCRIBIR CÓDIGO: ¿QUIÉN PUEDE HACERLO REALMENTE BIEN?

Para facilitar la adopción de IA en desarrollo de software, diseñamos un modelo donde perfiles técnicos con visión analítica primero generaban código con herramientas de inteligencia artificial y evaluaban la calidad del output. Se hizo imperante entonces crear frameworks de trabajo para las IA donde estipulamos tecnologías preferidas y buenas prácticas de desarrollo que están en nuestro ADN pero que la IA no conocía. CONTEXTO es la palabra clave y que define el output.

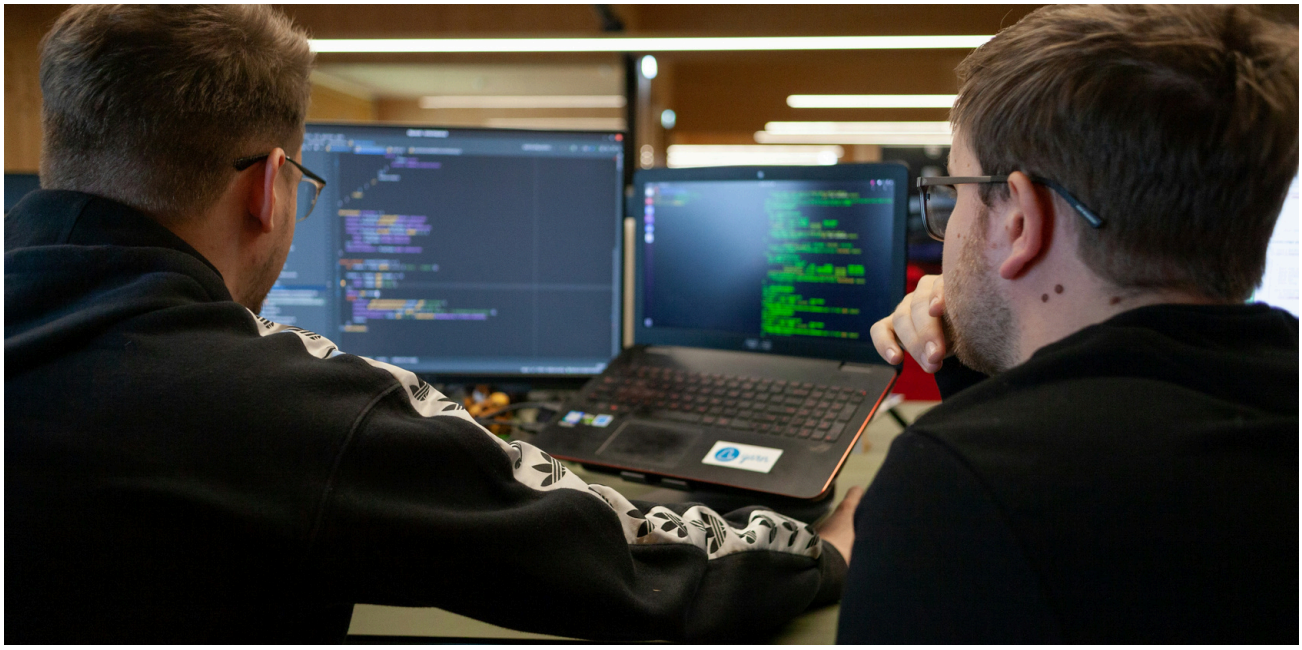
Luego empezamos a incorporar capas, por ejemplo usando **Taskmaster para ayudarnos a gestionar** las tareas de desarrollo, descomponiendo un Documento de Requisitos de Producto (**PRD**) en tareas más pequeñas. Algo que ya hacían los devs, pero con el beneficio de que ahora se hace en segundos y genera código a partir de estas tareas. Esto no solo reduce los tiempos de desarrollo aproximadamente a la mitad y mejora la calidad del software. Y aca muchos nos dijeron “suerte revisando ese código” pero la verdad es que:
1. La mayoría de las veces la comprobaciones es si hace o no lo que se programó para hacer y 2. Cuando tenes devs profesionales, seniors o semi seniors que están acostumbrados a leer código, los errores un poco te pegan en el ojo, ¿no? Con esto el rol de nuestros desarrolladores evolucionó hacia el control y validación de la salida de la IA. Eso sí, la validación humana del PRD es FUN-DA MEN-TAL.

IA PARA ESCRIBIR CÓDIGO: ¿QUIÉN PUEDE HACERLO REALMENTE BIEN?

Los resultados iniciales mostraron una mejora concreta: menor tiempo en tareas repetitivas, ciclos de entrega más cortos, menor fricción en la operación diaria.
Sin embargo, surgió una pregunta clave: ¿quién puede realmente escribir código de calidad profesional con ayuda de IA?
Bueno, un poco lo mencione arriba pero vamos a limpiar un poco.
Para mi hay tres factores principales:

- Perfil del desarrollador: Su criterio, familiaridad con conceptos técnicos fundamentales y comprensión del dominio del negocio.
- Madurez organizacional: Cultura de revisión de código, procesos sólidos y métodos de verificación.
- Tipo de tarea: No es lo mismo generar una función simple que tomar decisiones de arquitectura.

Como dijo uno de nuestros Devs más experimentados: “Cualquiera puede usar IA para generar código, pero escribir buen software todavía requiere criterio técnico, experiencia y conciencia de impacto.”



Incubator

LO QUE DICE LA COMUNIDAD TÉCNICA: TRES PERSPECTIVAS RECURRENTES

En las conversaciones con otros equipos y líderes técnicos, surgieron puntos de vista que vale la pena destacar:

1. Evolución natural de la programación
Muchos coinciden en que la IA no reemplaza el desarrollo, sino que representa una nueva capa de abstracción. Como ocurrió antes con lenguajes de alto nivel, el foco cambia: se automatiza lo repetitivo, y se sube el nivel de los problemas que resolvemos.

2. Riesgos de calidad a mediano plazo
Varios arquitectos alertaron sobre un problema silencioso: la IA puede generar código funcional, pero muchas veces introduce deuda técnica, mal diseño, o acoplamientos innecesarios. Los síntomas no se ven de inmediato, pero emergen cuando hay que escalar o mantener ese código.

3. Formación técnica debilitada
Un punto crítico: si las nuevas generaciones de desarrolladores comienzan delegando desde el inicio, ¿cuándo desarrollan criterio técnico? Validar el código que genera la IA requiere entender bien los fundamentos. Pero esos fundamentos, históricamente, se adquieren escribiendo código.

PILOTOS Y DESARROLLADORES: UNA ANALOGÍA CLAVE

Una comparación que resonó en nuestras discusiones fue con la aviación. Los pilotos ya no vuelan todo el tiempo: gran parte del vuelo está automatizado. Pero no por eso son menos competentes. Se entrenan en tomar decisiones, supervisar sistemas y resolver situaciones de alto riesgo. En el desarrollo de software podría pasar algo similar. El rol del Dev no desaparecería sino que evolucionaría: se volvería más estratégico, más enfocado en arquitectura, diseño de sistemas y toma de decisiones. Pero cabe preguntarnos, ¿son roles equivalentes? Los pilotos operan sistemas y los Devs los crean. Las tareas son en sí diferentes. Pero se puede pensar que sí hay algo que los devs pueden incorporar del los pilotos: las maneras en las que se integran las herramientas de automatización (IA). Los pilotos se forman y estudian para poder entender y saber manejar bien el modo automático. Todos entienden cómo funciona y qué hay detrás del “los aviones se manejan solos”. Ahí debería estar la clave: los devs (y cualquier persona que use IA como su herramienta laboral) debería primero estudiar y entender cómo funciona la herramienta para poder usarla de manera eficiente y con criterio.

Incubator

LO QUE DICE LA CIENCIA: EL COSTO DE NO PENSAR

Un estudio del MIT en 2023 mostró que, si bien las herramientas de IA aumentan la eficiencia en tareas bien definidas, también pueden reducir la capacidad de resolver problemas complejos cuando se depende demasiado de ellas.
Resolver desafíos difíciles fortalece conexiones neuronales y construye criterio a largo plazo. Delegar sistemáticamente ese esfuerzo puede afectar la capacidad cognitiva del equipo a futuro.

Referencia: MIT, 2023 - AI and Cognitive Load in Knowledge Work

DOS CAMINOS PROFESIONALES EMERGENTES

De nuestra experiencia y conversaciones con otros, vemos una diferenciación creciente en la profesión:

- Ingenieros con IA: Comprenden sistemas complejos, diseñan arquitecturas, validan resultados y guían herramientas de IA con criterio. Este perfil sube de valor en el mercado.
- Implementadores dependientes de IA: Pueden generar código funcional, pero carecen de las habilidades necesarias para escalar sistemas, enfrentar errores inesperados o tomar decisiones de alto nivel.

Incubator

¿EL PROBLEMA ES LA IA? NO. ES CÓMO SE IMPLEMENTA.

La mayoría de los desafíos que enfrentamos no tienen que ver con la tecnología, sino con las decisiones organizacionales.
Implementar IA solo para ganar velocidad, sin pensar en formación, calidad y bienestar del equipo, es una receta para futuros problemas:

- Deuda técnica oculta
- Equipos poco resilientes
- Alta rotación de talento

Un enfoque más saludable combina automatización con formación, velocidad con criterio, y tecnología con cultura.

Incubator

QUÉ ESTAMOS HACIENDO EN INCUBATOR:

Frente a este panorama, tomamos decisiones concretas para balancear velocidad y sostenibilidad:

- Espacios de práctica manual: Mantenemos momentos donde los Devs escriben código sin IA, para fortalecer habilidades fundamentales.
- Revisión técnica avanzada: Evaluamos funcionalidad, pero también arquitectura, mantenibilidad y deuda técnica.
- Formación continua: Invertimos en habilidades complementarias a la IA: diseño de sistemas, toma de decisiones técnicas, comunicación.
- Métricas balanceadas: Medimos productividad, pero también deuda técnica, motivación del equipo y capacidad de resolver problemas.

Transparencia total: Reconocemos que estamos en proceso de aprendizaje. Lo compartimos, no como receta, sino como camino en construcción.

Incubator

REFLEXIÓN FINAL: NO ES UN RETO TÉCNICO, ES CULTURAL

Después de todo lo vivido, hay una idea que se mantiene clara: el desafío de implementar IA en desarrollo no es técnico, es organizacional.
Las herramientas funcionan. El código que generan es útil. Pero construir organizaciones que escalen con IA y desarrollen talento al mismo tiempo requiere pensamiento sistémico, decisiones conscientes y una cultura que valore tanto el resultado como el proceso.

¿Cómo está evolucionando esto en tu organización?
¿Ya integraste IA para escribir código?
¿Qué desafíos encontraste al balancear productividad y desarrollo profesional?

Incubator

AUTORES

Sobre los autores:

Nicolas, CEO de Incubator

Marcos, CTO de Incubator

¿Preguntas? Contacta a ana.massacane@incubator.com.ar

Incubator