

**Escuela de Negocios**

**Tipo de documento:** Tesis de maestría



*EMBA | Executive MBA*

# Recomendaciones para consultoras de software frente a la IA y el nuevo rol del desarrollador: Caso Rootstrap

**Autoría:** Mansilla, Matías

**Año:** 2025

## ¿Cómo citar este trabajo?

Mansilla, M. (2025) "*Recomendaciones para consultoras de software frente a la IA y el nuevo rol del desarrollador: Caso Rootstrap*". [Tesis de maestría. Universidad Torcuato Di Tella].

Repositorio Digital Universidad Torcuato Di Tella.

<https://repositorio.utdt.edu/handle/20.500.13098/13893>

El presente documento se encuentra alojado en el **Repositorio Digital de la Universidad Torcuato Di Tella** bajo una licencia Creative Commons Atribución-No Comercial-Compartir Igual 4.0 Internacional  
**Dirección:** <https://repositorio.utdt.edu>



**UNIVERSIDAD  
TORCUATO DI TELLA**

**RECOMENDACIONES PARA CONSULTORAS DE  
SOFTWARE FRENTE A LA IA Y EL NUEVO ROL DEL  
DESARROLLADOR: CASO ROOTSTRAP**

**2025**

**ALUMNO: Matías Mansilla**

**TUTOR: Alejandro Restuccia**

## Agradecimientos

*A mi tutor, Alejandro, por su acompañamiento a lo largo de este proceso y por haber nutrido esta tesis con ideas valiosas, preguntas desafiantes y una mirada siempre estimulante.*

*A mi pareja, Gimena, por su apoyo constante a lo largo de estos más de dos años de estudio en otro país, siempre acompañando con paciencia y generosidad.*

*A mis compañeros y compañeras de estudio más cercanos, Lucía, Juanfra y Carolina, con quienes compartir este recorrido del MBA fue no solo enriquecedor, sino también profundamente placentero. Su compañía hizo la experiencia mucho más significativa.*

*Y a mis padres, Blanca y Raúl, por enseñarme desde siempre el valor del estudio, la formación, la constancia y el esfuerzo. Esta tesis es también reflejo de sus enseñanzas y su ejemplo.*

## Resumen ejecutivo

La industria del software ha atravesado grandes transformaciones en las últimas décadas, evolucionando a un ritmo acelerado impulsado por avances tecnológicos constantes. Según (JetBrains, The State of Developer Ecosystem, 2025), solamente desde el 2019 hasta el 2024, el número de desarrolladores de software en el mundo aumentó de 14,1 millones a 19,6 millones, equivalente a un 39%. Las empresas de consultoría de software han surgido como actores clave en este ecosistema, respondiendo a la necesidad de diversas organizaciones de llevar adelante su transformación digital y encontrar socios tecnológicos estratégicos que faciliten la adopción e implementación de nuevas tecnologías.

La reciente irrupción de la inteligencia artificial (IA) ha generado un cuestionamiento profundo sobre los paradigmas tradicionales del desarrollo de software y el rol de las empresas de consultoría. El objetivo de este trabajo es analizar esta disrupción y desarrollar recomendaciones estratégicas para las empresas de consultoría de software, utilizando como caso de estudio a Rootstrap (Rootstrap, n.d.). Para ello, se realizan entrevistas a ocho líderes tecnológicos, en conjunto con análisis de voces de expertos en entrevistas existentes, y tendencias de mercado.

Como resultado de este trabajo, se concluye que el rol del desarrollador tenderá a transformarse hacia un perfil similar al de un orquestador de sistemas de inteligencia artificial, con un enfoque creciente en aspectos de negocio y producto, además de sus responsabilidades técnicas. La demanda futura de desarrolladores y para las consultoras de software se presenta ambigua en términos cuantitativos, pero lo que sí cambia de forma clara es la expectativa cualitativa sobre los perfiles requeridos, incluido el del desarrollador. Frente a este escenario de transformación, las consultoras deberán definir estrategias concretas para adaptarse y mantenerse competitivas. En este trabajo se presenta un conjunto de recomendaciones orientadas a guiar ese proceso de adaptación.

## Palabras clave

- INTELIGENCIA ARTIFICIAL,
- CONSULTORÍA DE SOFTWARE,
- DEMANDA DE SOFTWARE,
- PRODUCTIVIDAD EN EL DESARROLLO DE SOFTWARE

## Índice / Tabla de Contenido

Agradecimientos .....	2
Resumen ejecutivo .....	3
Palabras clave .....	3
Índice / Tabla de Contenido.....	4
Tablas, Ilustraciones e Imágenes .....	7
Introducción/Preguntas.....	8
<b>CAPÍTULO 1 - EL DESARROLLO DE SOFTWARE EN LA ACTUALIDAD .....</b>	<b>9</b>
1.1 - Introducción.....	9
1.1.1 - El marco Scrum.....	9
1.2 - Principales responsabilidades del desarrollador .....	11
1.3 - Equipos de desarrollo de software y sus roles .....	13
1.3.1 - Roles .....	13
1.3.2 - Impacto del tamaño y estructura del equipo .....	16
1.4 - Medición de la productividad en el desarrollo de software .....	16
1.4.1 - Métodos tradicionales de medición .....	17
1.4.2 - Modelos recientes: SPACE y sus aplicaciones .....	18
<b>CAPÍTULO 2 - FUNDAMENTOS TEÓRICOS DE LA INTELIGENCIA ARTIFICIAL .....</b>	<b>20</b>
2.1 - Conceptos clave en Inteligencia Artificial.....	20
2.1.1 - Definición y evolución de la IA .....	20
2.1.2 - Subcampos de la IA .....	22
2.2 - El boom de la IA generativa .....	22
2.2.1 - La explosión del interés global en IA.....	22
2.2.2 - Impacto de la IA generativa en la industria .....	24
2.2.3 - Impacto de la IA generativa en la sociedad .....	24
2.3 - El auge de los agentes inteligentes y sistemas multiagente .....	25
2.3.1 - ¿Qué son los agentes y sistemas multiagente? .....	25
2.3.2 - Por qué están ganando popularidad.....	26
2.3.3 - Por qué son importantes los agentes en la actualidad y futuro .....	26
2.4 - Uso de la Inteligencia Artificial en diferentes verticales de negocio .....	27
2.4.1 - IA en la educación: Personalización del aprendizaje y asistentes inteligentes .....	27
2.4.2 - IA en el entretenimiento y la creatividad: La revolución del contenido digital.....	28
2.4.3 - IA en la industria y manufactura: Optimización de procesos y mantenimiento predictivo .....	29
<b>CAPÍTULO 3 - IA Y SU IMPACTO EN EL DESARROLLO DE SOFTWARE .....</b>	<b>31</b>
3.1 - Herramientas y Tecnologías de IA Aplicadas al Desarrollo de Software .....	31
3.1.1 - Modelos de IA usados en programación .....	31
3.1.2 - Adopción de herramientas AI según JetBrains .....	33
3.1.3 - Adopción de herramientas AI según Stack Overflow.....	34
3.1.4 - Limitaciones y desafíos de la IA en entornos de desarrollo.....	36
3.2 - IA y Productividad en el Desarrollo de Software .....	37
3.2.1 - Estudio de Microsoft (2023): IA y productividad en una muestra diversa de desarrolladores.....	37
3.2.2 - Estudio de Google: IA y su impacto en el desarrollo empresarial .....	37
3.2.3 - Evolución de las herramientas de IA y perspectivas futuras .....	38
<b>CAPÍTULO 4: SOFTWARE, DEMANDA HISTÓRICA Y ROL DE LA CONSULTORÍA TECNOLÓGICA.....</b>	<b>39</b>

4.1 - Evolución de la demanda de software desde el año 2000: hitos y tendencias recientes.....	39
4.1.1 - Expansión del software empresarial y consolidación del sector (2000-2010) .....	40
4.1.2 - Auge del desarrollo móvil y la transformación digital (2010-2020) .....	40
4.1.3 - Impacto de la pandemia y aceleración de la transformación digital (2020-2022) .....	41
4.1.4 - Ajuste del mercado y transformación con IA generativa (2023 en adelante) .....	41
4.1.5 - Industrias líderes y rezagadas en la transformación digital.....	43
4.2 - El papel de las empresas de servicios de tecnología .....	43
4.2.1 - Modelos de consultoría de software: onshore, nearshore y offshore .....	44
4.2.2 - Tamaño actual de mercado de las empresas de servicio de tecnología .....	44
4.2.3 - Principales empresas de servicios de tecnología: tamaño e ingresos anuales.....	44
4.2.4 - Desafíos actuales de las empresas de servicios de tecnología .....	45
4.3 - Caso Rootstrap.....	46
4.3.1 - Categoría de empresa .....	46
4.3.2 - Clientes .....	46
4.3.3 - Equipo y cultura .....	46
4.3.4 - Posicionamiento.....	47
<b>CAPÍTULO 5 - METODOLOGÍA DE LA INVESTIGACIÓN .....</b>	<b>49</b>
5.1 Enfoque metodológico .....	49
5.2 Técnicas de recolección de datos .....	49
5.2.1 Datos primarios .....	49
5.2.2 Datos secundarios .....	50
<b>CAPÍTULO 6 - ENTREVISTAS A LÍDERES DE TECNOLOGÍA .....</b>	<b>51</b>
6.1 - Resumen de entrevistas .....	51
6.2 - Análisis comparativo de resultados de entrevistas .....	59
6.2.1 Cambios proyectados en el rol del desarrollador según líderes de la industria tecnológica .....	60
6.2.2 Cambios proyectados en la demanda del rol del desarrollador según líderes de la industria tecnológica .....	61
<b>CAPÍTULO 7 - DATOS SECUNDARIOS .....</b>	<b>65</b>
7.1 - Proyecciones de crecimiento del mercado del software .....	65
7.2 - Proyecciones de empleo en el sector tecnológico .....	66
7.3 - Voces de expertos de la industria sobre el futuro del rol del desarrollador y su demanda .....	68
7.3.1 - Resumen de voces de expertos.....	68
7.3.2 - Análisis comparativo de las voces de expertos sobre el futuro del rol del desarrollador y su demanda .....	74
7.3.3 - Reflexión final sobre opinión de expertos .....	76
<b>CAPÍTULO 8 - PRODUCTIVIDAD, ROL Y DEMANDA FUTURA DEL DESARROLLADOR .....</b>	<b>78</b>
8.1 - Conclusiones primarias.....	78
8.1.1 - Aumento de productividad .....	78
8.1.2 - Evolución estratégica del rol del desarrollador .....	79
8.1.3 - Tendencia a la reducción del tamaño de los equipos .....	81
8.1.4 - Ambigüedad en la demanda .....	81
8.2 - Conclusiones secundarias.....	83
8.2.1 - Reconfiguración del talento: seniority, competencias y responsabilidades .....	83
8.2.2 - Democratización del software .....	84
8.2.3 - Actualización de la formación académica .....	85
<b>CAPÍTULO 9 - RECOMENDACIONES PARA CONSULTORAS DE TECNOLOGÍA.....</b>	<b>86</b>

9.1 - Resumen de recomendaciones .....	86
9.2 - Desarrollo de Recomendaciones .....	87
9.2.1 - Posicionamiento “AI-Expert” y socio estratégico .....	87
9.2.2 - Nuevos modelos de pricing para capturar el valor de la IA .....	88
9.2.3 - Demostrar experiencia real en IA .....	88
9.2.4 - Alianzas estratégicas con proveedores de modelos y plataformas de infraestructura.....	89
9.2.5 - Equipos pequeños con expertise en IA .....	90
9.2.6 - Reducir los tiempos de entrega y salida a producción .....	91
9.2.7 - Generalistas sobre especialistas .....	91
9.2.8 - Roles especialistas en IA.....	92
9.2.9 - Arquetipo del desarrollador del futuro.....	92
9.2.10 - Desarrollo de competencias clave para desarrolladores.....	93
9.2.11 - Alfabetización en IA en toda la organización .....	93
BIBLIOGRAFÍA .....	95
Anexo 1 – Guía de preguntas para entrevistas.....	100
Anexo 2 – Transcripción de entrevistas.....	102
Transcripción entrevista a Camila Moscatelli .....	102
Transcripción entrevista a Sebastian Melgar.....	110
Transcripción entrevista a Juan Pablo Mazza .....	114
Transcripción entrevista a Santiago Perez.....	119
Transcripción entrevista a José Gazzano .....	125
Transcripción entrevista a Cecilia Mariani.....	130
Transcripción entrevista a Anthony Figueroa.....	134
Transcripción entrevista a Luis Staudenmann.....	138

## Tablas, Ilustraciones e Imágenes

Ilustración 1. El marco de trabajo Scrum.....	11
Ilustración 2. Aplicación del modelo SPACE en pods ágiles. ....	19
Ilustración 3. Línea del tiempo de la IA. ....	21
Ilustración 4. Comparativa de resultados de motores de búsqueda entre ChatGPT y Google. ....	23
Ilustración 5. Adopción de IA generativa en empresas a principios de 2024. ....	24
Ilustración 6. Mapa de herramientas de asistencia al desarrollo con IA. ....	32
Ilustración 7. Percepción de beneficios de IA según encuestas a desarrolladores. ....	33
Ilustración 8. Ahorro de tiempo percibido por el uso de IA en el desarrollo. ....	34
Ilustración 9. Uso de herramientas de IA en el proceso de desarrollo (Stack Overflow, 2024).....	35
Ilustración 10. Sentimiento hacia herramientas de IA entre desarrolladores (Stack Overflow, 2024).35	
Ilustración 11. Beneficio de herramientas IA entre desarrolladores (Stack Overflow, 2024).....	36
Ilustración 12. Evolución de contrataciones y despidos en empresas de tecnología (Pragmatic Engineer, 2024). ....	40
Ilustración 13. Crecimiento de trabajos de ingeniería de software (Pragmatic Engineer, 2024). ....	42
Ilustración 14. Proyección de crecimiento del mercado global de software (2025–2030). ....	66
Ilustración 15. Proyección de empleo de desarrolladores de software en EE. UU. (2024–2034). ....	67
Ilustración 16. Proyección de empleo total en el sector tecnológico según datos de CompTIA (2024–2034). ....	67

## Introducción/Preguntas

El desarrollo de software ha sido históricamente una disciplina altamente especializada, dependiente de equipos de profesionales capacitados para crear soluciones tecnológicas a medida. Sin embargo, con la proliferación de herramientas de automatización y el avance de la inteligencia artificial (IA), el proceso de desarrollo está experimentando una transformación sin precedentes. Modelos de IA como asistentes de código y sistemas de desarrollo autónomos han comenzado a modificar la manera en que los desarrolladores trabajan, reduciendo tiempos de implementación y ampliando el acceso a la creación de software a una audiencia más amplia.

Este cambio plantea interrogantes fundamentales sobre el futuro del desarrollo de software y la sostenibilidad del modelo tradicional de consultoría tecnológica. Por un lado, la automatización promete mejorar la productividad de los desarrolladores y optimizar los costos de producción. Por otro lado, surgen dudas sobre cómo evolucionará la demanda de servicios de software y qué impacto tendrá la IA en la estructura y composición de los equipos de desarrollo. En este contexto, las empresas de consultoría de software deben redefinir su rol para mantenerse competitivas y seguir aportando valor en un entorno donde las herramientas de IA están democratizando la creación de software.

Las preguntas de investigación que guían este estudio incluyen:

- ¿Cómo está cambiando la productividad del desarrollador de software, su rol actual y la composición de los equipos de desarrollo con la adopción de IA?
- ¿Cómo impactará la inteligencia artificial en la demanda de desarrollo de software a corto y mediano plazo?
- ¿Qué estrategias deben adoptar las empresas de consultoría de software para mantenerse competitivas en un mercado donde la IA está transformando el desarrollo de software?

El tipo de Investigación e hipótesis son:

- Tipo de Investigación: el estudio tiene un enfoque descriptivo y exploratorio, con un diseño no experimental, basado en entrevistas con expertos, artículos de la industria, análisis de entrevistas existentes a expertos de la industria, e información y proyecciones de mercado. Se buscará comprender la evolución del sector y proyectar escenarios futuros, identificando patrones y estrategias que permitan a las empresas de consultoría de software navegar por el cambio tecnológico.
- Hipótesis: la hipótesis central de este trabajo sostiene que, aunque la IA transformará radicalmente el desarrollo de software y redefinirá el rol del desarrollador, la demanda por estos servicios no disminuirá. La paradoja de Jevons (Alcott, 2025) es un potencial sustento para explicar la hipótesis, donde la mayor eficiencia en el uso del recurso no reduce su demanda, sino que la incrementa. Adicionalmente, la democratización del software impulsada por la IA puede causar que más actores accedan al desarrollo de soluciones tecnológicas, expandiendo aún más la necesidad de servicios especializados en consultoría de software.

# MARCO TEÓRICO

## CAPÍTULO 1 - EL DESARROLLO DE SOFTWARE EN LA ACTUALIDAD

### 1.1 - Introducción

El desarrollo de software es una disciplina compleja que abarca desde la comprensión del problema y la planificación del producto, hasta la implementación, prueba y mantenimiento del código. De forma general, un desarrollador de software no solo escribe código, sino que participa en actividades estratégicas y colaborativas que garantizan la calidad del producto y su alineación con los objetivos de negocio.

En el contexto actual, muchas organizaciones han adoptado marcos ágiles para manejar la incertidumbre y la naturaleza iterativa del desarrollo de software. Estos métodos, a diferencia de los enfoques tradicionales de ciclo de vida en cascada (waterfall), permiten iterar constantemente sobre el producto, recibiendo retroalimentación temprana y mejorando la calidad del entregable. Como señala Hadida (Hadida, 2020), las metodologías ágiles representan una propuesta superadora frente a contextos de ambigüedad e inestabilidad, ya que favorecen la flexibilidad y la capacidad de adaptación continua del equipo de desarrollo. Entre todas las metodologías ágiles, Scrum se ha convertido en una de las metodologías más utilizadas en la industria, gracias a su enfoque flexible y su clara definición de roles y eventos. Un estudio reciente destaca que, entre todas las metodologías ágiles, Scrum sigue siendo la más adoptada por su practicidad, estructura y facilidad de implementación (Anshu, Anurag, Naveen, & Ashok, 2023).

Para ilustrar cómo se organiza el trabajo en equipos de desarrollo y cuáles son las responsabilidades de un desarrollador de software, este capítulo se basa principalmente en Scrum como referencia. Aun así, es importante destacar que existen otras metodologías (Kanban, Lean, XP, etc.) que también ofrecen ventajas y prácticas complementarias.

En los apartados siguientes, se hace una revisión del marco Scrum (Scrum Guides, 2020) y, posteriormente, se describen las principales responsabilidades de un desarrollador de software, así como el modo en que se conforman e interactúan los equipos de desarrollo en la actualidad.

#### 1.1.1 - El marco Scrum

Scrum (Scrum Guides, 2020) es un marco de trabajo ágil diseñado para desarrollar y mantener productos complejos de forma iterativa e incremental. Su objetivo principal es generar valor de manera frecuente y sostenida, de modo que los equipos puedan adaptar rápidamente su planificación a partir de la retroalimentación de los usuarios o clientes. Se apoya en tres pilares: transparencia, inspección y adaptación, promoviendo que todo el proceso de desarrollo sea visible para los interesados, que se revisen continuamente los avances y que se realicen ajustes conforme cambien las prioridades o se detecten oportunidades de mejora.

##### *1.1.1.1 - Definiciones y elementos clave*

**Sprint:** El Sprint es el ciclo de trabajo que marca el ritmo de Scrum. Generalmente dura entre una y cuatro semanas, siendo dos semanas la longitud más común en la industria. Durante este período, el equipo concentra sus esfuerzos en un conjunto específico de tareas orientadas a la construcción o evolución del producto. Al finalizar el Sprint, se presenta y evalúa el trabajo completado, se recopila retroalimentación y se decide cómo proceder en la siguiente iteración. Este enfoque iterativo reduce

los riesgos y facilita la introducción de mejoras continuas, ya que cada Sprint entrega un avance tangible.

**Product Backlog:** El Product Backlog es la lista priorizada de todo lo que el producto necesita para satisfacer los requerimientos del negocio o del usuario final. A lo largo del tiempo, evoluciona de acuerdo con los cambios en las necesidades y la retroalimentación recibida tras cada entrega. Se espera que el equipo extraiga de este conjunto de requisitos los elementos que se abordarán en cada Sprint, manteniendo así una visión estratégica del producto y un orden claro de ejecución según el valor aportado.

**Sprint Backlog:** Una vez definido el objetivo del Sprint, se seleccionan del Product Backlog aquellos elementos que se llevarán a cabo durante esa iteración. El Sprint Backlog actúa como un plan de trabajo a corto plazo, donde cada tarea se detalla lo suficiente para permitir que el equipo avance de manera autoorganizada. A medida que se ejecuta el Sprint, el equipo puede ajustar las tareas o la priorización interna para cumplir con los objetivos, siempre asegurando la visibilidad de estos cambios.

**Incremento de producto:** El resultado de cada Sprint es un “Incremento” que cumple con los criterios de calidad establecidos, comúnmente referidos como “Definición de Terminado” (Scrum Guides, 2020). Esto garantiza que el trabajo realizado esté listo para su potencial liberación y uso, o al menos, para ser evaluado por los interesados. En ocasiones, en etapas iniciales de desarrollo o cuando se experimenta con nuevas ideas, se habla de un MVP (Producto Mínimo Viable), que consiste en la versión más básica del producto que permite validar hipótesis de negocio y obtener retroalimentación real sin invertir esfuerzos excesivos en funcionalidades no críticas.

#### *1.1.1.2 - Principales ceremonias o reuniones*

Scrum organiza el trabajo en torno a eventos que dan estructura y permiten la continua inspección y adaptación. Al inicio de cada Sprint se realiza una planificación para definir qué se ejecutará en ese periodo y cómo se abordarán las tareas “Sprint Planning”. De manera diaria “Daily Scrum”, una reunión corta ayuda al equipo a sincronizar progreso, identificar bloqueos y reajustar tareas. Al final del Sprint, durante la revisión, se muestra el Incremento a los interesados “Sprint Review”, recogiendo comentarios valiosos. Finalmente, el equipo reflexiona en la retrospectiva para mejorar los procesos y la colaboración “Sprint Retrospective”. En la Ilustración 1. El marco de trabajo Scrum se observa una representación del marco de trabajo Scrum y sus principales eventos.

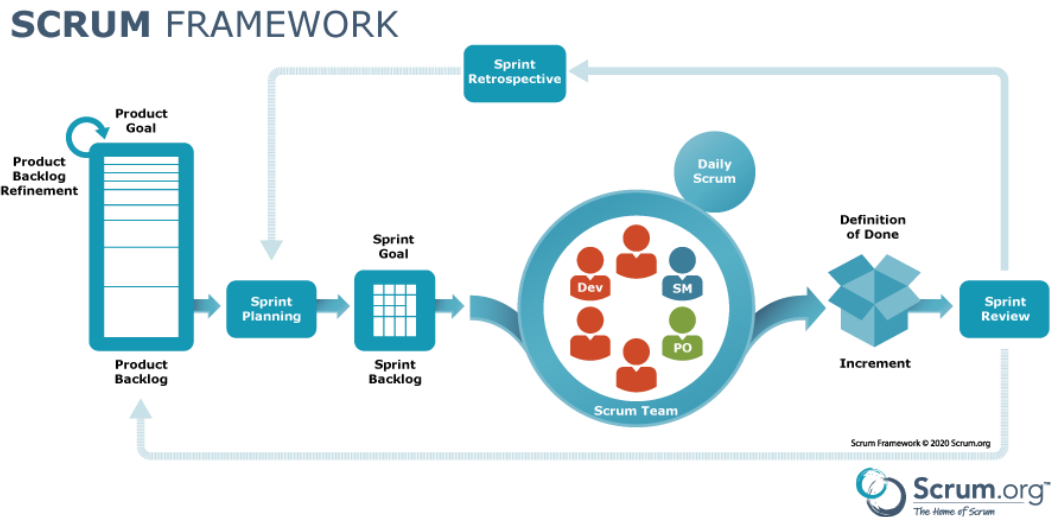


Ilustración 1. El marco de trabajo Scrum

**Nota.** Adaptado de The Scrum Framework Poster (Scrum Guides, 2020) <https://www.scrum.org/resources/scrum-framework-poster>

## 1.2 - Principales responsabilidades del desarrollador

En este apartado se describen las funciones esenciales que un desarrollador de software desempeña en un contexto de desarrollo ágil. Aunque los marcos de trabajo como Scrum (Scrum Guides, 2020), analizado en el apartado 1.1.1, orientan la forma en que se organizan las iteraciones y las ceremonias, las responsabilidades expuestas a continuación son independientes de la metodología específica y se mantienen vigentes en la mayoría de los proyectos de software.

### Comprensión del negocio y análisis de requerimientos

Antes de desarrollar cualquier solución tecnológica, es fundamental que el desarrollador comprenda el contexto del negocio y las necesidades del usuario final. Esta fase involucra la interpretación de requisitos funcionales y no funcionales, la colaboración con equipos de producto, diseñadores y clientes, y la transformación de estas necesidades en una especificación técnica clara.

Dentro de entornos ágiles como Scrum, los desarrolladores participan activamente en la planificación del Sprint, principalmente en ceremonias de planificación (Scrum Guides, 2020) como se mencionó antes. Esto les permite definir el alcance del trabajo y asegurarse de que las funcionalidades a implementar resuelven efectivamente los problemas del usuario.

### Desarrollo de código y pruebas

La programación es la responsabilidad nuclear del desarrollador. Su labor incluye escribir código, mantenerlo y mejorarlo, apoyándose en buenas prácticas como la reutilización, la modularidad y el uso de patrones de diseño. Paralelamente, las pruebas (unitarias, de integración o automatizadas) forman parte integral de este proceso, permitiendo descubrir y corregir errores con anticipación.

En muchos equipos ágiles, la calidad es responsabilidad de todos sus integrantes, por lo que el desarrollador no solo implementa nuevas funcionalidades, sino que también verifica que se cumplan los criterios de calidad establecidos. Esto puede incluir la adopción de una “Definición de Terminado”

clara, tal como lo establece Scrum (Scrum Guides, 2020), la cual determina el estado de completitud de una tarea o funcionalidad antes de su entrega.

#### Documentación y revisión del trabajo de otros

La documentación es esencial para garantizar la mantenibilidad del software y la continuidad de los proyectos. Aunque a menudo se pasa por alto, resulta clave para que otros desarrolladores o el mismo equipo, en el futuro, entiendan las decisiones de diseño y la arquitectura del sistema. Esta documentación puede incluir la descripción de APIs, guías de usuario y registros de cambios (changelogs), así como directrices sobre la configuración del entorno de desarrollo.

#### Revisión de código

La revisión de código es una práctica extendida que consiste en analizar el trabajo realizado por los compañeros antes de que sea incorporado a la base principal del proyecto. Gracias a esta actividad, el equipo garantiza que se cumplan los estándares de calidad, las buenas prácticas de programación y la coherencia general del proyecto. Adicionalmente, la revisión de código promueve el aprendizaje colectivo y contribuye a detectar posibles errores o vulnerabilidades de forma temprana.

#### Presentación del trabajo a stakeholders

El trabajo del desarrollador no se limita a la implementación técnica. En muchas ocasiones, debe comunicar avances y decisiones clave a stakeholders no técnicos, como clientes, gerentes de producto y ejecutivos.

Esta responsabilidad incluye:

- Explicar el estado de desarrollo y las funcionalidades implementadas.
- Justificar decisiones técnicas y discutir posibles ajustes de alcance.
- Recibir retroalimentación y adaptar el desarrollo a nuevas necesidades del negocio.

En marcos ágiles como Scrum, los desarrolladores presentan el producto en la Revisión del Sprint (Scrum Guides, 2020) donde muestran lo que se ha construido y reciben retroalimentación de los stakeholders. Este proceso no solo permite validar el trabajo realizado, sino que también facilita la alineación del producto con las expectativas del cliente.

#### Comunicación y colaboración

El desarrollo de software es un esfuerzo de equipo que requiere una comunicación fluida y continua. Los desarrolladores colaboran con profesionales de diseño, QA y gestión de producto para encauzar el proyecto hacia las metas pactadas. Aunque algunas metodologías ágiles establecen reuniones diarias o retrospectivas, la esencia de estas prácticas es fomentar la coordinación del equipo, la identificación de bloques y la búsqueda de soluciones conjuntas.

Los canales de comunicación asincrónica (como herramientas de mensajería o tableros de tiques) también cumplen un rol fundamental, sobre todo en equipos distribuidos o con diferentes zonas horarias. En conjunto, estas dinámicas contribuyen a la eficiencia y transparencia del proceso de desarrollo.

## 1.3 - Equipos de desarrollo de software y sus roles

Los equipos de desarrollo de software están compuestos por múltiples roles que trabajan en conjunto para construir productos tecnológicos eficientes y escalables. Si bien las estructuras de los equipos pueden variar en función del marco metodológico adoptado por cada organización, existen ciertos roles fundamentales que se encuentran presentes en la mayoría de los entornos de desarrollo.

En Scrum, los equipos son autoorganizados, autosuficientes y multidisciplinarios (Scrum Guides, 2020), lo que significa que los diferentes miembros del equipo trabajan en estrecha comunicación para lograr los objetivos del Sprint. Sin embargo, aunque esta estructura es una de las más ampliamente utilizadas en la industria del software, cada organización puede adaptar estos roles en función de su contexto y necesidades específicas.

A continuación, se describen los principales roles que suelen conformar un equipo de desarrollo de software, incluyendo tanto los definidos formalmente por el marco de trabajo Scrum como otros roles ampliamente adoptados en la industria. Si bien algunos de estos no están contemplados en la guía oficial de Scrum, representan funciones especializadas que muchas organizaciones integran en sus equipos según sus necesidades, estructuras y niveles de madurez técnica. En todos los casos, se analiza además cómo cada uno de estos roles se relaciona con el rol del desarrollador.

### 1.3.1 - Roles

#### Desarrollador

El desarrollador es el responsable de construir el producto digital, escribiendo, probando y optimizando código para garantizar el correcto funcionamiento del software. Su labor no se limita únicamente a la programación, sino que también abarca la revisión de código, la integración de sistemas, la documentación técnica y la colaboración con otros miembros del equipo para alinear los objetivos técnicos con las necesidades del negocio.

En metodologías ágiles como Scrum, los desarrolladores trabajan de manera autoorganizada dentro del equipo, participando en la planificación del trabajo y en la toma de decisiones sobre la implementación de funcionalidades. Además, son responsables de garantizar que cada incremento de software cumpla con los criterios de calidad definidos en la Definición de Terminado (Scrum Guides, 2020).

#### Quality Assurance (QA)

El rol de QA es clave para garantizar la calidad del producto desarrollado, mediante la planificación, ejecución y automatización de pruebas. Aunque no está formalmente definido en el marco de trabajo Scrum, las tareas de aseguramiento de calidad suelen ser asumidas como una especialidad dentro del equipo de Developers. Scrum establece un equipo de desarrollo multifuncional responsable de entregar valor en cada Sprint, por lo tanto, quienes se especializan en QA forman parte de ese equipo, contribuyendo con su expertise en testing manual o automatizado, integración continua y mejora continua del producto.

Según Florea (Radu, Vladislav, & Dag I.K., 2023), en contextos ágiles los profesionales de QA no solo ejecutan pruebas, sino que asumen una variedad de sub-roles que aportan a la estrategia de calidad del equipo: desde diseñar escenarios de prueba hasta monitorear el comportamiento del sistema en producción. Su participación desde etapas tempranas permite detectar defectos de forma anticipada y acelerar la entrega sin comprometer la estabilidad del software. A diferencia de enfoques

tradicionales donde las pruebas se realizaban tras la implementación, en Scrum la validación de calidad es un proceso continuo, integrado y compartido por todo el equipo.

#### Product Designer

El diseñador de producto (Product Designer) es responsable de la experiencia y usabilidad del software, asegurándose de que la interfaz sea intuitiva y eficiente para el usuario final. Su trabajo implica la creación de prototipos, la validación de diseños con usuarios y la colaboración con los desarrolladores para garantizar que la implementación respete las directrices de diseño.

Aunque el marco Scrum no define un rol específico para diseño de experiencia de usuario, en la práctica, los diseñadores de producto suelen integrarse dentro del equipo de desarrollo y colaborar estrechamente con los Product Owners y Developers. Según Bruun (Anders, Marta K., Lone, Per A., & Jesper S., 2018), estos profesionales asumen una amplia gama de responsabilidades que van desde la investigación con usuarios hasta la participación en pruebas de aceptación, y su inclusión en equipos ágiles ha demostrado mejorar la calidad del producto sin comprometer la velocidad de entrega.

En Scrum, los diseñadores trabajan junto con los desarrolladores en la planificación de sprints, asegurando que las decisiones de diseño se integren adecuadamente en el flujo de desarrollo. Esta interacción es fundamental para evitar problemas de implementación y garantizar que los requerimientos de usabilidad se cumplan sin afectar la viabilidad técnica del proyecto.

#### Product Owner

El Product Owner es el responsable de maximizar el valor del producto desarrollado por el equipo. Su principal función es la gestión del backlog del producto: definir, priorizar y refinar las funcionalidades a desarrollar, alineándolas con las necesidades del negocio y de los usuarios. Participa activamente en las ceremonias ágiles, colabora con el equipo técnico y asegura que el desarrollo esté enfocado en generar el mayor impacto posible.

En algunos equipos, especialmente aquellos que trabajan en productos complejos, con múltiples stakeholders o en etapas tempranas de descubrimiento, este rol es ocupado por un Product Manager. A diferencia del Product Owner, que se centra principalmente en la ejecución del backlog, el Product Manager tiene una mirada más estratégica, orientada a la visión del producto, el posicionamiento en el mercado, la validación de hipótesis y la gestión del ciclo de vida completo del producto. En contextos ágiles, ambos roles pueden convivir o integrarse, dependiendo del tamaño del equipo y la madurez del producto.

El Product Owner trabaja en estrecha colaboración con los desarrolladores, brindándoles claridad sobre qué se debe construir y por qué. Esta interacción constante permite al equipo tomar decisiones más informadas, priorizar correctamente y adaptar el desarrollo a cambios en el contexto o en las necesidades del negocio. El éxito del rol depende, en gran medida, de su capacidad para comunicar una visión clara y traducirla en objetivos concretos para el equipo técnico. (Scrum Guides, 2020).

#### Scrum Master

El Scrum Master es un facilitador dentro del equipo de desarrollo, asegurando que las prácticas y principios de Scrum se apliquen correctamente. Su rol no es el de un gerente tradicional, sino el de un mentor que ayuda al equipo a optimizar su flujo de trabajo y eliminar impedimentos que puedan afectar su rendimiento.

Desde la perspectiva de los desarrolladores, el Scrum Master es un aliado en la gestión del equipo, asegurando que puedan enfocarse en su trabajo sin interrupciones innecesarias. También fomenta la

mejora continua a través de eventos como las retrospectivas, donde el equipo analiza lo que funcionó bien y qué aspectos pueden mejorarse en futuros sprints (Scrum Guides, 2020).

#### Team Lead

El rol de Team Lead, aunque no es parte de la estructura formal de Scrum, es común en equipos de desarrollo más grandes, con desafíos técnicos complejos y una alta necesidad de coordinación con otros roles como diseño, producto y áreas de negocio. El Team Lead suele ser un desarrollador con experiencia técnica y habilidades de liderazgo, que guía al equipo en términos de buenas prácticas de desarrollo, revisión de código, toma de decisiones técnicas y resolución de problemas. También actúa como un nexo entre el equipo y otros interlocutores clave, ayudando a alinear el trabajo técnico con los objetivos generales del producto.

Según el marco DSDM del Agile Business Consortium (Agile Business Consortium), que suele utilizarse como una extensión o complemento de Scrum en contextos organizacionales más complejos, el rol de Team Leader tiene como propósito facilitar la colaboración dentro del equipo, asegurar la calidad de la entrega técnica y promover una comunicación efectiva tanto interna como externa. En línea con esto, el Team Lead colabora de manera cercana con los desarrolladores, ofreciendo soporte técnico, mentoring y ayudando a remover obstáculos que puedan afectar la productividad del equipo. Su rol es clave para fomentar una cultura de mejora continua, calidad en el código y crecimiento profesional dentro del equipo, manteniendo un equilibrio entre la autonomía de los desarrolladores y la alineación técnica general.

#### Project Manager

El rol de Project Manager no está formalmente definido en Scrum, donde la gestión del trabajo recae colectivamente en el equipo. Sin embargo, en muchas organizaciones, especialmente en consultoras tecnológicas o proyectos de cierta escala, el Project Manager continúa desempeñando un papel clave en la planificación, el seguimiento y la coordinación del proyecto. Suele asumir responsabilidades como la definición de cronogramas, la gestión de presupuestos y riesgos, y la comunicación con stakeholders externos. Con frecuencia actúa como puente entre el equipo técnico y otras áreas de la organización, facilitando la ejecución del proyecto de forma alineada con los objetivos del negocio.

Según una revisión sistemática de Gandomani (Tahir, Zohreh, Halimah, & Hamid, 2020), aunque Agile no contempla explícitamente esta figura, en la práctica muchas organizaciones la integran como un complemento útil para resolver desafíos que exceden el alcance del equipo Scrum. En estos contextos, el Project Manager tiende a adoptar un rol de facilitador, ayudando a remover impedimentos organizacionales, coordinando a los distintos actores involucrados y promoviendo una visión compartida del proyecto. De esta manera, colabora con los desarrolladores y otros roles ágiles generando las condiciones necesarias para que el equipo pueda enfocarse en entregar valor de forma continua.

#### DevOps

El rol de DevOps se enfoca en integrar el desarrollo y las operaciones, facilitando la entrega continua de software mediante la automatización de procesos, la gestión de infraestructura y la supervisión del rendimiento del sistema. Aunque DevOps no es un rol formalmente definido dentro del marco Scrum, esta función suele ser desempeñada por miembros del equipo de Developers con especialización en operaciones y despliegue. En equipos Scrum, se espera que los Developers sean multifuncionales, por lo que las prácticas de DevOps forman parte integral del trabajo colaborativo necesario para entregar incrementos funcionales de forma continua y confiable, sin embargo, en varios casos es un especialista en DevOps quien asume dichas responsabilidades.

Según Lwakatare (Lucy Ellen, et al., 2019), la implementación de prácticas DevOps en conjunto con metodologías ágiles permite acelerar los ciclos de entrega, reducir errores en despliegue y mejorar la colaboración entre equipos de desarrollo y operaciones. En este contexto, los ingenieros DevOps suelen integrarse al equipo de desarrollo o colaborar estrechamente con él, contribuyendo al proceso de integración continua y asegurando que cada incremento de producto pueda ser liberado de manera confiable y sostenible.

### 1.3.2 - Impacto del tamaño y estructura del equipo

El tamaño y la estructura de un equipo de desarrollo de software influyen directamente en su productividad y capacidad para entregar software de calidad. Según Scrum (Scrum Guides, 2020), el tamaño óptimo de un equipo de desarrollo varía entre 3 y 9 miembros. Equipos más pequeños tienden a ser más ágiles y eficientes, facilitando la toma de decisiones y reduciendo la complejidad en la comunicación. En contraste, equipos más grandes pueden enfrentar problemas de coordinación y necesitar mecanismos adicionales para mantener la alineación y evitar bloqueos en el flujo de trabajo.

La forma en que se estructuran los equipos también tiene un impacto en la productividad. Existen diferentes modelos, entre ellos:

- Equipos multifuncionales: Combinan distintos especialistas (desarrolladores, QA, DevOps, Product Designers) dentro de un mismo equipo, lo que facilita la autonomía y la entrega de software funcional en cada iteración.
- Equipos especializados: Separan a los desarrolladores en áreas específicas como backend, frontend o infraestructura, permitiendo una mayor profundidad técnica en cada área, pero requiriendo más coordinación interdepartamental.
- Equipos distribuidos: Se componen de miembros en distintas ubicaciones geográficas, lo que puede aumentar la diversidad y disponibilidad del talento, pero también requiere estrategias efectivas de comunicación asincrónica.

Elegir la estructura adecuada depende del tipo de producto que se esté desarrollando, el tamaño de la organización y la madurez del equipo en metodologías ágiles. La combinación de una estructura bien definida y una comunicación efectiva es clave para mantener un flujo de trabajo eficiente y entregar software de alta calidad.

## 1.4 - Medición de la productividad en el desarrollo de software

La productividad en el desarrollo de software es un concepto amplio que ha evolucionado con el tiempo. Su medición no puede limitarse a métricas simples como la cantidad de código escrito o las tareas completadas, sino que debe considerar la calidad del producto, la eficiencia del equipo y el impacto en el negocio. Diferentes enfoques han surgido para evaluar la productividad en este campo, desde métricas tradicionales hasta modelos más holísticos que buscan capturar la complejidad del trabajo del desarrollador.

En esta sección se presentan los principales mecanismos utilizados para medir la productividad en el desarrollo de software, comenzando por los enfoques tradicionales y culminando con modelos más recientes como SPACE, que integran dimensiones cualitativas y cuantitativas (Forsgren, Storey, & Humble, 2021).

### 1.4.1 - Métodos tradicionales de medición

A lo largo de los años, se han utilizado diversas métricas para evaluar la productividad en el desarrollo de software. Un estudio sistemático de métricas aplicadas a equipos (Hernández, 2019) identificó diferentes enfoques que pueden agruparse según el aspecto que buscan medir:

#### Métricas de entrega y frecuencia de desarrollo

Estas métricas evalúan qué tan rápido y con qué frecuencia un equipo es capaz de entregar software funcional.

- Tiempo de entrega: Mide el tiempo transcurrido desde que una tarea es asignada hasta que se entrega en producción.
- Tiempo de ciclo: Evalúa la duración total de una tarea desde su inicio hasta su finalización.
- Frecuencia de despliegues: Indica cuántas veces se liberan cambios en producción en un período determinado.

Estas métricas son especialmente relevantes en entornos ágiles y DevOps, donde la velocidad de entrega es un factor crítico.

#### Métricas de esfuerzo y eficiencia

Miden la cantidad de trabajo realizado y el esfuerzo invertido en el desarrollo de software.

- Velocidad del equipo (Velocity en Scrum): Representa la cantidad de trabajo completado en un sprint, generalmente medida en story points.
- Throughput: Mide la cantidad de tareas finalizadas en un período determinado, sin considerar su tamaño.
- Esfuerzo por iteración: Evalúa cuántas horas de trabajo se dedicaron a una tarea o sprint.

Si bien estas métricas permiten hacer estimaciones y planificaciones, pueden ser afectadas por la subjetividad en la asignación de story points o la variabilidad en la complejidad de las tareas.

#### Métricas de valor añadido al producto

Se centran en medir el impacto del software desarrollado en el negocio y el usuario final.

- Puntos de función entregados: Evalúan la cantidad de funcionalidades desarrolladas desde la perspectiva del usuario.
- Valor de negocio generado: Relaciona las características implementadas con su impacto en los objetivos de la organización.
- Satisfacción del cliente: Se mide a través de encuestas o feedback directo de los usuarios finales.

Estas métricas son importantes para evaluar la efectividad del desarrollo, pero requieren mecanismos adecuados para capturar correctamente la percepción del usuario y el negocio.

#### Métricas de calidad del software

Evalúan qué tan confiable y mantenible es el código entregado.

- Número de defectos: Indica la cantidad de errores reportados en el software en un período determinado.

- Tasa de defectos: Calcula la relación entre defectos encontrados y código desarrollado.
- Tiempo de corrección de errores: Mide cuánto tiempo tarda el equipo en resolver defectos una vez reportados.

Un código con alta calidad tiende a reducir la deuda técnica y mejorar la sostenibilidad del producto a largo plazo.

#### Métricas de colaboración y comunicación

Evalúan el impacto del trabajo en equipo y la cultura organizacional en el desarrollo de software.

- Participación en reuniones: Mide la cantidad y calidad de la interacción del equipo en eventos de planificación y retrospectiva.
- Satisfacción del equipo: Se obtiene mediante encuestas internas para evaluar el bienestar de los desarrolladores.
- Tasa de rotación del equipo: Analiza la cantidad de miembros que abandonan el equipo en un período determinado.

Estos indicadores pueden influir directamente en la productividad, ya que equipos con problemas de comunicación o alta rotación suelen experimentar más bloqueos en el desarrollo.

#### Métricas de planificación y estimación

Evalúan la precisión con la que un equipo puede planificar y cumplir sus objetivos.

- Precisión en estimaciones: Compara el esfuerzo estimado con el esfuerzo real invertido.
- Cumplimiento de plazos: Analiza cuántas tareas fueron completadas dentro del tiempo planeado.
- Índice de retrabajo: Mide la cantidad de tareas que requieren ajustes después de ser consideradas terminadas.

Estas métricas son útiles para optimizar procesos y mejorar la capacidad predictiva del equipo.

#### Métricas de código y automatización

Relacionadas con la calidad del código fuente y el nivel de automatización en el desarrollo.

- Complejidad del código: Evalúa la dificultad inherente del código fuente, considerando factores como la cantidad de decisiones lógicas, bucles y caminos independientes que pueden recorrer los flujos de ejecución.
- Cobertura de pruebas: Indica el porcentaje del código que ha sido evaluado mediante pruebas automatizadas.
- Tasa de automatización de pruebas: Mide el nivel de automatización en la estrategia de testing del equipo.

Equipos con una alta cobertura de pruebas y un código bien estructurado tienden a experimentar menos problemas en producción y mayor facilidad en el mantenimiento del software.

#### 1.4.2 - Modelos recientes: SPACE y sus aplicaciones

Los métodos tradicionales de medición, aunque útiles, han demostrado ser insuficientes para capturar la complejidad de la productividad en el desarrollo de software. Por ello, modelos recientes como

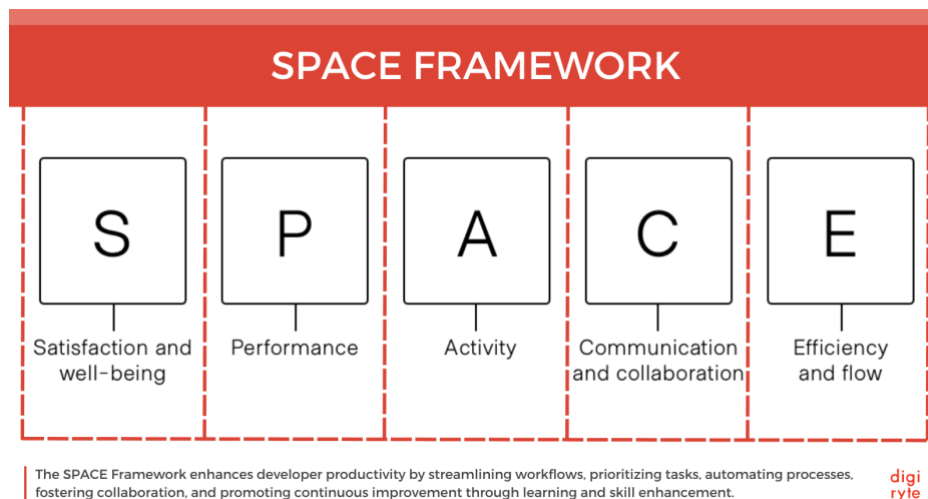
SPACE (Forsgren, Storey, & Humble, 2021), propuesto por investigadores de Microsoft Research, han surgido como alternativas más completas.

El marco SPACE propone que la productividad no puede medirse únicamente por la cantidad de trabajo entregado, sino que debe evaluarse en función de cinco dimensiones clave:

1. Satisfacción y bienestar: Cómo los desarrolladores perciben su propio rendimiento y entorno de trabajo.
2. Rendimiento: Impacto del software en el usuario y el negocio.
3. Actividad: Cantidad de tareas completadas o entregables generados.
4. Comunicación y colaboración: Calidad de las interacciones dentro del equipo.
5. Eficiencia y flujo: Capacidad de los equipos para completar trabajo sin interrupciones.

Este enfoque permite analizar la productividad desde una perspectiva más integral, considerando no solo la cantidad de trabajo realizado, sino también el impacto de factores humanos y organizacionales.

En el contexto de la inteligencia artificial, SPACE cobra aún más relevancia, ya que permite evaluar no solo cuánto código genera la IA, sino también cómo impacta en la calidad del software y en la dinámica del equipo de desarrollo. A medida que las herramientas de IA se integren en los procesos de desarrollo, será crucial adoptar modelos de medición que reflejen el impacto real de estas tecnologías en la productividad y el bienestar del equipo. La Ilustración 2. Aplicación del modelo SPACE en pods ágiles ilustra cómo se representa visualmente el modelo SPACE y sus cinco dimensiones clave en el contexto de pods ágiles.



*Ilustración 2. Aplicación del modelo SPACE en pods ágiles*

**Nota.** Adaptado de *Learn how Digiryte's agile pods leverage the SPACE framework*, por Digiryte, s.f., <https://www.digiryte.com/article/learn-how-digiryte-s-agile-pods-leverage-the-space-framework/>

# CAPÍTULO 2 - FUNDAMENTOS TEÓRICOS DE LA INTELIGENCIA ARTIFICIAL

## 2.1 - Conceptos clave en Inteligencia Artificial

La inteligencia artificial (IA) se ha consolidado como una disciplina fundamental en la industria tecnológica, impulsando la creación de soluciones cada vez más sofisticadas y redefiniendo la manera en que el software es concebido, desarrollado y mantenido. Con la salida de ChatGPT (OpenAI, 2022) y la popularización de modelos basados en transformers (Vaswani, et al., 2023), se produjo un crecimiento exponencial en la adopción de herramientas de IA en distintos ámbitos, desde el desarrollo de software hasta la generación de contenido y la automatización de procesos.

Este apartado revisa los conceptos esenciales de la IA, sus principales subcampos y la diferencia entre las IA generativas y otras vertientes de la disciplina, con el propósito de establecer una base teórica sólida antes de profundizar en su impacto en el desarrollo de software.

### 2.1.1 - Definición y evolución de la IA

La inteligencia artificial (IA) puede definirse como la rama de la computación que busca crear sistemas capaces de realizar tareas que, en principio, requieren de habilidades humanas para la resolución de problemas, el razonamiento, la toma de decisiones y el aprendizaje (Russell & Norvig, 2020). Esta definición abarca una amplia gama de técnicas, desde la lógica simbólica inicial hasta el aprendizaje profundo de las últimas décadas.

La historia de la IA está marcada por fases de gran entusiasmo y otros periodos de estancamiento (conocidos como “inviernos de la IA”), así como por avances tecnológicos que abrieron nuevas posibilidades de aplicación. Tal como se describe en Bit Brain (Bit2Brain, 2023), su evolución podría sintetizarse en los siguientes hitos:

- Década de 1950: Los primeros pasos: Tras la publicación del artículo de Alan Turing “Computing Machinery and Intelligence” en 1950, la comunidad científica empezó a preguntarse si las máquinas podían *realmente* pensar. El famoso Test de Turing supuso un punto de partida para la discusión sobre la posible “inteligencia” de las máquinas.
- Conferencia de Dartmouth (1956): Considerada por muchos como el “acta de nacimiento” de la IA. Durante este encuentro, John McCarthy, Marvin Minsky y otros pioneros propusieron investigar cómo lograr que las máquinas fueran capaces de *aprender* y *razonar* tal cual los seres humanos. El optimismo inicial llevó a grandes promesas y expectativas sobre lo que se podría alcanzar en corto tiempo.
- Auge simbólico y primeros sistemas expertos (décadas de 1960-1970): Los investigadores desarrollaron métodos basados en reglas y representaciones lógicas para modelar el conocimiento. A pesar de sus éxitos puntuales en dominios restringidos, estos sistemas resultaban rígidos ante situaciones complejas o datos inciertos. Durante estos años surgieron sistemas expertos capaces de tomar decisiones en contextos especializados, pero con grandes limitaciones prácticas.
- Los “inviernos de la IA” (décadas de 1970-1980 y finales de 1980-1990): Debido al exceso de promesas incumplidas, la falta de resultados comerciales tangibles y la dificultad de escalar los sistemas basados en reglas, la financiación y el interés general por la IA disminuyeron drásticamente en distintos periodos. En estos “inviernos”, la investigación continuó, pero con menor apoyo económico e institucional.
- Resurgimiento con el Machine Learning y el auge del Big Data (décadas de 1990-2000): La disponibilidad creciente de datos masivos (Big Data) y el aumento de la capacidad

computacional reactivaron la investigación en algoritmos estadísticos. El concepto de aprendizaje automático (Machine Learning) cobró fuerza, desplazando el paradigma rígido de las reglas a modelos capaces de generalizar a partir de ejemplos.

- Advenimiento del Deep Learning (a partir de la década de 2010): Investigaciones en redes neuronales profundas y la llegada de hardware especializado (como GPUs) permitieron aumentar la precisión en tareas de visión por computadora, procesamiento del lenguaje natural y muchas otras. Empresas tecnológicas comenzaron a lanzar asistentes virtuales y sistemas avanzados de análisis, inaugurando la actual era de la IA en nuestras vidas cotidianas.
- IA generativa y consolidación (2020 en adelante): Modelos como GPT (OpenAI, 2022), DALL-E (OpenAI, 2021) o Stable Diffusion (Stability AI, 2022) han llevado la IA a un público amplio, demostrando habilidades sorprendentes para generar texto, imágenes o incluso música. Esto no solo ha despertado un renovado interés de inversores y usuarios, sino que ha planteado importantes reflexiones éticas y debates regulatorios en torno al uso masivo de estas tecnologías.

La evolución descrita demuestra que la IA no se ha desarrollado de forma lineal, sino a través de ciclos de gran impulso y periodos de retroceso. En la actualidad, asistimos a una fase de consolidación donde la IA generativa y las aplicaciones de aprendizaje profundo se han convertido en piezas clave para múltiples industrias, ofreciendo nuevas soluciones y, al mismo tiempo, nuevos retos. La Ilustración 3. Línea del tiempo de la IA resume los diferentes hitos antes mencionados, posicionando cada uno, en una línea temporal desde 1950 hasta 2024.



Ilustración 3. Línea del tiempo de la IA

**Nota.** Reproducido de *Historia de la IA*, por Bit Brain, 2023 (Bit Brain, 2023)

### 2.1.2 - Subcampos de la IA

La inteligencia artificial es un campo amplio que abarca múltiples subdisciplinas, cada una especializada en resolver distintos tipos de problemas. A continuación se presentan los principales subcampos, según Russell y Norvig (Russell & Norvig, 2020):

1. Aprendizaje Automático (Machine Learning, ML): Se centra en algoritmos que permiten a las máquinas aprender patrones a partir de datos y mejorar su desempeño con la experiencia. Se utiliza en tareas como clasificación, regresión y toma de decisiones bajo incertidumbre.
2. Aprendizaje Profundo (Deep Learning, DL): Es una subárea del aprendizaje automático que emplea redes neuronales artificiales con múltiples capas para modelar relaciones complejas en grandes volúmenes de datos. Tiene aplicaciones destacadas en visión por computadora, procesamiento de lenguaje natural y juegos.
3. Procesamiento del Lenguaje Natural (Natural Language Processing, NLP): Busca la comprensión, generación y traducción del lenguaje humano por parte de las máquinas. Incluye tareas como análisis sintáctico, reconocimiento de entidades, generación de texto y diálogo conversacional.
4. Visión por Computadora (Computer Vision): Permite a los sistemas interpretar imágenes y secuencias de video, extrayendo información útil del entorno visual. Es clave para tareas como el reconocimiento facial, la conducción autónoma y el análisis médico.
5. Aprendizaje por Refuerzo (Reinforcement Learning): Se basa en agentes que aprenden a actuar en un entorno mediante la maximización de recompensas acumuladas. Ha sido fundamental en desarrollos de IA en juegos, robótica y optimización.
6. Representación del conocimiento y razonamiento automatizado: Estudia cómo representar formalmente el conocimiento del mundo y cómo razonar a partir de él para resolver problemas complejos. Este subcampo es la base de los sistemas expertos y los sistemas de diagnóstico automático.
7. Robótica: Integra IA con sensores, actuadores y control mecánico para desarrollar agentes físicos autónomos capaces de percibir su entorno, planificar y actuar. Es esencial en la manufactura automatizada, exploración espacial y asistencia personal.
8. Sistemas Expertos: Son aplicaciones que emulan el razonamiento de un especialista humano en un dominio específico, utilizando reglas lógicas y bases de conocimiento estructurado.
9. Optimización y Búsqueda Heurística: Se refiere al uso de métodos como algoritmos genéticos, búsqueda A\*, y otras técnicas para encontrar soluciones óptimas o aproximadas en espacios de búsqueda complejos.

## 2.2 - El boom de la IA generativa

Si bien la inteligencia artificial ha evolucionado a lo largo de décadas, el verdadero punto de inflexión ocurrió con la irrupción de la IA generativa, particularmente con el lanzamiento de ChatGPT (OpenAI, 2022) en noviembre de 2022. Basado en modelos de lenguaje de gran escala (LLMs) como GPT-3.5 y GPT-4, el éxito de esta tecnología marcó el inicio de una nueva era en la adopción masiva de IA, transformando radicalmente la forma en que interactuamos con la tecnología y acelerando la integración de IA en múltiples sectores.

### 2.2.1 - La explosión del interés global en IA

El lanzamiento de ChatGPT (OpenAI, 2022) en noviembre de 2022 marcó un hito en la historia de la inteligencia artificial, popularizando a gran escala los modelos generativos y mostrando un potencial de uso transversal a múltiples industrias. Tan solo en los primeros cinco días de su salida, ChatGPT (OpenAI, 2022) superó el millón de usuarios, convirtiéndose en una de las plataformas de más rápido crecimiento en la historia de la tecnología, ver Ilustración 4. Tiempo para alcanzar el millón de usuarios. Para 2024, informes recientes estiman que esta herramienta supera los 100 millones de

usuarios activos mensuales, un dato que refleja la magnitud de su adopción a nivel mundial (WiserNotify, 2024).

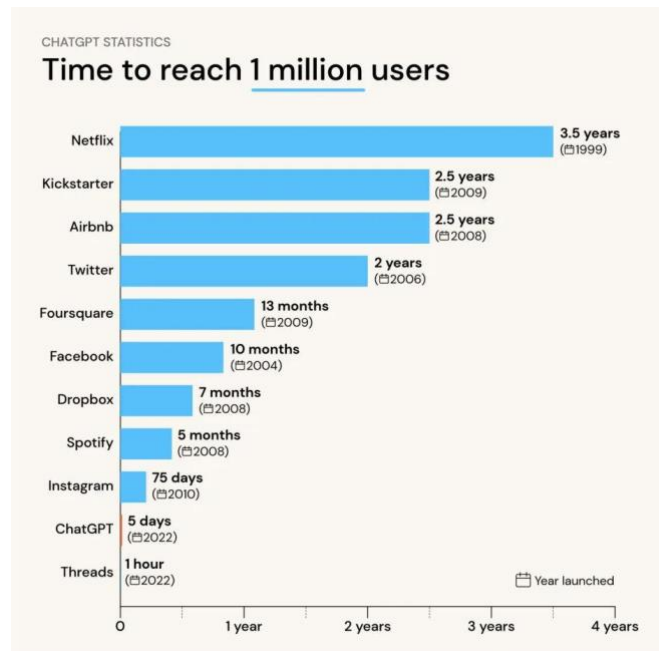


Ilustración 4. Tiempo para alcanzar el millón de usuarios

**Nota.** Adaptado de ChatGPT & Co. vs. Google: The AI Search Engine Showdown, por Tooltester, 2024, <https://www.tooltester.com/en/blog/chatgpt-vs-google/>

Detrás de este vertiginoso crecimiento se encuentra OpenAI, la organización responsable de ChatGPT (OpenAI, 2022), que en la actualidad ostenta una valoración cercana a los 29 mil millones de dólares. Este aumento en su valor de mercado es indicativo no solo de la adopción masiva por parte de usuarios y empresas, sino también de la confianza que los inversores han depositado en el potencial de la IA generativa como motor de innovación y disrupción en un futuro cada vez más digital.

Este auge no solo quedó limitado a ChatGPT (OpenAI, 2022), sino que rápidamente otras empresas tecnológicas comenzaron a desarrollar e implementar modelos similares. Entre los más destacados se encuentran:

- Google Bard (ahora Gemini): Alternativa de Google basada en modelos de la familia PaLM y Gemini (Google, 2023).
- Claude (Anthropic): Un modelo diseñado para ofrecer respuestas más alineadas con principios de seguridad y ética (Anthropic, 2023).
- LLAMA (Meta): Modelo de código abierto enfocado en el desarrollo comunitario y la transparencia en IA (Meta AI, 2023).
- Stable Diffusion: Herramienta especializada en la generación de imágenes a partir de texto (Stability AI, 2022).
- DALL-E: Herramienta especializada en la generación de imágenes a partir de texto (OpenAI, 2021).
- DeepSeek: Un modelo de código abierto con capacidades avanzadas de razonamiento matemático, generación de código y respuesta conversacional (DeepSeek AI, 2023).

Este crecimiento exponencial de modelos de IA generativa cambió la percepción de la inteligencia artificial en la sociedad, pasando de ser un concepto técnico y abstracto a una tecnología accesible y presente en la vida cotidiana.

### 2.2.2 - Impacto de la IA generativa en la industria

Según el informe global de McKinsey publicado en 2024 (McKinsey & Company, 2024), como se observa en Ilustración 5. Adopción de IA generativa en empresas a principios de 2024 la adopción de inteligencia artificial generativa por parte de las empresas ha crecido de forma acelerada. El 65% de las organizaciones encuestadas declaró utilizar IA generativa regularmente en al menos una función, casi el doble que el año anterior. Además, el 72% de las empresas ya ha adoptado algún tipo de tecnología de inteligencia artificial, alcanzando el nivel más alto en seis años.

Esta adopción se da principalmente en áreas como marketing y ventas, tecnología, y desarrollo de productos y servicios, y en muchos casos, las organizaciones están implementando en dos o más funciones simultáneamente. Según McKinsey, ya se observan beneficios tangibles: reducción de costos en funciones administrativas, como Recursos Humanos, y aumento de ingresos en procesos logísticos o de gestión de inventario. Estos resultados reflejan que la IA generativa ha dejado de ser una promesa para convertirse en un activo operativo y estratégico en muchas compañías.

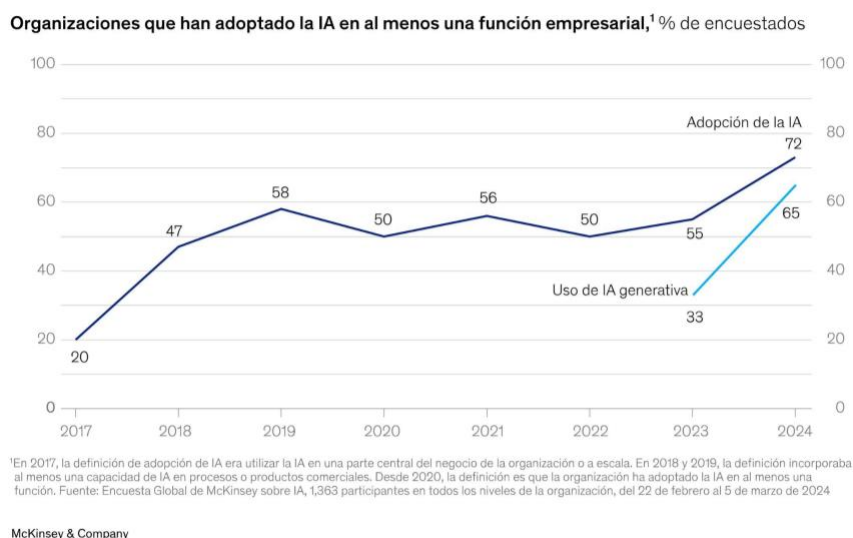


Ilustración 5. Adopción de IA generativa en empresas a principios de 2024

**Nota.** Adaptado de *El estado de la IA a principios de 2024*, por McKinsey & Company, <https://www.mckinsey.com/locations/south-america/latam/hispanoamerica-en-potencia/el-estado-de-la-ia-a-principios-de-2024-la-adopcion-de-la-ia-generativa-aumenta-y-comienza-a-generar-valor/es-CL>

### 2.2.3 - Impacto de la IA generativa en la sociedad

La inteligencia artificial generativa ya está comenzando a transformar aspectos concretos de la vida cotidiana de las personas. Según el informe global *Our Life with AI* (Google & Ipsos, 2023), elaborado

por Google e Ipsos en 2023, una parte significativa de la población percibe que esta tecnología ya está teniendo un impacto positivo en distintas dimensiones de su vida.

A través de una encuesta realizada en más de una docena de países, el informe revela que las áreas donde los encuestados perciben mayor impacto positivo en el presente son:

- Educación: 52%
- Salud y bienestar: 47%
- Capacidad de aprendizaje: 45%
- Comprensión de temas complejos: 43%
- Trabajo diario: 42%
- Situación financiera personal: 41%
- Calidad de vida: 39%
- Perspectivas de carrera: 38%
- Capacidad para escribir o crear: 38%

Estos datos (Google & Ipsos, 2023) reflejan que, más allá del potencial a futuro, una proporción considerable de la población ya percibe beneficios tangibles derivados del uso de la IA. Esto incluye desde el apoyo en actividades educativas hasta mejoras en la eficiencia laboral, la creatividad o incluso la comprensión de información compleja.

Aunque todavía existen desafíos en materia de acceso, regulación y educación digital, el nivel de adopción y la valoración positiva por parte de los usuarios sugieren que la IA generativa se está integrando de forma creciente en las rutinas personales y profesionales a escala global.

## 2.3 - El auge de los agentes inteligentes y sistemas multiagente

El surgimiento de los agentes inteligentes y los sistemas multiagente ha captado una atención creciente dentro del ecosistema de la inteligencia artificial moderna. En este nuevo paradigma, la idea de un único modelo de lenguaje cumpliendo todas las funciones da paso a arquitecturas colaborativas, donde múltiples agentes especializados interactúan entre sí para resolver tareas complejas de manera más eficiente, escalable y autónoma.

Este enfoque no solo responde a una necesidad técnica, sino también a una lógica organizacional: así como los equipos humanos distribuyen responsabilidades entre miembros con distintas habilidades, los sistemas multiagente adoptan una forma de inteligencia colectiva que permite mayor adaptabilidad y robustez. Este capítulo explora qué son los agentes, cómo funcionan los sistemas multiagente y por qué se han vuelto una pieza central en el futuro cercano de la IA aplicada.

### 2.3.1 - ¿Qué son los agentes y sistemas multiagente?

En el contexto de los modelos fundacionales (foundation models), un agente puede definirse como una instancia de un modelo de lenguaje con un rol específico, que posee autonomía, capacidad de razonamiento y la posibilidad de interactuar con el entorno o con otros agentes. Cada agente cuenta con un *perfil funcional*, por ejemplo, analista, planificador, ejecutor, y está equipado con memoria, herramientas y objetivos que guían su comportamiento (Wang, et al., 2024).

Un sistema multiagente (Multi-Agent Foundation Model System, o MAFM) consiste en una red de estos agentes, organizados para cooperar entre sí. Según Wang (Wang, et al., 2024), estos sistemas pueden clasificarse por:

- Tipo de arquitectura: jerárquica (con un líder o planificador), descentralizada (todos los agentes se comunican entre sí) o facilitada (coordinados por un módulo externo).
- Nivel de colaboración: desde agentes totalmente independientes hasta agentes que comparten memoria, herramientas y objetivos comunes.
- Grado de especialización: algunos agentes son generalistas, mientras que otros se diseñan para tareas puntuales.

La interacción entre agentes no es trivial: requiere protocolos de comunicación, gestión del contexto compartido y mecanismos para la resolución de conflictos o inconsistencias. En la práctica, los agentes suelen estar implementados en frameworks como AutoGen (Microsoft, 2023) o CrewAI (CrewAI, 2024), que permiten estructurar estas dinámicas de manera programática.

### 2.3.2 - Por qué están ganando popularidad

El auge de los sistemas multiagente se explica por varios factores convergentes:

- Limitaciones de los modelos únicos: Aunque los LLMs han avanzado enormemente, un solo modelo tiene restricciones cuando se enfrenta a tareas largas, contextos cambiantes o múltiples objetivos simultáneos. Dividir responsabilidades entre agentes mejora la eficiencia y la claridad de propósito.
- Especialización y modularidad: Al asignar tareas específicas a cada agente, los sistemas se vuelven más mantenibles, adaptables y fáciles de escalar. Esto imita principios de ingeniería de software como la separación de responsabilidades.
- Aprovechamiento del razonamiento colectivo: Varios estudios han demostrado que la interacción entre agentes mejora el rendimiento en tareas complejas, especialmente en planificación, toma de decisiones y ejecución en entornos inciertos (Wang, et al., 2024).
- Nuevas herramientas y entornos de desarrollo: Frameworks como LangChain (LangChain, 2023), CrewAI (CrewAI, 2024) y AutoGen (Microsoft, 2023) han facilitado la creación de sistemas multiagente, permitiendo a los desarrolladores definir agentes con roles, memorias, herramientas y ciclos de vida diferenciados.
- Aplicaciones reales de negocio: Desde automatización de procesos internos hasta copilotos de desarrollo o asistentes de ventas, los sistemas multiagente están encontrando casos de uso tangibles en múltiples industrias, incluyendo software, salud, finanzas y educación.

### 2.3.3 - Por qué son importantes los agentes en la actualidad y futuro

El interés en los agentes inteligentes y su organización en sistemas multiagente va más allá de la eficiencia técnica: plantea una forma radicalmente nueva de estructurar el trabajo, tanto humano como digital. Su relevancia futura se puede entender en tres niveles:

- Escalabilidad y resiliencia: Al distribuir la inteligencia y la carga de trabajo entre múltiples unidades, los sistemas se vuelven más robustos frente a fallos, y pueden escalar horizontalmente de forma más natural.
- Colaboración emergente: En lugar de secuencias rígidas, los sistemas multiagente permiten la aparición de soluciones complejas a partir de interacciones simples, al igual que sucede en los sistemas humanos o biológicos.
- Transformación organizacional: En empresas tecnológicas, los agentes podrían convertirse en unidades básicas de operación, permitiendo una nueva clase de workflows donde humanos y agentes co-trabajan, diseñando productos, escribiendo código, realizando pruebas, gestionando clientes o coordinando tareas sinérgicamente.

Según Wang (Wang, et al., 2024), los agentes representan un cambio de paradigma: pasamos de ver a los modelos de IA como "herramientas" a concebirlos como colaboradores autónomos. Esto redefine no sólo la interfaz entre humano y máquina, sino también los modelos de negocio, los procesos internos y las habilidades necesarias en el futuro del trabajo.

## 2.4 - Uso de la Inteligencia Artificial en diferentes verticales de negocio

La inteligencia artificial ha sido adoptada en una amplia variedad de sectores, impulsando mejoras en productividad, automatización y toma de decisiones. Su impacto ha transformado industrias clave, permitiendo optimizar procesos, reducir costos operativos y mejorar la experiencia de los usuarios. En este capítulo, se repasa su impacto en sectores como la salud, la educación y la manufactura para ilustrar cómo está revolucionando distintos ámbitos, aunque su alcance se extiende a muchas otras industrias, desde el comercio hasta la energía y la seguridad.

Si bien el objetivo principal de esta sección no es analizar la demanda de talento especializado en IA, el crecimiento exponencial en su adopción en múltiples sectores sugiere que el uso masivo de estas tecnologías podría generar una mayor demanda en la capa de aplicaciones. A medida que más industrias incorporan IA en sus procesos, la necesidad de integrar, personalizar y optimizar estas soluciones se vuelve cada vez más relevante, lo que podría impactar en la estructura del mercado laboral y en las habilidades requeridas en diversas disciplinas.

### 2.4.1 - IA en la educación: Personalización del aprendizaje y asistentes inteligentes

La inteligencia artificial ha transformado el ámbito educativo al permitir la personalización del aprendizaje y mejorar la interacción entre estudiantes y plataformas digitales. Mediante el uso de modelos de aprendizaje adaptativo y asistentes inteligentes, la IA ajusta contenidos y metodologías en función del desempeño y necesidades individuales de cada estudiante, optimizando su experiencia educativa (Abdul Rahman, Michael, & Lee, 2023).

Estudios recientes han demostrado que la implementación de IA en educación superior ha incrementado la retención del conocimiento y mejorado el desempeño académico en un 25% en comparación con métodos tradicionales (Sánchez, Torres, & Fernández, 2023). Esto se debe a la capacidad de los algoritmos de IA para analizar patrones de aprendizaje y predecir dificultades antes de que el estudiante se enfrente a ellas.

#### Aprendizaje adaptativo y asistentes inteligentes

El aprendizaje adaptativo basado en IA permite ajustar dinámicamente la dificultad de los contenidos, recomendar ejercicios específicos y ofrecer retroalimentación inmediata a los estudiantes. Plataformas como Coursera (Coursera, 2023) y edX (edX, 2024) han implementado estos modelos para personalizar la experiencia de cada usuario y mejorar la efectividad del aprendizaje.

Por otro lado, los asistentes inteligentes, como los chatbots educativos, han demostrado ser herramientas eficaces en la educación superior. Un estudio en universidades mostró que el 80% de los estudiantes que usaron asistentes virtuales mejoraron su autonomía y motivación (Sánchez, Torres, & Fernández, 2023). Estos sistemas pueden:

- Responder preguntas frecuentes sobre contenidos académicos.
- Ofrecer explicaciones detalladas en tiempo real.
- Analizar el desempeño y sugerir estrategias de estudio personalizadas.

En universidades y plataformas de educación en línea, estos asistentes han reducido la carga administrativa de los docentes, permitiendo que se enfoquen en tareas más estratégicas (Abdul Rahman, Michael, & Lee, 2023).

#### Desafíos y oportunidades de la IA en educación

La incorporación de inteligencia artificial en el ámbito educativo ofrece oportunidades significativas, pero también plantea una serie de desafíos que requieren una profunda reflexión pedagógica, tecnológica y ética. Según Abdul Rahman, Michael y Lee (Abdul Rahman, Michael, & Lee, 2023), el uso de asistentes inteligentes habilitados por IA para el aprendizaje personalizado y adaptativo en la educación superior puede generar impactos transformadores, siempre que se integren de manera cuidadosa y responsable.

Entre las principales oportunidades se destacan:

- Personalización del aprendizaje: los sistemas inteligentes permiten adaptar los contenidos y métodos a las necesidades individuales de cada estudiante, mejorando la retención del conocimiento y fomentando un aprendizaje más autónomo.
- Acceso ampliado al conocimiento: la IA puede facilitar el acceso a materiales educativos personalizados y de calidad, superando barreras geográficas, económicas o idiomáticas.
- Asistencia en tiempo real: los chatbots y tutores virtuales permiten brindar acompañamiento continuo, resolver dudas inmediatas y reforzar conceptos sin requerir la presencia permanente del docente.

A pesar de su potencial, la integración de la IA en educación también presenta desafíos relevantes, entre ellos:

- La necesidad de redefinir los métodos de evaluación: el uso de asistentes inteligentes capaces de resolver ejercicios, redactar textos o sintetizar información obliga a repensar cómo se evalúa el aprendizaje. El riesgo es que el estudiante delegue en la tecnología procesos que antes implicaba esfuerzo cognitivo, dificultando la validación de la comprensión real. Esto demanda desarrollar nuevas formas de evaluación que valoren la reflexión, el análisis crítico y la aplicación práctica del conocimiento.
- La dependencia tecnológica y la brecha digital: si bien la IA puede democratizar el acceso, también corre el riesgo de profundizar desigualdades si no se acompaña con políticas inclusivas y capacitación adecuada para docentes y estudiantes.
- Cuestiones éticas y de privacidad: el uso de datos personales para entrenar algoritmos educativos plantea interrogantes sobre el consentimiento, el almacenamiento seguro de la información y la posible reproducción de sesgos algorítmicos.

En resumen, la IA en la educación tiene el potencial de transformar profundamente los procesos de enseñanza y aprendizaje, pero su implementación debe estar guiada por principios pedagógicos claros, garantías éticas y marcos de evaluación acordes al nuevo contexto.

#### 2.4.2 - IA en el entretenimiento y la creatividad: La revolución del contenido digital

La inteligencia artificial (IA) ha revolucionado la industria del entretenimiento y la creatividad, permitiendo la generación de contenido digital de manera automatizada y facilitando la colaboración entre humanos y sistemas inteligentes. Desde la creación de guiones hasta la composición musical y la edición de video, los avances en IA generativa han transformado los flujos de trabajo y han ampliado las posibilidades de expresión artística (Anantrasirichai, Zhang, & Bull, 2025)

Los modelos avanzados de IA, como transformers y redes neuronales profundas, han mejorado la calidad de la producción de contenido, logrando resultados que antes requerían intervención manual. A medida que la IA sigue evolucionando, su impacto en la creatividad continúa expandiéndose, impulsando la personalización del entretenimiento y redefiniendo el papel de los creadores humanos (Fernández, Liu, & Rossi, 2025).

#### IA generativa en la creación de contenido digital

El auge de la IA generativa ha permitido la creación de contenido original en múltiples formatos, como texto, imágenes, música y video. Herramientas como DALL·E (OpenAI, 2021), MidJourney (MidJourney, 2023) y Stable Diffusion (Stability AI, 2022) han facilitado la generación de imágenes de alta calidad a partir de descripciones textuales, lo que ha sido adoptado en industrias como la publicidad y el diseño gráfico.

En el ámbito audiovisual, modelos avanzados han sido utilizados para la creación de efectos visuales y la postproducción de películas, optimizando procesos que anteriormente requerían costosos equipos de producción (Anantrasirichai, Zhang, & Bull, 2025). Un estudio reciente destaca que la IA ha reducido los tiempos de edición de video en un 40%, permitiendo una producción más eficiente y accesible para creadores independientes y grandes estudios (Fernández, Liu, & Rossi, 2025).

La industria musical también ha sido transformada por la IA. Modelos como Jukebox (OpenAI, 2020) y MusicLM (Google Research, 2023) pueden componer piezas originales en diversos géneros, adaptándose a estilos específicos y permitiendo a artistas experimentar con nuevas formas de expresión. Además, la IA ha facilitado la restauración de grabaciones antiguas y la mejora de calidad en producciones de audio (Anantrasirichai, Zhang, & Bull, 2025).

#### Personalización del entretenimiento y desafíos éticos

Además de la generación de contenido, la IA ha mejorado la personalización de la experiencia del usuario en plataformas de streaming y redes sociales. Algoritmos avanzados analizan patrones de consumo para recomendar contenido adaptado a las preferencias individuales de cada usuario, optimizando la interacción con el entretenimiento digital (Fernández, Liu, & Rossi, 2025).

Sin embargo, el crecimiento de la IA en el ámbito creativo también plantea desafíos éticos y regulatorios. Algunos de los principales puntos de debate incluyen:

- Derechos de autor y propiedad intelectual: ¿Quién es el propietario de una obra creada por IA?
- Sesgos en la generación de contenido: La IA puede replicar estereotipos si los datos de entrenamiento no son equilibrados.
- Autenticidad y creatividad humana: Existe preocupación sobre cómo la IA afecta el valor del trabajo de artistas y creadores.

A pesar de estos desafíos, el avance de la IA en la creatividad sigue siendo imparable. Con el desarrollo de nuevas regulaciones y el uso responsable de estas tecnologías, la inteligencia artificial seguirá siendo una herramienta clave para potenciar la creatividad humana y expandir los límites del entretenimiento digital (Anantrasirichai, Zhang, & Bull, 2025).

#### 2.4.3 - IA en la industria y manufactura: Optimización de procesos y mantenimiento predictivo

La inteligencia artificial (IA) ha revolucionado la industria y la manufactura, permitiendo la optimización de procesos y la implementación de estrategias avanzadas de mantenimiento predictivo. Gracias a la combinación de Big Data, Internet Industrial de las Cosas (IIoT) y modelos de aprendizaje

automático, las empresas pueden mejorar la eficiencia operativa, reducir costos y minimizar tiempos de inactividad no planificados (Smith, Wang, & Rodríguez, 2020).

El mantenimiento predictivo basado en IA permite anticipar fallos en maquinaria antes de que ocurran, mediante el análisis en tiempo real de datos provenientes de sensores y sistemas de monitoreo. Además, tecnologías como los gemelos digitales han cobrado relevancia, al simular el comportamiento de equipos industriales para optimizar estrategias de mantenimiento y prever fallos con mayor precisión (Patel, Müller, & Zhang, 2024).

#### Mantenimiento predictivo con IA y gemelos digitales

Uno de los avances más significativos en la industria es la integración de IA y gemelos digitales en el mantenimiento predictivo. Un gemelo digital es una representación virtual de un sistema físico que recibe datos en tiempo real de sensores conectados, permitiendo el monitoreo y análisis continuo del estado de los equipos (Patel, Müller, & Zhang, 2024).

Estudios recientes han demostrado que la implementación de gemelos digitales en la manufactura ha logrado:

- Reducir en un 25% los costos de mantenimiento al predecir fallos con mayor precisión.
- Optimizar la vida útil de los equipos, ajustando dinámicamente los parámetros de operación.
- Minimizar el tiempo de inactividad al planificar mantenimientos en función de la predicción de fallos.

El uso de IA en mantenimiento predictivo ha sido adoptado en industrias como la automotriz, la aeronáutica y la energética, donde la confiabilidad de los sistemas es crítica para la operación continua (Smith, Wang, & Rodríguez, 2020).

#### Optimización de procesos industriales con IA e IIoT

La IA también ha transformado la gestión de procesos industriales, facilitando la automatización de decisiones y la optimización en tiempo real. Gracias al uso de IIoT y aprendizaje automático, las fábricas pueden:

- Ajustar dinámicamente las velocidades de producción en función de la demanda.
- Detectar ineficiencias en la cadena de suministro y reducir desperdicios.
- Implementar sistemas de control de calidad basados en visión por computadora (Smith, Wang, & Rodríguez, 2020).

Un estudio reciente destaca que la integración de IA e IIoT ha mejorado la eficiencia operativa hasta en un 30%, permitiendo a las empresas ajustar su producción en tiempo real para maximizar el rendimiento y reducir costos (Patel, Müller, & Zhang, 2024)

A pesar de sus beneficios, la adopción de IA en la industria enfrenta desafíos, como la seguridad en la gestión de datos, la necesidad de infraestructura tecnológica avanzada y la capacitación del personal para operar y supervisar estos sistemas. Sin embargo, a medida que las tecnologías continúan evolucionando, se espera que la IA juegue un papel aún más relevante en la manufactura del futuro (Smith, Wang, & Rodríguez, 2020).

## CAPÍTULO 3 - IA Y SU IMPACTO EN EL DESARROLLO DE SOFTWARE

En capítulos anteriores se exploró el impacto de la inteligencia artificial (IA) en diversas industrias, desde la educación y el entretenimiento hasta la manufactura y la logística, donde ha optimizado procesos, automatizado tareas y mejorado la toma de decisiones. Sin embargo, uno de los usos más revolucionarios de la IA es su aplicación en el desarrollo de software, un área donde está redefiniendo la manera en que los desarrolladores trabajan y colaboran.

Este capítulo analiza el impacto práctico de la IA en el desarrollo de software, explorando las tecnologías clave utilizadas en programación, su influencia en la productividad y la evolución del rol del desarrollador. Además, se examinan estudios de casos que evidencian cómo estas herramientas están optimizando los procesos en la industria y qué desafíos plantea su adopción.

### 3.1 - Herramientas y Tecnologías de IA Aplicadas al Desarrollo de Software

El avance de la IA ha impulsado el desarrollo de herramientas que optimizan distintas fases del proceso de programación. Desde modelos capaces de generar código hasta sistemas que mejoran la detección de errores y la automatización de pruebas, estas soluciones están cambiando la manera en que los desarrolladores construyen software.

En 2023, Github reveló que más del 30% del código escrito en su plataforma proviene de herramientas como GitHub Copilot (GitHub, n.d.), lo que refleja un cambio significativo en la forma en que los programadores interactúan con su entorno de desarrollo (GitHub, 2023).

#### 3.1.1 - Modelos de IA usados en programación

La integración de modelos de inteligencia artificial en el desarrollo de software ha evolucionado rápidamente, con herramientas diseñadas para asistir en múltiples etapas del proceso de programación. Estas herramientas pueden agruparse en dos grandes categorías: aquellas que se integran a entornos de desarrollo existentes, y aquellas que constituyen entornos de desarrollo en sí mismas.

Las primeras comenzaron a emerger con fuerza entre 2021 y 2022. Se trata de asistentes de código como GitHub Copilot (GitHub, n.d.), Amazon CodeWhisperer (Amazon Web Services, n.d.) y Tabnine (Tabnine, n.d.), que funcionan como extensiones dentro de IDEs tradicionales como Visual Studio Code (Microsoft, n.d.) o IntelliJ IDEA (JetBrains, n.d.). Estas soluciones actúan como copilotos del desarrollador, ofreciendo sugerencias de código en tiempo real, completando funciones, generando pruebas unitarias o incluso explicando fragmentos existentes, siempre dentro del flujo de trabajo habitual del programador.

Por otro lado, una segunda ola de innovación ha dado lugar a herramientas más recientes, que surgieron entre 2023 y 2024, como Cursor (Cursor, n.d.) y Windsurf (Windsurf, n.d.). A diferencia de las anteriores, estas no se integran a un IDE, sino que lo reemplazan. Son entornos de desarrollo creados desde cero con inteligencia artificial como componente central. Estas plataformas permiten no solo escribir código, sino también interactuar con un asistente virtual mediante chat, aceptar o descartar sugerencias con un solo clic, ejecutar comandos en una consola integrada, y hasta seleccionar qué modelo de lenguaje (LLM) utilizar. La experiencia es más fluida, conversacional y centrada en la colaboración entre humano e IA, lo que marca un cambio sustancial en la dinámica tradicional de programación.

En la Ilustración 6. Mapa de herramientas de asistencia al desarrollo con IA se presenta un resumen visual de diversas herramientas clasificadas según dos dimensiones: el grado de especialización (de genéricas a especializadas) y el tipo de interacción que ofrecen (desde asistentes que ayudan al programador hasta agentes más autónomos). Este mapa permite entender el panorama actual de soluciones disponibles y cómo varían en su nivel de automatización y enfoque funcional.

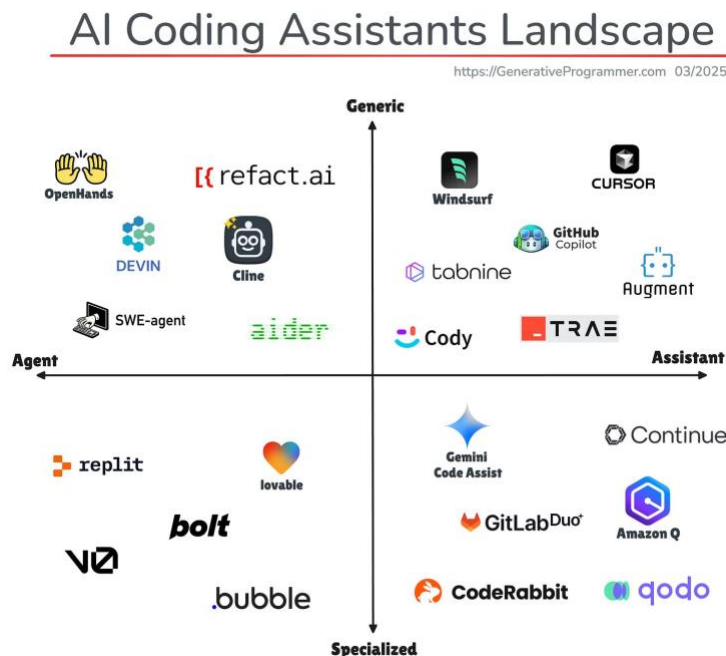


Ilustración 6. Mapa de herramientas de asistencia al desarrollo con IA

**Nota.** Adaptado de *AI Coding Assistants Landscape*, por Generative Programmer, <https://generativeprogrammer.com/p/ai-coding-assistants-landscape>

Más allá de la generación de código, los modelos de IA han ampliado su funcionalidad, asistiendo en diversas tareas dentro del desarrollo de software, tales como:

- **Testing:** Generación automatizada de pruebas unitarias e integración, agilizando la validación del código.
- **Documentación:** Creación de documentación técnica y comentarios en el código para facilitar su mantenimiento.
- **Revisión de código:** Análisis de código en busca de errores, sugerencias de optimización y detección de vulnerabilidades.
- **Escritura de tiques:** Redacción automatizada de tiques en plataformas como Jira (Atlassian, 2024) o Trello (Atlassian, 2024), facilitando la gestión del trabajo.

A medida que la adopción de estos modelos sigue en aumento, también lo hace su impacto en la dinámica de trabajo de los desarrolladores. La evolución de estas herramientas está dando lugar a un nuevo paradigma en el desarrollo de software, en el que las habilidades tradicionales de programación deben complementarse con nuevas competencias, como la validación del código generado por IA y la capacidad de optimizar las interacciones con estos modelos.

### 3.1.2 - Adopción de herramientas AI según JetBrains

JetBrains es una reconocida empresa de desarrollo de software conocida por crear entornos de desarrollo integrados (IDEs) ampliamente utilizados como IntelliJ IDEA (JetBrains, n.d.), PyCharm (JetBrains, 2024), WebStorm (JetBrains, 2010) y otros. Además de sus productos, la compañía lleva adelante una encuesta anual llamada *State of Developer Ecosystem*, que busca capturar un panorama amplio del mundo del desarrollo de software a través de preguntas sobre herramientas, lenguajes, hábitos y percepciones. En su edición 2024, el informe se elaboró a partir de las respuestas de 23.262 desarrolladores de todo el mundo, lo que le otorga un alto grado de representatividad sobre las tendencias y comportamientos actuales de la comunidad técnica global (JetBrains, Developer Ecosystem Survey 2024, 2024).

El informe de 2024 (JetBrains, Developer Ecosystem Survey 2024, 2024), revela que el 69% de los desarrolladores ha probado herramientas de IA como ChatGPT (OpenAI, 2022) o GitHub Copilot (GitHub, n.d.) en tareas de programación, y un 49% utiliza regularmente ChatGPT (OpenAI, 2022) en ese contexto. GitHub Copilot, por su parte, fue probado por el 40% de los encuestados y es usado con frecuencia por el 26%. Estas cifras reflejan una adopción creciente de herramientas basadas en inteligencia artificial en el ámbito del desarrollo de software. Si bien muchos programadores aún se encuentran en una etapa exploratoria, los datos muestran que una parte considerable ya incorporó estas soluciones a su flujo de trabajo diario.

Entre quienes utilizan este tipo de herramientas, el 46% lo hace de forma ocasional, mientras que el 31% las emplea con regularidad. Las funcionalidades más valoradas incluyen la generación de código, la explicación de fragmentos existentes y la asistencia en la depuración “debugging”. En la Ilustración 7. Percepción de beneficios de IA según encuestas a desarrolladores, se puede observar los beneficios más comunes que encuentran los desarrolladores del uso de estas herramientas.



Ilustración 7. Percepción de beneficios de IA según encuestas a desarrolladores

**Nota.** Adaptado de *The State of Developer Ecosystem 2024*, por JetBrains, 2024, <https://www.jetbrains.com/lp/devecosystem-2024/>

En cuanto al impacto percibido en la productividad, como se observa en Ilustración 8. Ahorro de tiempo percibido por el uso de IA en el desarrollo, el 32% de los desarrolladores indicó que las herramientas de IA les permiten ahorrar entre 1 y menos de 2 horas por semana, mientras que un 27% afirma ahorrar entre 2 y menos de 4 horas. Solo un 4% respondió que no experimenta ningún

ahorro de tiempo. Estos datos sugieren que, para la mayoría de los desarrolladores, las herramientas de IA están contribuyendo de forma tangible a optimizar su flujo de trabajo, especialmente en tareas repetitivas o rutinarias.



Ilustración 8. Ahorro de tiempo percibido por el uso de IA en el desarrollo

**Nota.** Adaptado de *The State of Developer Ecosystem 2024*, por JetBrains, 2024, <https://www.jetbrains.com/lp/devecosystem-2024/>

### 3.1.3 - Adopción de herramientas AI según Stack Overflow

Stack Overflow es una de las plataformas más influyentes en la comunidad global de desarrollo de software, utilizada por millones de programadores para compartir conocimiento, resolver problemas técnicos y mantenerse actualizados. Desde hace más de una década, Stack Overflow realiza una encuesta anual llamada *Developer Survey*, que recoge información detallada sobre herramientas, lenguajes, tecnologías emergentes, entorno laboral y percepciones del sector. La edición 2024 de la encuesta (Stack Overflow, 2024), realizada en mayo, contó con la participación de 65.000 desarrolladores de todo el mundo, lo que la convierte en una de las fuentes más completas y representativas sobre el estado actual del ecosistema de programación.

Según la encuesta anual de Stack Overflow de 2024, el 76% de los desarrolladores indicó que ya está utilizando o planea utilizar herramientas de IA en su flujo de trabajo, ver Ilustración 9. Uso de herramientas de IA en el proceso de desarrollo. Entre los usuarios actuales, GitHub Copilot (GitHub, n.d.) lidera la adopción, seguido por ChatGPT (OpenAI, 2022) y Amazon CodeWhisperer (Amazon Web Services, n.d.). El uso de herramientas de IA ha mostrado una evolución significativa en apenas un año. En 2023, el 44% de los encuestados afirmaba utilizarlas semanalmente. Para 2024, esa cifra aumentó al 62%, consolidando a la inteligencia artificial como una tecnología ya incorporada de forma rutinaria por una mayoría de desarrolladores.

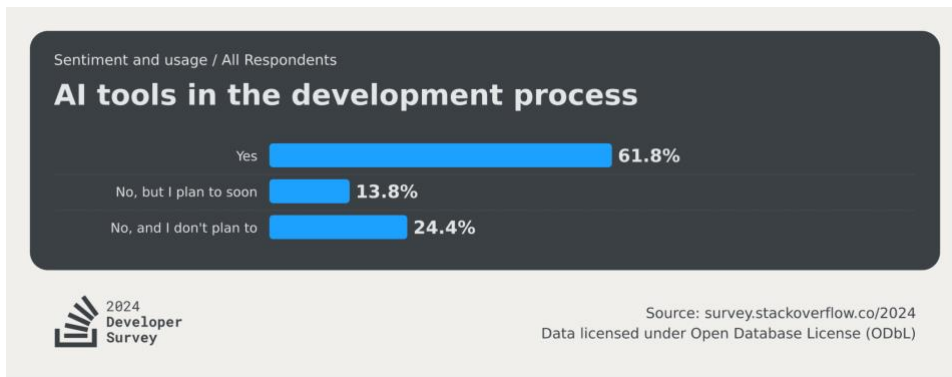


Ilustración 9. Uso de herramientas de IA en el proceso de desarrollo

**Nota.** Adaptado de *Developer Survey 2024 – Artificial Intelligence*, por Stack Overflow, 2024, <https://survey.stackoverflow.co/2024/ai>

En cuanto a las percepciones, de acuerdo a Ilustración 10. Sentimiento hacia herramientas de IA entre desarrolladores, el 72% de los desarrolladores manifiesta una actitud positiva hacia el uso de herramientas de IA, lo que refleja una aceptación generalizada y una valoración favorable de su incorporación en los procesos de desarrollo.

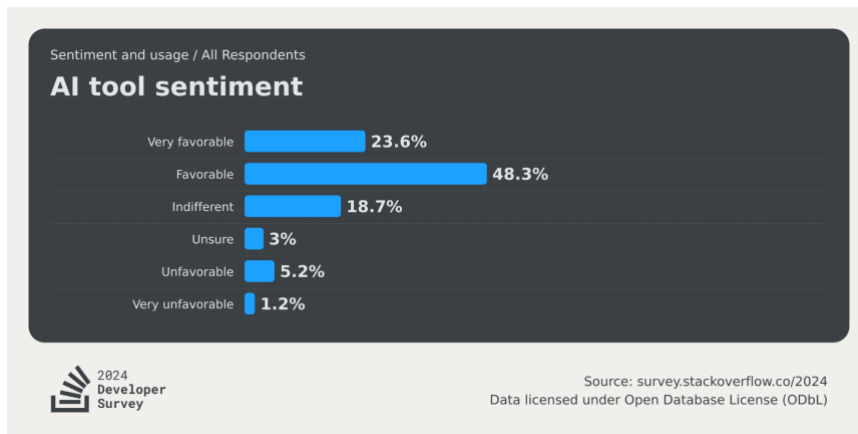


Ilustración 10. Sentimiento hacia herramientas de IA entre desarrolladores

**Nota.** Adaptado de *Developer Survey 2024 – Artificial Intelligence*, por Stack Overflow, 2024, <https://survey.stackoverflow.co/2024/ai>

En cuanto a las áreas donde los desarrolladores sienten que la IA ya los está ayudando en su trabajo, las respuestas se concentran en mejoras dentro del flujo técnico diario. Según la encuesta, el 82% reporta que la IA les permite ser más productivos, el 53% destaca que ayuda a reducir trabajo repetitivo, y el 45% afirma que mejora la calidad del código. Además, un 35% considera que la IA les permite enfocarse en tareas más importantes. Estos datos se resumen en la Ilustración 11. Beneficio de herramientas IA entre desarrolladores, que presenta los principales beneficios percibidos por los desarrolladores al utilizar herramientas de IA. Estos resultados reflejan que los desarrolladores perciben beneficios concretos y actuales en su productividad y calidad de trabajo al incorporar herramientas de inteligencia artificial en su práctica profesional.

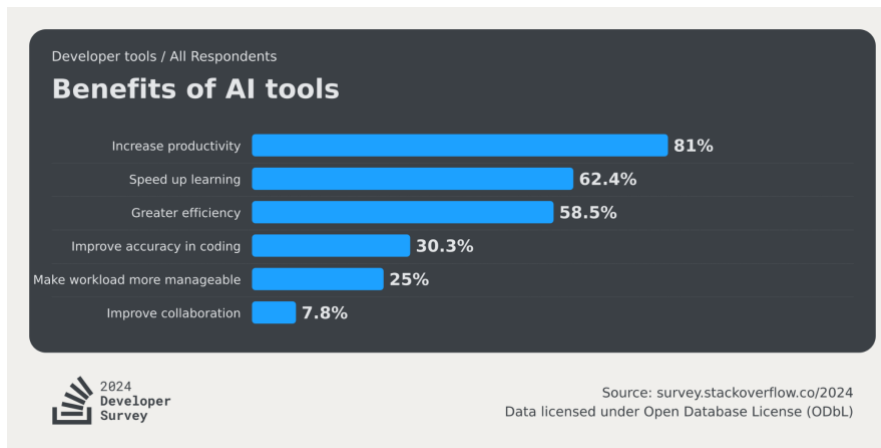


Ilustración 11. Beneficio de herramientas IA entre desarrolladores

**Nota.** Adaptado de *Developer Survey 2024 – Artificial Intelligence*, por Stack Overflow, 2024, <https://survey.stackoverflow.co/2024/ai>

### 3.1.4 - Limitaciones y desafíos de la IA en entornos de desarrollo

Si bien las herramientas de inteligencia artificial aplicadas al desarrollo de software han demostrado ser útiles en diversas tareas, su incorporación también ha revelado una serie de limitaciones y desafíos que deben ser considerados por individuos y organizaciones. Entre las principales preocupaciones se encuentra la calidad del código generado. Según un estudio reciente de Kalliamvakou (Kalliamvakou, et al., 2024) los desarrolladores reportan que los asistentes de codificación basados en IA, como GitHub Copilot (GitHub, n.d.), pueden generar código que contiene errores sintácticos, problemas semánticos o soluciones técnicamente correctas pero subóptimas. En consecuencia, la revisión activa por parte del desarrollador sigue siendo esencial, ya que no se puede confiar ciegamente en las respuestas ofrecidas por el modelo.

Otra limitación significativa es la capacidad limitada de estos modelos para comprender el contexto completo del sistema o proyecto en desarrollo. Si bien pueden ser efectivos para tareas localizadas, como completar una función o generar una prueba unitaria, presentan dificultades para trabajar con múltiples archivos, flujos de ejecución complejos o convenciones específicas de un equipo o empresa. Esta falta de entendimiento global puede llevar a inconsistencias o errores difíciles de detectar (Kalliamvakou, et al., 2024).

El estudio también destaca el fenómeno del desaprendizaje técnico. Algunos desarrolladores expresaron preocupación por la posibilidad de volverse dependientes de las sugerencias automáticas, lo cual podría limitar el desarrollo de habilidades fundamentales como la resolución de problemas, el diseño algorítmico y el conocimiento profundo de los lenguajes de programación. Esta percepción fue frecuente entre usuarios habituales de estas herramientas (Kalliamvakou et al., 2024).

Por último, se identificaron desafíos de tipo organizacional y legal. Muchas empresas aún no han definido políticas claras sobre el uso de herramientas de IA generativa, especialmente en relación con la privacidad del código fuente, el posible envío de información sensible a servidores externos, y la ambigüedad sobre la propiedad intelectual del código generado. Estos temas se suman a una creciente necesidad de gobernanza y regulación en torno a la adopción de estas tecnologías (Kalliamvakou, et al., 2024).

En resumen, si bien los asistentes de codificación representan un avance significativo, su implementación debe hacerse con una comprensión clara de sus limitaciones, acompañada de prácticas de revisión, documentación y resguardo ético y legal.

## 3.2 - IA y Productividad en el Desarrollo de Software

En secciones anteriores se presentó la percepción de los desarrolladores respecto al impacto de la inteligencia artificial en su productividad, a partir de datos recopilados por encuestas de referencia como las de Stack Overflow (Stack Overflow, 2024) y JetBrains (JetBrains, Developer Ecosystem Survey 2024, 2024). Estas fuentes reflejan que una porción significativa de los desarrolladores visualiza mejoras en su eficiencia diaria, reducción de trabajo repetitivo y optimización de tareas técnicas gracias al uso de herramientas basadas en IA.

En este apartado se analizan estudios de carácter académico y científico que, a diferencia de las encuestas perceptuales, se enfocan en medir el impacto concreto en términos de actividad y rendimiento de los desarrolladores. Estas investigaciones aportan evidencia más estructurada sobre cómo las herramientas de IA están modificando el flujo de trabajo y los entregables de los equipos de desarrollo.

### 3.2.1 - Estudio de Microsoft (2023): IA y productividad en una muestra diversa de desarrolladores

En 2023, Microsoft realizó un estudio (Cui, et al., 2025) con 4.867 desarrolladores en tres empresas: Microsoft, Accenture y una compañía Fortune 100. Su objetivo era evaluar el impacto de GitHub Copilot (GitHub, n.d.) en la productividad de los desarrolladores midiendo el rendimiento de dos grupos: uno que utilizó Copilot y otro que trabajó sin asistencia de IA.

Los resultados mostraron un incremento del 26,08 % en la productividad de los desarrolladores que usaron Copilot, medido en términos de cantidad de tareas completadas en menor tiempo (Cui, et al., 2025). Además, el estudio destacó que los desarrolladores menos experimentados fueron quienes más se beneficiaron de la herramienta, ya que pudieron completar tareas con mayor rapidez y reducir el tiempo dedicado a depuración y corrección de errores.

Uno de los puntos clave del estudio fue la diferencia en cómo se usó la IA en distintos niveles de experiencia. Los desarrolladores junior aprovecharon Copilot como un mecanismo de aprendizaje acelerado, mientras que los más experimentados utilizaron la herramienta para reducir tareas repetitivas. Sin embargo, el estudio también identificó que una adopción excesiva de la IA puede generar una dependencia poco saludable en los desarrolladores menos experimentados si no se combina con una comprensión adecuada del código.

### 3.2.2 - Estudio de Google: IA y su impacto en el desarrollo empresarial

El mismo año, Google llevó a cabo un experimento (Paradis, et al., 2024) con 96 ingenieros de software en el que se evaluó el impacto de herramientas de IA como AI Code Completion y Smart Paste en la velocidad y calidad del desarrollo. A diferencia del estudio de Microsoft, este experimento se centró en cómo la IA influía en desarrolladores con distintos niveles de experiencia y tipos de tareas asignadas.

Los resultados indicaron que los desarrolladores que utilizan herramientas de IA completaron tareas un 21 % más rápido en comparación con aquellos que no usaron IA (Paradis, et al., 2024). Sin embargo, a diferencia del estudio de Microsoft, Google encontró que los desarrolladores más experimentados fueron los más beneficiados. Esto se debe a que quienes tenían mayor conocimiento previo en

desarrollo lograban interpretar mejor las sugerencias de la IA, hacer ajustes más eficientes y detectar errores generados por la herramienta con mayor facilidad.

El estudio también sugirió que el beneficio de la IA no es uniforme en todas las tareas de desarrollo. Si bien la generación de código y la refactorización fueron actividades altamente optimizadas por la IA, la planificación, la integración de sistemas complejos y la toma de decisiones arquitectónicas continuaron dependiendo en gran medida del criterio humano.

### 3.2.3 - Evolución de las herramientas de IA y perspectivas futuras

Los estudios científicos presentados anteriormente, realizados en 2023, muestran mejoras concretas en productividad mediante pruebas empíricas controladas con desarrolladores en ambientes reales. Estas investigaciones demostraron que el uso de asistentes de programación basados en IA puede acelerar la ejecución de tareas, reducir errores y aumentar la eficiencia en distintos entornos. A su vez, las encuestas de JetBrains (JetBrains, Developer Ecosystem Survey 2024, 2024) y Stack Overflow (Stack Overflow, 2024) reflejan percepciones similares: una mayoría de desarrolladores reconoce que la IA ya les está ayudando a ser más productivos, reducir trabajo repetitivo y mejorar la calidad del código.

Sin embargo, es importante considerar que estas encuestas fueron realizadas en los primeros meses de 2024, cuando la adopción de herramientas más avanzadas, como IDEs nativamente potenciados por IA (por ejemplo, Cursor y Windsurf), aún era incipiente. Estas plataformas, que comenzaron a ganar visibilidad a mediados de ese año, permiten una interacción mucho más profunda y contextual con la IA: los desarrolladores pueden chatear directamente con el asistente, ejecutar la consola dentro del IDE, elegir qué modelo de lenguaje utilizar, y beneficiarse de una comprensión más amplia del contexto del proyecto completo. Por lo tanto, es razonable suponer que, hacia 2025, el impacto positivo de la IA en la productividad del desarrollo de software será incluso mayor al reportado en los estudios y encuestas más recientes.

En este contexto de evolución continua, las perspectivas a futuro señalan una creciente integración de la IA en todas las fases del ciclo de desarrollo, desde el diseño y la codificación hasta las pruebas, el debugging y el mantenimiento. Esto no solo incrementa la velocidad de entrega y la calidad del producto final, sino que también transforma el rol del desarrollador, promoviendo un enfoque más estratégico, creativo y colaborativo.

## CAPÍTULO 4: SOFTWARE, DEMANDA HISTÓRICA Y ROL DE LA CONSULTORÍA TECNOLÓGICA

La industria del software ha experimentado una transformación significativa en las últimas décadas, impulsada por la creciente demanda de soluciones tecnológicas en diversos sectores. Factores como la digitalización de procesos empresariales, la adopción de tecnologías emergentes y la necesidad de innovación constante han sido motores clave de este crecimiento. Todo este fenómeno se enmarca en lo que se conoce como transformación digital, un proceso que implica la integración de tecnologías digitales en todas las áreas de una organización, cambiando fundamentalmente cómo opera y aporta valor a sus clientes (Coursera, 2023). Esta transformación forma parte de un cambio más amplio conocido como Industria 4.0 (IBM), que se refiere a una nueva etapa en la evolución industrial impulsada por tecnologías como la inteligencia artificial, el internet de las cosas y la automatización avanzada. En este contexto, muchas fábricas y empresas adoptan sistemas que les permiten recopilar datos en tiempo real, automatizar procesos y tomar decisiones sin intervención humana, con el objetivo de ser más eficientes, flexibles y competitivas.

Uno de los componentes esenciales en este ecosistema son las consultorías de software o empresas de servicios de tecnología, que brindan servicios especializados en desarrollo, implementación y mantenimiento de soluciones tecnológicas. Estas empresas han permitido a organizaciones de distintos tamaños y sectores acceder a conocimientos técnicos avanzados, optimizando sus operaciones y mejorando su competitividad en el mercado.

Este capítulo examina cómo ha evolucionado la demanda de software en las últimas décadas y el papel que han desempeñado las consultorías tecnológicas en la industria. Se explorarán los principales hitos que han marcado el crecimiento del sector, los distintos modelos de consultoría y la relevancia de estas empresas en la transformación digital de diversas industrias.

### 4.1 - Evolución de la demanda de software desde el año 2000: hitos y tendencias recientes

Este capítulo analiza la evolución del mercado de desarrollo de software en los últimos 25 años, con foco en los principales hitos que marcaron cambios en la demanda de servicios y en las prioridades tecnológicas de las organizaciones. Para su elaboración, se tomó como base el análisis y la información presentada por Gergely Orosz en *The Pragmatic Engineer*, especialmente en dos artículos recientes que abordan en profundidad la transformación del mercado tecnológico en el contexto postpandemia y la influencia del entorno macroeconómico sobre la industria: “The State of the Tech Engineering Market in 2024” (Orosz, *The Pragmatic Engineer*, 2024) y “ZIRP Software Engineers” (Orosz, *ZIRP software engineers*, 2024).

El periodo se divide en cuatro etapas principales que permiten comprender cómo se llegó al escenario actual, caracterizado por la desaceleración de la demanda, el ajuste de las estructuras de costos, y una creciente presión por mejorar la productividad y justificar cada inversión en desarrollo. La narrativa propuesta contextualiza estos cambios para entender mejor el surgimiento de tecnologías como los asistentes de código basados en IA, y el tipo de necesidades organizacionales a las que responden. En la Ilustración 12. Eventos claves en la industria del software se pueden observar diferentes eventos que tuvieron un alto impacto en la industria del software y que fueron parte de cada una de las cuatro etapas principales que se describen a continuación.

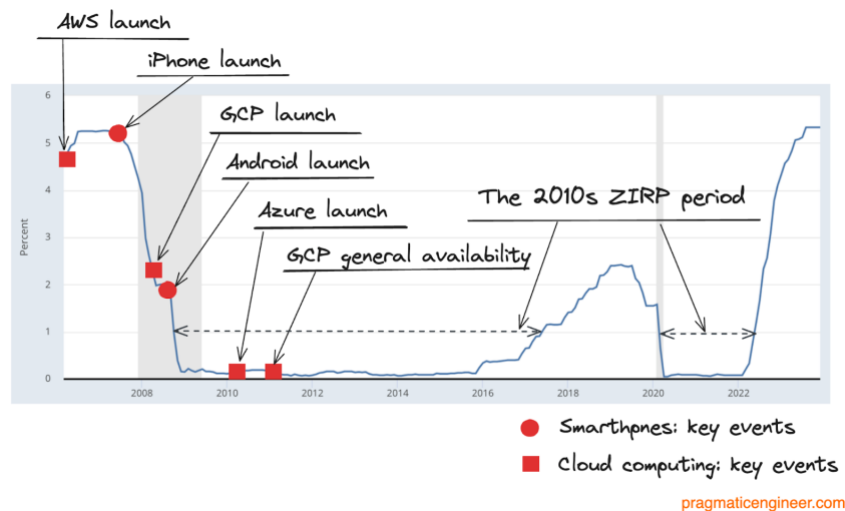


Ilustración 12. Eventos claves en la industria del software

**Nota.** Adaptado de *The State of the Engineering Market – 2024*, por Gergely Orosz, 2024, <https://newsletter.pragmaticengineer.com/p/state-of-eng-market-2024>

#### 4.1.1 - Expansión del software empresarial y consolidación del sector (2000-2010)

A inicios de la década del 2000, la industria del software experimentó un crecimiento sostenido, impulsado principalmente por la consolidación del software empresarial en grandes organizaciones. Las compañías comenzaron a invertir en soluciones de planificación de recursos empresariales (ERP), gestión de relaciones con clientes (CRM) y software de productividad, lo que generó una alta demanda de ingenieros especializados en la implementación y mantenimiento de estos sistemas.

Simultáneamente, la proliferación de Internet y la expansión de la Web 2.0 trajeron consigo nuevas oportunidades de negocio. Empresas como Google y Facebook surgieron en este período, promoviendo la necesidad de desarrolladores web y expertos en sistemas escalables. La adopción de software basado en la nube comenzó a ganar tracción en los últimos años de la década, sentando las bases para el crecimiento exponencial de SaaS (Software as a Service) en los años siguientes.

A pesar de la crisis financiera de 2008, el sector tecnológico demostró una notable resiliencia. Aunque se observó una ligera contracción en la contratación durante 2009, la digitalización de procesos empresariales y el auge de los servicios en línea permitieron que la industria del software continuará su expansión en la siguiente década.

#### 4.1.2 - Auge del desarrollo móvil y la transformación digital (2010-2020)

La siguiente década estuvo marcada por dos tendencias clave: la proliferación de aplicaciones móviles y la masificación de la computación en la nube. El lanzamiento del iPhone en 2007 y la posterior expansión del ecosistema Android impulsaron la demanda de desarrolladores de software especializados en aplicaciones móviles, un segmento que creció de forma exponencial en la década de 2010.

En paralelo, las empresas comenzaron a migrar sus infraestructuras a la nube, adoptando plataformas como Amazon Web Services (AWS), Microsoft Azure y Google Cloud. Este cambio estructural impulsó la demanda de ingenieros en infraestructura, DevOps y especialistas en seguridad en la nube. El

modelo SaaS se consolidó como el estándar de la industria, con gigantes como Salesforce y Dropbox liderando el segmento.

Durante esta década, la demanda de talento tecnológico alcanzó niveles récord. La expansión del ecosistema móvil, la digitalización de procesos en todos los sectores, y el crecimiento del modelo SaaS provocaron una aceleración en la contratación de desarrolladores e ingenieros. El auge de las startups tecnológicas, respaldadas por inversiones de capital de riesgo, incentivó una intensa competencia por atraer talento, con salarios en ascenso y paquetes de beneficios cada vez más atractivos.

Un factor macroeconómico clave que contribuyó a este auge fue el contexto de tasas de interés cercanas a cero, conocido como el periodo ZIRP (*Zero Interest Rate Policy*). Tras la crisis financiera de 2008, los bancos centrales mantuvieron políticas monetarias ultra expansivas para estimular la economía. Esto provocó un aumento significativo en la disponibilidad de capital de riesgo y una expansión agresiva del financiamiento a startups tecnológicas. En este entorno, las empresas podían contratar equipos de desarrollo sin una presión inmediata por generar ingresos, lo que facilitó una cultura de crecimiento acelerado (“growth at all costs”) y de abundancia de recursos en las áreas técnicas (Orosz, ZIRP software engineers, 2024). La combinación de innovación tecnológica, demanda corporativa y capital barato dio lugar a una década dorada para la industria del software.

#### 4.1.3 - Impacto de la pandemia y aceleración de la transformación digital (2020-2022)

El año 2020 trajo consigo un punto de inflexión en la industria del software debido a la pandemia de COVID-19. La crisis sanitaria global impulsó la adopción masiva de soluciones digitales, acelerando la transformación digital en sectores tradicionales como educación, salud y comercio minorista.

Las empresas tuvieron que adaptar rápidamente sus operaciones al trabajo remoto, lo que disparó la demanda de herramientas de colaboración como Zoom, Slack y Microsoft Teams. Al mismo tiempo, el comercio electrónico experimentó un crecimiento sin precedentes, con plataformas como Shopify y Amazon alcanzando volúmenes récord de ventas.

Este período estuvo caracterizado por una sobredemanda de talento tecnológico. Empresas de todos los sectores aumentaron sus inversiones en software para digitalizar sus operaciones, lo que generó una fuerte expansión en la contratación de desarrolladores. Sin embargo, esta tendencia no se mantendría indefinidamente.

#### 4.1.4 - Ajuste del mercado y transformación con IA generativa (2023 en adelante)

A partir de 2022 y con mayor fuerza en 2023, la industria tecnológica ingresó en una nueva etapa marcada por el reajuste macroeconómico y el fin de un ciclo prolongado de crecimiento sostenido. El detonante de este cambio fue el cierre del periodo ZIRP (*Zero Interest Rate Policy*), que había caracterizado a la economía global desde la crisis financiera de 2008. Ante un contexto inflacionario global, los principales bancos centrales, en particular la Reserva Federal de Estados Unidos, comenzaron a subir las tasas de interés de forma agresiva. El dinero dejó de ser barato, lo que modificó radicalmente las condiciones de financiamiento para startups y grandes empresas tecnológicas (Orosz, ZIRP software engineers, 2024).

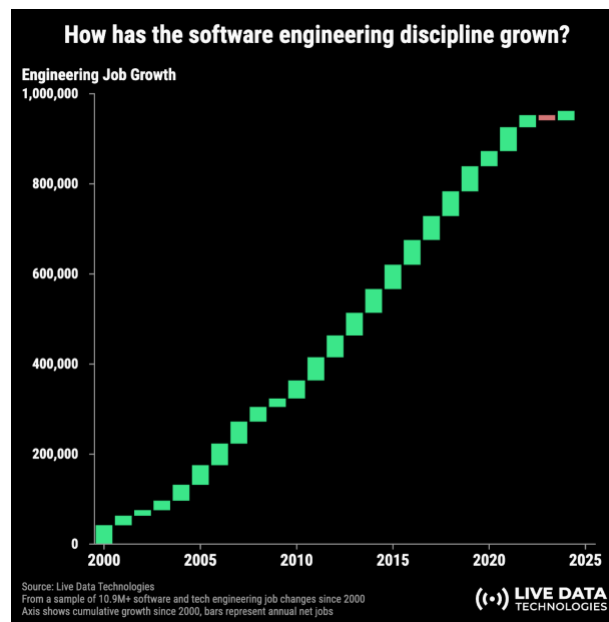
Este nuevo entorno puso fin a la lógica de “growth at all costs” que había dominado durante más de una década. Las startups se vieron obligadas a recortar costos y priorizar la rentabilidad por sobre el crecimiento acelerado, mientras que las grandes empresas tecnológicas iniciaron rondas masivas de despidos y reestructuraciones internas, ver Ilustración 13. Crecimiento de trabajos de ingeniería de software donde se evidencia una meseta en el crecimiento a partir de 2022. Se consolidó una narrativa centrada en la eficiencia operativa, el control de costos y la productividad. Esta transformación no solo

impactó las estructuras organizativas, sino también las expectativas sobre los equipos de ingeniería, que pasaron de ser motores de expansión a ser unidades bajo escrutinio por su capacidad de generar valor medible.

En este nuevo escenario emergió la inteligencia artificial generativa como una tecnología con el potencial de redefinir los modelos de trabajo en ingeniería de software. La irrupción de herramientas como GitHub Copilot (GitHub, n.d.), ChatGPT (OpenAI, 2022) y otras soluciones basadas en modelos de lenguaje generativo cambió rápidamente las dinámicas de desarrollo. Su adopción fue impulsada no solo por su capacidad para asistir en tareas de codificación, sino también por su promesa de mejorar la productividad individual y de los equipos técnicos.

Muchas organizaciones comenzaron a experimentar con estas herramientas como una forma de mitigar las presiones por hacer más con menos. La IA generativa se posicionó, así como un nuevo vector de eficiencia, alineado con la necesidad de racionalizar los costos en un contexto donde cada inversión tecnológica debe ser justificada con retornos concretos (Orosz, The Pragmatic Engineer, 2024).

Este cambio de paradigma aún está en evolución, pero representa una transición clara en la forma en que las empresas entienden el valor de la ingeniería de software y las habilidades requeridas en el nuevo ciclo económico. Lejos de la expansión desenfrenada del periodo ZIRP, la industria se orienta ahora hacia la optimización, la automatización y la adaptación ágil a un entorno de recursos más escasos y métricas más exigentes.



*Ilustración 13. Crecimiento de trabajos de ingeniería de software*

Evolución de contrataciones y despidos en empresas de tecnología (Pragmatic Engineer, 2024).

#### 4.1.5 - Industrias líderes y rezagadas en la transformación digital

Tras más de dos décadas de avances tecnológicos, la transformación digital no se ha desplegado de manera uniforme en todos los sectores económicos. Algunas industrias han liderado esta transición, impulsadas por la presión competitiva, la necesidad de eficiencia operativa o la evolución en las expectativas del cliente. Otras, en cambio, han mostrado una adopción más lenta, ya sea por barreras estructurales, regulatorias o culturales.

Según un estudio global de BCG (Boston Consulting Group, 2022), solo alrededor del 30% de las empresas logra una transformación digital exitosa que genere valor sostenible. Las industrias con mayor éxito incluyen servicios de TI, empresas basadas en internet, hardware y dispositivos, y plataformas de software. Estas organizaciones han demostrado una mayor madurez digital y capacidad para implementar tecnologías emergentes de manera efectiva.

De forma complementaria, Whatfix (Whatfix, 2023) identifica los sectores más impactados por la digitalización en términos de disrupción, adopción tecnológica y cambio organizacional.

Entre las industrias que han liderado la transformación digital se destacan:

- Servicios financieros: pionera en automatización de procesos, digitalización de servicios, open banking y uso de inteligencia artificial para personalización y gestión de riesgos.
- Retail y e-commerce: impulsada por la necesidad de mejorar la experiencia del cliente, gestionar grandes volúmenes de datos y operar de forma omnicanal.
- Tecnología y telecomunicaciones: actores centrales en el desarrollo y adopción de soluciones digitales, con foco en infraestructura, conectividad y escalabilidad.
- Manufactura avanzada: mediante la implementación de tecnologías como IoT, robótica, big data y sistemas ciber físicos en el marco de la llamada Industria 4.0.

Por el contrario, sectores como el de la salud, educación, construcción y sector público presentan mayores desafíos en su transformación. Si bien existen casos de innovación, en general estos sectores enfrentan dificultades como la fragmentación de sistemas, la resistencia al cambio, o entornos regulatorios complejos que limitan la velocidad de adopción tecnológica.

En síntesis, la transformación digital no avanza con la misma intensidad en todos los sectores. Identificar qué industrias están liderando este proceso, y cuáles requieren mayores esfuerzos, es clave para comprender las dinámicas de demanda de software en la actualidad y anticipar sus trayectorias futuras.

#### 4.2 - El papel de las empresas de servicios de tecnología

Las empresas de servicio de tecnología son empresas especializadas que ofrecen servicios de desarrollo, implementación y mantenimiento de soluciones tecnológicas a otras organizaciones. Estas consultoras permiten a las empresas acceder a conocimientos técnicos avanzados y recursos especializados sin la necesidad de desarrollar internamente estas capacidades, facilitando la adaptación a un entorno tecnológico en constante evolución.

A lo largo de las últimas décadas, las consultorías de software han desempeñado un rol esencial en la transformación digital de múltiples sectores, proporcionando soluciones personalizadas que optimizan procesos, mejoran la eficiencia operativa y potencian la competitividad de las empresas en el mercado global.

#### 4.2.1 - Modelos de consultoría de software: onshore, nearshore y offshore

Una de las formas más habituales de clasificar a las empresas de consultoría de software es según la ubicación geográfica de sus equipos en relación con sus clientes:

- Onshore: Consultoras que operan en el mismo país que la empresa contratante. Este modelo facilita la comunicación y la alienación cultural, aunque suele implicar costos más elevados debido a los salarios locales.
- Nearshore: Consultoras ubicadas en países cercanos, generalmente en la misma zona horaria o con diferencias mínimas. Este enfoque ofrece un equilibrio entre costos y eficiencia operativa, permitiendo una colaboración más fluida.
- Offshore: Consultoras situadas en regiones más distantes con costos laborales más bajos. Aunque este modelo puede ofrecer reducciones significativas en costos, puede presentar desafíos en la gestión debido a diferencias horarias y culturales.

Cada uno de estos modelos presenta ventajas y desafíos específicos, y la elección entre ellos depende de factores como el presupuesto, la complejidad del proyecto y la necesidad de interacción directa con el equipo de desarrollo.

#### 4.2.2 - Tamaño actual de mercado de las empresas de servicio de tecnología

El mercado global de servicios de tecnología y consultoría de software ha mostrado un crecimiento significativo en los últimos años. Según datos de Business Research Insights (Business Research Insights, 2024), el tamaño del mercado global de servicios de TI se estimó en 1.125,66 mil millones de dólares en 2023 y se proyecta que alcance 1.617,03 mil millones de dólares para 2032, con una tasa de crecimiento anual compuesta (CAGR) del 4,10 % durante este período. Este sector abarca una amplia gama de servicios, desde desarrollo de software hasta consultoría en transformación digital, gestión de infraestructuras tecnológicas y soluciones de ciberseguridad.

Las empresas de servicios de tecnología desempeñan un papel crucial en la digitalización de grandes corporaciones en diversos sectores. Empresas como The Walt Disney Company, McDonald's y JP Morgan Chase han trabajado con firmas tecnológicas como Globant, Capgemini y Accenture en diversas iniciativas de mejora digital. Estos ejemplos reflejan cómo las empresas de tecnología no solo desarrollan software, sino que también brindan asesoría estratégica y soporte técnico a algunas de las corporaciones más grandes del mundo.

#### 4.2.3 - Principales empresas de servicios de tecnología: tamaño e ingresos anuales

A continuación, se presentan algunas de las principales consultoras de software a nivel mundial, destacando su tamaño e ingresos anuales:

- Accenture (Accenture, 2024): Multinacional de consultoría estratégica, servicios tecnológicos y externalización, con más de 700.000 empleados en 120 países. En el año fiscal 2022, reportó ingresos de 61.600 millones de dólares.
- Globant (Globant, 2024): Empresa argentina especializada en desarrollo de software y servicios de TI, con presencia en más de 25 países y aproximadamente 27.000 empleados. En 2022, alcanzó ingresos de 1.780 millones de dólares.
- Perficient: Consultora digital estadounidense que ofrece servicios de tecnología y estrategia, con más de 6.500 empleados. En 2023, reportó ingresos por 906 millones de dólares (Stock Analysis, 2024)).

- Thoughtworks (Thoughtworks, 2023): Empresa global de consultoría de software que se especializa en la integración de tecnología y diseño, con más de 12.500 empleados en 18 países. En 2022, reportó ingresos de 1.296 millones de dólares.
- GlobalLogic: Proveedor de servicios de ingeniería de productos digitales, con más de 25.000 empleados y presencia en 14 países. En 2021, fue adquirida por Hitachi en una transacción que valoró a la empresa en 9.600 millones de dólares (Hitachi, 2021).

Estas empresas ejemplifican la diversidad y el alcance global de las consultoras de software, atendiendo a una amplia gama de industrias y necesidades tecnológicas.

#### 4.2.4 - Desafíos actuales de las empresas de servicios de tecnología

El contexto actual plantea una serie de desafíos estructurales y estratégicos para las empresas de servicios de tecnología, derivados tanto del nuevo entorno macroeconómico como de los avances tecnológicos recientes.

El fin del periodo ZIRP (Zero Interest Rate Policy), que había facilitado la expansión de las startups tecnológicas mediante financiamiento abundante y barato, marcó el inicio de un nuevo ciclo. A partir de 2022, con la suba de tasas de interés por parte de los bancos centrales, el capital de riesgo se volvió más escaso y exigente. Las empresas tecnológicas, clientes tradicionales de las consultoras, comenzaron a priorizar la eficiencia operativa y la rentabilidad por sobre el crecimiento acelerado. Este giro impactó directamente en los presupuestos destinados a servicios externos, generando una presión creciente sobre los márgenes de las consultoras.

En paralelo, la irrupción de herramientas de inteligencia artificial generativa, como ChatGPT (OpenAI, 2022), GitHub Copilot (GitHub, n.d.), y otras soluciones basadas en modelos fundacionales, ha introducido una disrupción en las dinámicas tradicionales del desarrollo de software. Muchas tareas que antes requerían múltiples horas de trabajo humano pueden ahora resolverse en minutos o parcialmente automatizarse. Esto genera tensiones dentro del modelo de negocio tradicional de las consultoras, que históricamente ha estado basado en la venta de tiempo/hombre y la escala de equipos. Surge así la pregunta: ¿cómo capturar valor cuando parte del trabajo puede ser reemplazado o acelerado por una IA?

Estas transformaciones también imponen una nueva narrativa dominante en el sector: la de la productividad. Ya no alcanza con entregar software funcional. Los clientes exigen resultados medibles, indicadores de eficiencia, y un retorno claro sobre su inversión. Esto obliga a las consultoras a repensar sus servicios, incorporando herramientas de IA en su operación interna, automatizando tareas repetitivas, y diseñando soluciones orientadas a impacto más que a esfuerzo. Al mismo tiempo, deben hacerlo sin erosionar sus propios márgenes, lo que genera un equilibrio difícil entre innovación, eficiencia y rentabilidad.

Frente a este escenario, las consultoras enfrentan dilemas estratégicos profundos:

- ¿Cómo adoptar tecnologías que aumenten la eficiencia sin reducir el valor percibido de sus servicios?
- ¿Cómo justificar tarifas similares cuando las herramientas pueden acelerar entregables?
- ¿Cómo construir relaciones de largo plazo con clientes que privilegian el corto plazo y la optimización de costos?

Estas tensiones configuran un terreno desafiante, pero también fértil para la innovación. Las empresas que logren adaptarse, repensar su propuesta de valor, y posicionarse como aliadas estratégicas en la

adopción inteligente de nuevas tecnologías, tendrán mayores chances de sostener su relevancia en esta nueva etapa de la industria.

### 4.3 - Caso Rootstrap

Rootstrap (Rootstrap, n.d.) es una empresa de servicios de tecnología y consultoría de software con más de 14 años de trayectoria. Desde sus inicios, ha tenido como propósito brindar soluciones digitales de extremo a extremo, combinando ingeniería, diseño y estrategia para resolver desafíos tecnológicos y de negocio.

A lo largo de su desarrollo, la empresa ha consolidado una estructura internacional que le permite operar con agilidad, adaptarse a distintos mercados y establecer relaciones de largo plazo con clientes de diversos sectores.

#### 4.3.1 - Categoría de empresa

Rootstrap se posiciona dentro de la categoría de empresas nearshore, dado que su principal mercado es Estados Unidos y la mayoría de su equipo se encuentra en América Latina, lo que permite operar con cercanía horaria, afinidad cultural y costos competitivos.

Sin embargo, también cuenta con presencia estratégica en el mercado estadounidense, con roles clave como el CEO, VP de Growth y parte del equipo comercial ubicados en Los Ángeles y Nueva York. Esta combinación le permite mantener una propuesta de valor cercana y flexible para sus clientes, integrando la operación latinoamericana con un contacto directo en el país de destino.

#### 4.3.2 - Clientes

La empresa mantiene una base de aproximadamente 30 clientes activos, mayoritariamente en Estados Unidos, aunque también brinda servicios en menor escala a organizaciones en Latinoamérica y Europa.

Entre sus clientes se destacan nombres como MasterClass (Rootstrap, 2021), Emeritus (Rootstrap, 2022) y Google (Rootstrap, 2019), con quienes colabora en proyectos de alto impacto tecnológico. Rootstrap trabaja tanto con empresas que poseen equipos técnicos internos y requieren escalamiento de talento especializado, como con compañías que buscan un socio estratégico para abordar desafíos de negocio mediante el uso de tecnología.

Este enfoque centrado en el cliente se refleja en los resultados: en 2024, Rootstrap alcanzó un Net Promoter Score (NPS) del entorno del 85%, un indicador que evidencia el compromiso de la empresa con un nivel de servicio alto, consistente y orientado a la calidad.

#### 4.3.3 - Equipo y cultura

Rootstrap cuenta con un equipo de aproximadamente 200 personas, distribuidas entre Uruguay, Argentina, Colombia y Estados Unidos. Su cultura organizacional se caracteriza por un enfoque en la colaboración, la mejora continua y la búsqueda de impacto en los proyectos que desarrolla para sus clientes. Como muchas empresas de servicios tecnológicos, ha transitado diversas etapas de crecimiento y maduración, lo que también se ha reflejado en la composición y perfil de su equipo técnico.

Más allá de su estructura técnica, la empresa se caracteriza por una cultura organizacional centrada en las personas, con un fuerte énfasis en la cercanía, la colaboración y el crecimiento profesional. Este enfoque se refleja en indicadores organizacionales sólidos: en los últimos años, la tasa de salidas

voluntarias (Employee Voluntary Turnover) se ha mantenido por debajo del 15%, y la duración promedio en la empresa (Employee Tenure) supera los 4 años, lo que evidencia un entorno donde las personas desarrollan su carrera a largo plazo.

En los últimos años, a raíz del enfriamiento del mercado, Rootstrap ha tenido un ritmo de contratación más moderado. Este fenómeno, sumado a la antigüedad promedio del equipo, ha dado lugar a una composición marcadamente senior: aproximadamente el 60% del equipo técnico está conformado por perfiles senior, el 30% por semi seniors, y solo un 10% por juniors.

En cuanto a la formación, si bien una parte significativa del equipo cuenta con títulos universitarios en carreras vinculadas a la ingeniería o a la informática, Rootstrap también valora perfiles provenientes de trayectorias no tradicionales. La empresa ha contratado personas que se formaron de manera autodidacta o que realizaron programas intensivos como bootcamps, priorizando las competencias actuales y potencial de desarrollo por sobre la educación formal. Incluso ha organizado bootcamps propios, con el objetivo de capacitar y luego incorporar nuevos talentos.

Durante el período de crecimiento acelerado de la industria, especialmente entre 2020 y 2022, fue común ver cómo personas provenientes de otras profesiones, como diseño gráfico, administración e incluso contabilidad, comenzaron a desarrollarse en tecnología y dar sus primeros pasos como programadores. Este fenómeno también estuvo presente en Rootstrap y permitió una mayor diversidad de backgrounds dentro del equipo, enriqueciendo la dinámica y la perspectiva de los equipos de desarrollo. Sin embargo, esta apertura respondió en gran medida a un contexto de alta demanda y fuerte expansión, que obligó a las empresas a buscar talento más allá de los perfiles tradicionales. Con el enfriamiento del mercado observado a partir de 2023, ha resurgido una mayor exigencia en términos de formación técnica formal o experiencia comprobada en desarrollo, volviendo a priorizarse en muchos casos a perfiles con estudios universitarios en ingeniería o carreras afines, o con un historial sólido en proyectos de software. Esta transición evidencia que aquella diversidad de trayectorias fue, en parte, consecuencia de un pico excepcional en la necesidad de desarrolladores, más que una tendencia consolidada a largo plazo.

#### 4.3.4 - Posicionamiento

Las empresas de servicios de tecnología suelen organizar su propuesta de valor en torno a diferentes líneas de negocio, en función del tipo de solución que brindan a sus clientes. Entre las modalidades más comunes se encuentran el desarrollo de productos end-to-end, también conocido como full product development, y la modalidad de staff augmentation, o aumento de capacidad técnica.

En el modelo de full product development, la consultora asume la responsabilidad integral del diseño, desarrollo y evolución de un producto digital, trabajando desde etapas tempranas de definición hasta su lanzamiento y posterior mantenimiento. En cambio, en la modalidad de staff augmentation, la empresa cliente mantiene el liderazgo del producto y contrata a la consultora para sumar desarrolladores o equipos técnicos específicos a sus propios equipos, integrándose temporalmente para escalar capacidades internas.

Rootstrap opera en ambas modalidades, con aproximadamente un 50% de sus clientes en cada una. Esta diversificación le permite adaptarse a distintas necesidades del mercado, desde startups en fases iniciales que requieren apoyo estratégico y técnico para construir un producto desde cero, hasta compañías en etapas más avanzadas que buscan reforzar sus equipos internos con talento especializado.

La empresa trabaja con clientes en distintos segmentos de madurez, desde emprendimientos en etapa *pre-seed* hasta compañías con rondas de inversión en Series A, B, C y D, o incluso Enterprises que no

dependen de capital externo. Estas etapas se diferencian por el grado de desarrollo del producto, la validación del modelo de negocio y el capital disponible, siendo las rondas más avanzadas aquellas que exigen mayor escalabilidad y solidez operativa (Finmark, 2023). En este sentido, el posicionamiento y la apuesta principal de Rootstrap se orienta hacia organizaciones con mayor grado de financiación o tracción en sus mercados, dado el tipo de desafíos técnicos y organizativos que suelen presentar.

Históricamente, Rootstrap ha construido su posicionamiento en el desarrollo personalizado de productos digitales (custom product development) tanto web como mobile, así como en la provisión de talento técnico para equipos existentes. Su propuesta de valor combina capacidades técnicas, metodologías ágiles, y experiencia en diseño centrado en el usuario, lo cual le ha permitido operar en mercados exigentes como Estados Unidos y Europa.

En los últimos años, la empresa ha comenzado un proceso de redefinición estratégica, con el objetivo de posicionarse como un socio clave en el desarrollo de soluciones impulsadas por inteligencia artificial y datos. Este nuevo enfoque, basado en el concepto AI and Data Driven, busca acompañar a empresas que desean incorporar IA en sus operaciones o modelos de negocio, desarrollar nuevas aplicaciones que integren capacidades de IA, o escalar sus equipos técnicos con perfiles especializados en esta área. Este reposicionamiento busca anticiparse a las transformaciones del mercado, capturar nuevas oportunidades de valor, y mantenerse competitivo en un entorno tecnológico cada vez más dinámico.

Más información sobre la empresa puede encontrarse en su sitio web oficial: <https://www.rootstrap.com>

# INVESTIGACION EMPIRICA-DOCUMENTAL

## CAPÍTULO 5 - METODOLOGÍA DE LA INVESTIGACIÓN

### 5.1 Enfoque metodológico

El presente estudio adopta un enfoque descriptivo y exploratorio, con un diseño no experimental, adecuado para analizar fenómenos en contextos reales sin manipular variables. Se emplea una estrategia metodológica cualitativa que combina entrevistas semiestructuradas con líderes del sector y el análisis de fuentes secundarias relevantes. Esta triangulación metodológica de técnicas y fuentes permite obtener una visión más completa y contrastada del fenómeno bajo estudio: el impacto de la inteligencia artificial (IA) en el desarrollo de software, la evolución del rol del desarrollador, la demanda de talento y las implicancias estratégicas para las empresas de consultoría tecnológica.

La elección de una estrategia cualitativa se justifica por la naturaleza emergente del fenómeno, caracterizado por una evolución acelerada y un alto grado de incertidumbre. Las entrevistas permiten captar percepciones, experiencias y expectativas que no siempre están reflejadas en los datos estadísticos tradicionales. A su vez, el análisis de fuentes secundarias enriquece esta visión, aportando contexto macroeconómico, proyecciones de empleo, tendencias de mercado y opiniones de expertos reconocidos.

### 5.2 Técnicas de recolección de datos

#### 5.2.1 Datos primarios

Se realizaron entrevistas semiestructuradas a líderes tecnológicos con experiencia en dirección de equipos de desarrollo de software. La técnica de entrevista fue seleccionada por su capacidad para explorar en profundidad las percepciones y experiencias de los participantes, permitiendo adaptar el diálogo según los temas emergentes en cada conversación.

Los entrevistados fueron seleccionados priorizando perfiles con roles de liderazgo (como CTOs, VPs de Ingeniería o Directores de Ingeniería) y al menos 10 años de experiencia en la industria tecnológica. Todas las entrevistas fueron realizadas de manera virtual, tuvieron una duración de entre 30 y 45 minutos, y fueron registradas en audio con consentimiento previo, para su posterior transcripción y análisis.

El guion de preguntas utilizado en las entrevistas fue estructurado en seis grandes ejes temáticos, los cuales guiaron la conversación en función del perfil de cada entrevistado:

1. Transformaciones provocadas por la inteligencia artificial en la industria tecnológica y sus distintos roles.
2. Cambios esperados en el rol del desarrollador de software.
3. Competencias clave para los desarrolladores en el nuevo contexto impulsado por IA.
4. Impacto de la IA en la demanda de servicios de desarrollo y talento técnico.
5. Estrategias para empresas de servicios de tecnología ante el nuevo paradigma.
6. Reflexiones adicionales y aspectos emergentes identificados durante la entrevista.

El detalle completo del guión de entrevista se incluye como Anexo 1 – Guía de preguntas para entrevistas.

El análisis de las entrevistas se realizó mediante codificación temática, identificando categorías emergentes y patrones comunes entre los discursos. Esta técnica permitió organizar e interpretar la información cualitativa obtenida, en línea con los objetivos del estudio.

### 5.2.2 Datos secundarios

Adicionalmente, se analizaron fuentes de datos secundarias provenientes de estudios publicados por organismos e instituciones del sector tecnológico, publicaciones académicas y reportes de la industria. Esta información se organizó y presentará en el CAPÍTULO 7 - DATOS SECUNDARIOS, que incluirá:

- Proyecciones de empleo en el sector tecnológico.
- Proyecciones de crecimiento esperado del mercado global de software.
- Opiniones y análisis de referentes de la industria y expertos sobre la evolución del rol del desarrollador y la demanda de talento.

Este abordaje permite contextualizar los hallazgos cualitativos en un marco más amplio y robusto.

## CAPÍTULO 6 - ENTREVISTAS A LÍDERES DE TECNOLOGÍA

Con el objetivo de comprender cómo la inteligencia artificial está transformando el desarrollo de software desde una perspectiva práctica, esta sección recopila las visiones de líderes de la industria tecnológica que hoy ocupan roles estratégicos principalmente en empresas de consultoría y desarrollo, y también en otro tipo de empresas donde el software es un elemento fundamental. A través de entrevistas en profundidad, se exploran sus experiencias actuales, percepciones sobre el impacto de las nuevas tecnologías y proyecciones a futuro. Sus aportes permiten identificar tendencias emergentes en cuanto al perfil del desarrollador, las competencias que ganan relevancia, los cambios en la dinámica de los equipos técnicos, y las nuevas expectativas que se abren tanto desde el lado de la oferta como de la demanda de talento.

### 6.1 - Resumen de entrevistas

Camila Moscatelli - Directora de Ingeniería en Rootstrap



Camila Moscatelli, actual Directora de Ingeniería en Rootstrap (Rootstrap, n.d.), cuenta con una trayectoria de más de 10 años en la industria del software, evolucionando desde roles técnicos hasta posiciones estratégicas y de liderazgo. Desde su perspectiva, la inteligencia artificial (IA) está generando un cambio profundo en las expectativas relacionadas con la agilidad y velocidad del desarrollo de productos digitales. Destaca que, aunque actualmente estas expectativas aún no se han materializado por completo en la práctica, se observa una presión creciente para reducir tiempos de desarrollo y enfocarse más en la personalización y satisfacción del usuario final.

Moscatelli considera que la IA ya está impactando notablemente en etapas específicas del proceso de desarrollo, principalmente en las fases de programación y testing. Explica que tareas que anteriormente requerían una cantidad significativa de tiempo, como la creación de código básico o repetitivo, están siendo cada vez más automatizadas. Esto desplaza a los desarrolladores hacia un rol más estratégico y de supervisión, donde adquiere mayor relevancia la capacidad de entender profundamente los objetivos del negocio y garantizar que la solución técnica responda adecuadamente a esas necesidades.

En cuanto al futuro del rol del desarrollador, Moscatelli prevé un incremento en las expectativas de habilidades tradicionalmente asociadas a perfiles más senior o arquitectos, como el análisis crítico del código, la visión estratégica del producto y la capacidad de orquestar soluciones técnicas complejas. Considera que, aunque en el corto plazo no habrá un cambio drástico en la cantidad de desarrolladores necesarios, sí existirá una creciente demanda de perfiles más especializados en seguridad, análisis de datos y gestión estratégica del software. En contraste, anticipa que la necesidad de roles más operativos o juniors podría reducirse, dado que muchas tareas introductorias o repetitivas serán cubiertas directamente por herramientas de IA.

En relación con la productividad y adopción de herramientas basadas en inteligencia artificial, como los copilotos de código (Copilot, Cursor, Windsurf), Camila observa actualmente una adopción moderada debido principalmente a resistencias culturales y cierto escepticismo inicial en la industria. Sin embargo, predice un crecimiento significativo en la adopción de estas herramientas en los próximos años, lo que se traducirá en incrementos tangibles de productividad y una redefinición sustancial de las expectativas de los clientes respecto a tiempos y costos de desarrollo.

Finalmente, Moscatelli identifica que estos cambios impulsados por la IA generarán en el corto plazo un aumento en la demanda global de desarrollo de software, a medida que diversas industrias busquen aprovechar estas tecnologías para mejorar su eficiencia operativa y sus propuestas de valor. A largo plazo, sin embargo, no descarta la posibilidad de una reducción en la cantidad total de desarrolladores, dado el avance continuo de herramientas inteligentes. En términos estratégicos para empresas consultoras, recomienda fuertemente invertir en capacitación interna, fomentar una cultura organizacional ágil, y reforzar áreas clave como seguridad, análisis de datos e infraestructura tecnológica, aspectos fundamentales para enfrentar el futuro con éxito en un mercado cada vez más competitivo y dependiente de la inteligencia artificial.

Juan Pablo Mazza - CTO y Fundador de Effectus Software



Juan Pablo Mazza, fundador y actual CTO de Effectus Software (Effectus Software, n.d.), posee una amplia trayectoria en el desarrollo de software, con foco en tecnologías mobile y en la gestión estratégica de equipos tecnológicos orientados al mercado estadounidense. Desde su posición de liderazgo en una agencia de diseño y desarrollo con más de 40 personas, Mazza observa que la inteligencia artificial (IA) está generando un fuerte impacto, especialmente en tareas como testing automatizado, refactoring, documentación y procesos iniciales de desarrollo. Según su experiencia, estas herramientas permiten reducir significativamente las interacciones humanas en etapas creativas y técnicas del desarrollo, acelerando la validación y prototipado de productos digitales.

Mazza identifica al rol del tester como uno de los más afectados por la IA, debido a la alta automatización que pueden alcanzar sus tareas. No obstante, considera que los testers pueden reorientar sus habilidades hacia actividades de mayor valor estratégico. Por otro lado, respecto al rol del desarrollador, Mazza señala que, aunque herramientas como los copilotos de código ofrecen importantes ventajas en eficiencia, también traen consigo riesgos como la creciente "caja negra", que puede dificultar la resolución de problemas complejos cuando estos sistemas automáticos fallan. Por esta razón, enfatiza que los desarrolladores expertos siguen siendo fundamentales para agregar valor técnico profundo y crítico.

Al analizar la evolución futura del rol del desarrollador, Mazza prevé un perfil más integral y adaptativo, destacando la importancia de habilidades blandas como la comunicación efectiva, la flexibilidad y una comprensión sólida de los objetivos de negocio. Visualiza que, en los próximos años, los desarrolladores pasarán menos tiempo resolviendo problemas técnicos puros y más tiempos involucrados activamente en la estrategia de producto, ofreciendo soluciones alineadas a las necesidades del cliente. Además, coincide con la visión de un "desarrollador-orquestador", donde la habilidad clave radica en gestionar eficazmente múltiples herramientas tecnológicas para maximizar la productividad.

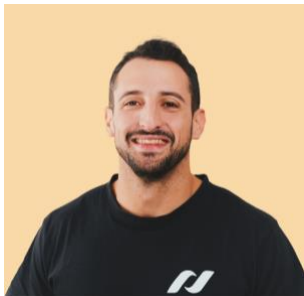
En cuanto al impacto específico de los asistentes de código (como Cursor), Mazza destaca que estas herramientas ya están cambiando significativamente la forma en que trabajan los desarrolladores. Relata cómo el tiempo de programación tradicional está migrando hacia la optimización y ajuste de prompts para inteligencia artificial, lo que considera una revolución de la productividad con potencial crecimiento exponencial. Mazza señala que este incremento de productividad aún no tiene un límite claramente definido, y predice que continuará transformando radicalmente la dinámica laboral en el sector.

Respecto al futuro del mercado laboral en la industria del software, Mazza proyecta que a corto y mediano plazo habrá un aumento inicial de demanda debido al entusiasmo por incorporar inteligencia

artificial en diversas soluciones industriales. Sin embargo, sostiene que, a largo plazo, la demanda total de desarrolladores podría disminuir debido al crecimiento exponencial de la productividad y la automatización de tareas. Según su perspectiva, los roles más demandados serán aquellos que integren habilidades técnicas con fuertes competencias en adaptabilidad y habilidades blandas, anticipando una reducción gradual en la cantidad total de desarrolladores necesarios.

Finalmente, desde una visión estratégica para empresas de consultoría tecnológica, Mazza recomienda claramente especializarse en nichos específicos, orientarse hacia la resolución de problemas industriales concretos y potenciar la comunicación y el marketing en torno a la inteligencia artificial. Subraya que la capacidad de ofrecer servicios relacionados con IA está dejando de ser un diferenciador y se está convirtiendo rápidamente en una expectativa básica por parte de los clientes. Asimismo, sugiere impulsar activamente la adopción y capacitación interna en nuevas herramientas, considerando indispensable fomentar una cultura organizacional de constante aprendizaje y adaptabilidad tecnológica.

Sebastián Melgar – CEO y Fundador de Neo Coast



Sebastián Melgar, CEO y fundador de Neocoast (Neocoast, n.d.), empresa dedicada a ofrecer servicios integrales de software principalmente al mercado estadounidense, presenta una visión clara y crítica sobre la influencia de la inteligencia artificial en el sector. Según Melgar, la IA ya forma parte central en todas las conversaciones sobre nuevos proyectos digitales, habiendo desplazado en importancia al desarrollo mobile, que fue tendencia predominante años atrás. Destaca que actualmente la mayoría de los clientes busca integrar IA en sus productos, ya sea por eficiencia operativa o para incrementar su atractivo ante inversores, aunque reconoce que no todos entienden claramente cómo aplicarla

estratégicamente.

Melgar enfatiza que, en el presente inmediato, la inteligencia artificial no está afectando negativamente a ningún rol específico en la industria del desarrollo de software; al contrario, observa un impacto positivo generalizado en eficiencia y productividad para desarrolladores, diseñadores, QA y project managers. Sin embargo, identifica una amenaza latente en el mediano plazo: la creciente capacidad de la IA para automatizar tareas técnicas podría reducir considerablemente la cantidad de recursos humanos necesarios por proyecto, generando presión a la baja sobre tarifas y salarios, y eventualmente modificando el panorama laboral del sector.

Con respecto al futuro del rol del desarrollador, Melgar prevé cambios importantes, especialmente en términos de competencias técnicas requeridas. A corto plazo (uno o dos años), anticipa que dominar herramientas de inteligencia artificial será un requisito excluyente para los desarrolladores, equiparable al dominio del idioma inglés. Mirando hacia el mediano plazo (tres años), sostiene que los desarrolladores deberán adoptar un enfoque más generalista, trabajando con múltiples tecnologías y lenguajes en simultáneo, asistidos significativamente por herramientas de IA. Además, visualiza un desplazamiento del trabajo hacia sistemas heredados o legacy en grandes organizaciones, anticipando que las tareas más rutinarias y sencillas serán completamente automatizadas mediante IA.

Melgar también señala que las habilidades clave para los desarrolladores del futuro próximo serán la capacidad de comunicarse efectivamente con herramientas de inteligencia artificial (prompt engineering), junto con sólidas habilidades blandas, como la adaptabilidad, comunicación y velocidad de respuesta. Subraya la necesidad de reinventar los sistemas educativos tradicionales en informática e ingeniería, cuestionando la utilidad de extensos programas académicos que podrían quedar desactualizados ante los rápidos cambios tecnológicos impulsados por la IA.

En términos del mercado general del software, Sebastián mantiene una perspectiva optimista moderada. A corto plazo, identifica oportunidades en la creciente necesidad de rehacer productos digitales generados con IA de forma rápida y superficial (MVPs), pero cuya calidad técnica podría ser deficiente. Sin embargo, a largo plazo prevé una reducción considerable del negocio de desarrollo de MVPs y productos sencillos, reemplazados por soluciones automatizadas robustas, y anticipa que la industria deberá enfocarse en proyectos más complejos y sistemas heredados.

Finalmente, desde un punto de vista estratégico para empresas consultoras, Melgar recomienda fervientemente mantener una fuerte capacidad de adaptación y aprendizaje continuo. Advierte contra la resistencia a adoptar nuevas tecnologías, señalando que esto podría volverse contraproducente rápidamente. Sugiere enfocarse en servicios especializados de alto valor agregado y aprovechar activamente el networking y el intercambio constante con líderes y expertos de la industria para anticipar y adaptarse a los constantes cambios tecnológicos que caracterizan al sector del software en esta nueva era impulsada por la inteligencia artificial.

Santiago Perez – Director de Ingeniería en Rootstrap



Santiago Perez es Director de Ingeniería en Rootstrap (Rootstrap, n.d.), liderando equipos técnicos con amplia experiencia en desarrollo, gestión y estrategia tecnológica. Desde su perspectiva, la inteligencia artificial ya está generando cambios profundos en todas las etapas del proceso de desarrollo de software, desde la definición inicial hasta el despliegue final, siendo especialmente notables los impactos en la generación automatizada de código, diseño y testing. Para Perez, estas herramientas no reemplazan por completo las funciones humanas, sino que incrementan significativamente la eficiencia en tareas repetitivas, mientras requieren aún de supervisión experta para resultados óptimos.

En relación con el impacto específico por roles, Perez identifica el área de Quality Assurance (QA) como una de las más susceptibles de transformación debido al potencial de automatización masiva del testing, aunque recalca que siempre será necesario un componente humano que aporte criterio y supervisión estratégica. En el caso de los desarrolladores, se observa que la inteligencia artificial está optimizando tareas rutinarias o de alcance limitado, permitiendo a los profesionales centrarse en actividades más estratégicas y complejas, lo que incrementa la calidad general del trabajo entregado.

Sobre la evolución del rol del desarrollador en los próximos años, Perez anticipa que la habilidad para utilizar herramientas basadas en IA será cada vez más determinante y esencial. Pronostica que ciertas tareas técnicas, como el testing unitario o el desarrollo inicial de interfaces, serán casi totalmente delegadas a sistemas inteligentes. En contrapartida, destaca la importancia creciente de las competencias relacionadas con la gestión estratégica de proyectos, supervisión técnica profunda y la capacidad de tomar decisiones críticas sobre resultados generados automáticamente, reforzando el valor del juicio humano experto.

En lo que respecta a herramientas específicas, como los IDEs asistidos por inteligencia artificial, Perez señala que su adopción ya es una realidad consolidada en Rootstrap, donde experimentos internos demuestran incrementos sustanciales de productividad. Considera que estos aumentos variarán según el contexto específico de cada proyecto, siendo más evidentes en trabajos repetitivos o claramente definidos y menos pronunciados en tareas complejas que requieren decisiones continuas. Aunque no se atreve a precisar cuánto podría incrementarse la productividad en el largo plazo, prevé un crecimiento sostenido conforme estas tecnologías mejoran en velocidad y precisión.

A nivel de competencias clave para los desarrolladores, Perez enfatiza el dominio técnico avanzado en el uso efectivo de herramientas de IA como requisito básico, pero destaca especialmente la relevancia creciente de habilidades blandas como comunicación efectiva, liderazgo y relaciones interpersonales, que se convertirán en factores diferenciadores importantes en el mercado laboral.

Perez se mostró cauteloso respecto a la evolución futura del mercado general del software, indicando incertidumbre acerca de si la demanda global crecerá o se mantendrá estable. No obstante, visualiza claramente un cambio significativo en la naturaleza interna de la demanda: espera una reconversión de roles hacia áreas técnicas más especializadas, capaces de manejar herramientas avanzadas y análisis de grandes volúmenes de datos. En términos netos, considera que la cantidad total de profesionales técnicos probablemente se mantendrá estable, aunque con un desplazamiento hacia nuevos tipos de habilidades y conocimientos técnicos más orientados a la gestión estratégica y el análisis avanzado.

Finalmente, desde su posición de liderazgo en una empresa de consultoría tecnológica, Perez recomienda enfáticamente cultivar un talento humano flexible y abierto al cambio, acompañado de una fuerte cultura organizacional orientada a la experimentación constante con nuevas herramientas y tecnologías. Subraya que este enfoque experimental no solo permite identificar rápidamente qué estrategias tecnológicas funcionan mejor, sino que también crea ventajas competitivas a largo plazo en un entorno tecnológico dinámico y en constante evolución.

Cecilia Mariani – CEO de Darwoft



Cecilia Mariani, actual CEO de Darwoft (Darwoft, n.d.) y con extensa trayectoria en empresas tecnológicas multinacionales como Motorola e Intel, comparte una visión pragmática sobre cómo la inteligencia artificial (IA) está afectando el proceso de desarrollo de software. Según Mariani, una de las primeras observaciones es que los desarrolladores más jóvenes están adoptando rápidamente herramientas de inteligencia artificial para acelerar la generación de código, mientras que los perfiles senior han mostrado una adopción más lenta debido a hábitos y prácticas establecidas durante años de experiencia previa. Aunque aún no tiene métricas precisas sobre aumentos significativos en productividad, Mariani señala que sus clientes ya comienzan a exigir la incorporación de estas herramientas en los procesos de desarrollo, principalmente con la expectativa de reducir tiempos y costos.

En cuanto a las áreas específicas del proceso de desarrollo impactadas por la IA, Cecilia identifica claramente un cambio notable en la experiencia de usuario (UX) y en las interfaces conversacionales, previendo transformaciones sustanciales en cómo se diseñan y se experimentan los productos digitales. Además, resalta un impacto importante en el desarrollo de código, particularmente en la reutilización de componentes y en la automatización del testing, que está simplificando considerablemente tareas repetitivas y que históricamente requerían gran cantidad de tiempo y recursos.

Respecto al futuro del rol del desarrollador, Mariani anticipa una evolución significativa en términos de habilidades requeridas. A corto y mediano plazo, prevé que habrá una reducción progresiva en la necesidad de perfiles junior destinados a tareas operativas o repetitivas, dado que estas labores podrán ser realizadas con mayor eficiencia por herramientas inteligentes. Esto derivará en equipos de desarrollo más pequeños y especializados, donde los roles clave serán aquellos capaces de realizar tareas estratégicas relacionadas con la arquitectura, el diseño lógico y el profundo entendimiento del negocio.

Mariani hace énfasis especial en que, si bien muchas tareas técnicas básicas pueden ser asumidas por IA, la importancia de las habilidades blandas y la capacidad analítica aumentará considerablemente. Entre las competencias esenciales que identifica están la lógica, el pensamiento crítico, el entendimiento profundo del negocio, y habilidades interpersonales como la comunicación efectiva y la resolución de conflictos. De hecho, destaca que actualmente en Darwoft se están priorizando capacitaciones en liderazgo, oratoria y pensamiento crítico por encima del entrenamiento puramente técnico, anticipando que estas serán las habilidades diferenciadoras del futuro.

En relación con la demanda global de software y el mercado tecnológico en general, Mariani es optimista sobre el crecimiento del sector, especialmente en áreas relacionadas con ingeniería de datos y productos altamente centrados en datos ("data-centric"). Considera que el volumen y la importancia estratégica de los datos seguirá aumentando, lo cual generará nuevas oportunidades y necesidades tecnológicas que deberán ser atendidas con soluciones inteligentes y escalables.

Sin embargo, cuando reflexiona específicamente sobre la demanda de desarrolladores de software, Cecilia proyecta que en los próximos años podría mantenerse relativamente estable en términos netos, aunque con una recomposición interna: habrá menos demanda de perfiles junior y más énfasis en desarrolladores capaces de aportar valor estratégico y entendimiento profundo del negocio. Además, plantea una observación generacional importante respecto a cómo las nuevas generaciones buscan equilibrar la productividad con la calidad de vida, prefiriendo invertir los incrementos de productividad en bienestar personal en lugar de aumentos absolutos de horas trabajadas.

Finalmente, desde su posición como CEO en una consultora tecnológica, Mariani recomienda enfáticamente a otras empresas del sector implementar procesos internos de experimentación continua y transformación digital integral, utilizando activamente inteligencia artificial para optimizar sus propias operaciones. Destaca que el mayor riesgo para estas empresas es quedarse atrás respecto a competidores más ágiles y eficientes, que están capitalizando mejor las ventajas competitivas ofrecidas por la IA. En este contexto, recomienda prepararse activamente y anticiparse a los cambios inminentes que seguirán afectando tanto las expectativas de los clientes como la dinámica interna del trabajo tecnológico.

La transcripción completa de la entrevista se encuentra en [Trascripción entrevista a Cecilia Mariani](#).



José Gazzano, fundador y actual director de Seta Workshop (Seta Workshop, n.d.), cuenta con más de 13 años de experiencia en la industria tecnológica, de los cuales los últimos 8 los ha dedicado al liderazgo de una empresa de desarrollo de productos digitales a medida. Desde su perspectiva como líder tecnológico y estratégico, analiza el impacto de la inteligencia artificial (IA) en el desarrollo de software, en los perfiles técnicos y en la dinámica de la industria, aportando una visión práctica basada en su experiencia de gestión directa de equipos y proyectos en este contexto de transformación.

Gazzano identifica un cambio irreversible impulsado por la IA, tanto en las modalidades de trabajo cotidianas como en el diseño y desarrollo de productos. En su visión, las principales áreas de impacto inmediato se encuentran en la validación temprana de ideas (prototipado y pruebas de concepto) y en el testing automatizado, procesos que antes eran costosos o postergados por los clientes, pero que hoy pueden abordarse ágilmente con herramientas de IA.

En cuanto al impacto por rol, observa que los desarrolladores junior son quienes más se benefician de estas herramientas, ya que resuelven dudas técnicas de forma más rápida y accesible, lo que reduce su dependencia de otros miembros del equipo. Sin embargo, advierte un riesgo: si los juniors se apoyan exclusivamente en la IA sin explorar otras fuentes como documentación oficial o buenas prácticas, pueden limitar su desarrollo profesional. Para contrarrestar esto, Gazzano ha ajustado los procesos de formación en su empresa, promoviendo habilidades como el pensamiento crítico, la curiosidad y la autonomía desde etapas tempranas.

Sobre el futuro del rol del desarrollador, proyecta un perfil cada vez más estratégico, orientado a la resolución de problemas complejos y a la comprensión profunda del contexto del producto. Anticipa que tareas más operativas, como la codificación repetitiva, serán progresivamente delegadas a asistentes de IA, mientras que aumentará la relevancia de competencias como el diseño de arquitectura, la visión sistémica y la capacidad de comunicación.

Respecto a herramientas como Copilot, Cursor o IDEs inteligentes, señala que el impacto varía según el tipo de proyecto. En contextos de prototipado y desarrollo de nuevas funcionalidades, los beneficios son evidentes. Pero en sistemas complejos o de larga data, como una aplicación en la que trabajan desde 2019, las sugerencias de IA tienden a ser imprecisas y, a veces, contraproducentes.

En relación con el mercado laboral, Gazzano evita hacer proyecciones tajantes. Observa señales contradictorias: por un lado, la IA permite a personas sin formación técnica construir productos básicos, lo cual podría reducir la demanda de desarrolladores en etapas iniciales. Por otro lado, la necesidad de escalar, robustecer y profesionalizar estos productos genera nuevas oportunidades para empresas especializadas. En ese marco, anticipó que los equipos tenderán a ser más pequeños, pero más eficientes y estratégicos, y que los clientes tendrán expectativas más exigentes en cuanto a costos y resultados.

Finalmente, a nivel estratégico, recomienda a las empresas de servicios de software diferenciarse no solo por su capacidad técnica o por precios competitivos, sino por el acompañamiento humano y el vínculo de partnership con sus clientes. Sostiene que, en un contexto donde las herramientas están cada vez más democratizadas, el verdadero valor diferencial está en la comprensión profunda del problema, la calidad del proceso y la confianza construida.



Anthony Figueroa es ingeniero en computación y cuenta con más de 17 años de experiencia en el desarrollo de software. Desde 2011 se desempeña como CTO y cofundador de Rootstrap (Rootstrap, n.d.), empresa de desarrollo de software que presta servicios a clientes de diversas industrias en Estados Unidos. Su rol se enfoca en conectar el aspecto técnico con las necesidades del negocio.

Figueroa destaca que la irrupción de herramientas de inteligencia artificial generativa como ChatGPT ha acelerado la adopción e internalización de su potencial por parte de emprendedores y empresas. Esto ha provocado un incremento en las expectativas de los clientes respecto al software, generando cambios tanto en la forma de resolver problemas como en las ideas mismas que se proponen para desarrollar.

En cuanto al impacto de la IA en el desarrollo de software, sostiene que su alcance es transversal, afectando áreas tan diversas como documentación, onboarding, seguridad, análisis y creatividad. Más que eliminar roles, opina que todos deben evolucionar, ya que las tareas se transforman, pero no desaparecen. Especialmente en el caso del desarrollador, considera que las herramientas actuales aumentan la productividad y reducen la carga operativa, pero no reemplazan la capacidad humana de tomar decisiones, analizar problemas o definir qué camino seguir.

Sobre el futuro del desarrollador, subraya que deberán enfocarse más en el pensamiento de alto nivel, el entendimiento del negocio, y la toma de decisiones estratégicas. Para ello, recomienda que los perfiles se vuelvan más generalistas, con capacidad para observar, interpretar y moldear la realidad, en contraposición con la ultraespecialización que antes era deseable.

En relación al uso de asistentes de código, reconoce que su impacto actual es significativo, sobre todo en etapas tempranas de los proyectos, donde pueden acelerar el desarrollo en un orden de magnitud. Sin embargo, señala que en proyectos complejos y con historia, el impacto es más limitado. Aun así, advierte que subestimar su evolución futura sería un error.

En materia de formación y captación de talento, identifica como desafío principal la velocidad con la que queda obsoleta una formación estática en contextos de cambio acelerado. Plantea que la capacidad para adoptar nuevas herramientas y adaptarse a distintos contextos debería ser un criterio clave al evaluar talento, más allá del conocimiento puntual.

Sobre el mercado del software, sostiene que su tamaño continuará expandiéndose, dado que cada industria está enfrentando presión para adoptar nuevas tecnologías. Aun cuando algunas empresas sufran por no adaptarse, la demanda general seguirá creciendo. En paralelo, reconoce que la cantidad de personas necesarias para desarrollar un volumen determinado de software podría disminuir, producto de la automatización.

Finalmente, desde su rol de CTO, aconseja a las empresas de servicios tecnológicos mantenerse actualizadas, abrir el juego al uso libre de herramientas de IA por parte de sus equipos, y medir resultados sin imponer procesos rígidos. Considera que el fenómeno actual no es un cambio tecnológico más, sino una transformación de orden estructural que requiere apertura, agilidad y adaptación continua.

La transcripción completa de la entrevista se encuentra en [Trascripción entrevista a Anthony Figueroa](#).



Luis Staudenmann es ingeniero en informática, con más de 25 años de experiencia en el desarrollo de software, especialmente orientado al backend y sistemas de telecomunicaciones de alta performance. Actualmente se desempeña como CTO en Condor Technologies (Condor Technologies, n.d.), empresa que fundó en 2004, la cual desarrolla sistemas de voz sobre IP y soluciones administrativas para redes de fibra óptica. Su rol abarca la supervisión del área de ingeniería, desarrollo y diseño de producto, además de mantener un fuerte involucramiento en el plano técnico y en la formación académica como docente universitario.

Desde su perspectiva, la irrupción de la inteligencia artificial generativa ha tenido un impacto creciente pero aún dispar. Señala que muchas empresas, incluida la suya, están explorando el uso de LLMs sin una estrategia del todo definida. Sin embargo, reconoce un progreso significativo en su utilidad práctica, especialmente en la generación de código repetitivo, casos de prueba, documentación técnica y scripts. La política que promueve en su empresa es clara: el uso de IA es bienvenido como medio para ganar productividad, pero la responsabilidad del resultado sigue siendo del desarrollador.

Staudenmann enfatiza que las tareas más beneficiadas hasta ahora son aquellas de menor complejidad, donde el ahorro de tiempo es tangible. En el mediano plazo, anticipa que el impacto será más profundo y transversal en todo el proceso de desarrollo. Considera que los desarrolladores deben estar preparados para validar y adaptar las soluciones propuestas por la IA, sobre todo para cumplir con requerimientos no funcionales como robustez, escalabilidad y flexibilidad.

Desde su experiencia como CTO y docente, propone una revisión profunda de los planes de estudio universitarios. Sugiere que la formación debería balancear el conocimiento profundo de bajo nivel (como estructuras de datos, algoritmos y redes) con habilidades más abstractas como la definición de arquitecturas y el diseño de soluciones complejas. Remarca que si bien las capas de abstracción siguen creciendo, el entendimiento de fundamentos seguirá siendo esencial para diagnosticar y controlar adecuadamente el comportamiento de los sistemas.

Sobre el futuro del rol del desarrollador, destaca como competencias clave la capacidad de especificar adecuadamente requerimientos, de construir y revisar arquitecturas que cumplan objetivos complejos, y la experiencia en debugging, que considera un “superpoder” forjado con los años. Sostiene que los ingenieros deben asumir el control del proceso, comprendiendo lo que genera la IA y sabiendo cuándo intervenir.

En relación al mercado, observa una posible caída en la demanda de desarrolladores para tareas simples debido a la automatización, aunque reconoce que nuevos desafíos y roles emergentes podrían equilibrar esta tendencia. A largo plazo, considera que se generarán nuevas oportunidades, y que el rol del desarrollador debe ser el de mantenerse flexible, curioso, y atento a aplicar IA de manera estratégica, evitando caer en su uso superficial o puramente marketinero.

## 6.2 - Análisis comparativo de resultados de entrevistas

Las entrevistas realizadas a líderes de empresas tecnológicas, muchos de ellos en roles de dirección dentro de compañías de consultoría y desarrollo de software, ofrecen una mirada rica y práctica sobre cómo la inteligencia artificial está transformando el perfil del desarrollador y reconfigurando la demanda de talento técnico. A lo largo de sus testimonios, emergen no sólo percepciones compartidas sobre qué tareas y competencias están ganando o perdiendo relevancia, sino también distintas

estrategias para enfrentar este nuevo escenario. Este apartado sistematiza y compara sus visiones en torno a dos ejes principales: la evolución del rol del desarrollador, y los posibles cambios en la cantidad y tipo de demanda laboral.

Los resultados aquí presentados fueron organizados a partir del proceso de codificación temática previamente descrito, lo que permitió identificar categorías emergentes comunes en los discursos de los entrevistados.

### 6.2.1 Cambios proyectados en el rol del desarrollador según líderes de la industria tecnológica

Los testimonios de los líderes entrevistados permiten identificar distintos grados de transformación proyectada en el rol del desarrollador a partir del avance de la inteligencia artificial. Si bien todos coinciden en que habrá cambios significativos, difieren en cuanto a la profundidad y naturaleza de dicha transformación. Para organizar el análisis, las opiniones fueron clasificadas en tres categorías principales:

- Transformación estratégica del rol: El desarrollador evoluciona hacia tareas de mayor abstracción, pensamiento crítico y alineación con objetivos de negocio, delegando las tareas operativas a herramientas inteligentes.
- Cambio técnico significativo pero incremental: El trabajo del desarrollador se mantiene esencialmente técnico, pero se ve potenciado por mejoras sustanciales en productividad y eficiencia gracias al uso de IA.
- Redefinición extrema: El rol del desarrollador tradicional pierde vigencia, siendo reemplazado en gran parte por sistemas automatizados.

La siguiente tabla resume la clasificación de cada entrevistado, con citas textuales que fundamentan su ubicación:

Líder tecnológico	Clasificación	Testimonios clave
Camila Moscatelli	Transformación estratégica del rol	“El desarrollador va a tener que entender mucho más de negocio, tener pensamiento crítico y poder evaluar si el código que propone la IA tiene sentido o no.”
Juan Pablo Mazza	Transformación estratégica del rol	“Vas a tener un perfil más de orquestador que de programador: sabes usar herramientas, sabes programar, pero usas la IA para resolver el 80% del problema.”
Sebastian Melgar	Transformación estratégica del rol	“Va a ser cada vez más importante entender el negocio, hablar bien, escribir bien, coordinar equipos, y trabajar con diferentes tecnologías.”
Santiago Perez	Transformación estratégica del rol	“Cada vez más el diferencial no va a estar en saber programar, sino en entender bien el problema, tener pensamiento crítico y tomar buenas decisiones.”
Cecilia Mariani	Transformación estratégica del rol	“El rol va a ser más lógico, más de diseño. Va a haber menos juniors, y más gente que piense. Se va a necesitar más entendimiento del negocio y del problema.”

José Gazzano	Transformación estratégica del rol	“El desarrollador se va a volcar más a la resolución de problemas y toma de decisiones, y menos a la escritura de código repetitivo.” “Vamos a integrar la IA como una capa de abstracción más, cómo fue pasar de Assembler a Java.”
Anthony Figueroa	Transformación estratégica del rol	“El desarrollador debe asumir un rol activo, tomar decisiones, entender el producto y automatizar el resto”
Luis Staudenmann	Transformación estratégica del rol	"Será clave la capacidad de validar arquitecturas, controlar requerimientos no funcionales y ejercer control técnico."

La siguiente tabla muestra la cantidad y el porcentaje de expertos en cada categoría:

Categoría	Cantidad	Porcentaje
Transformación estratégica del rol	8	100%
Cambio técnico pero incremental	0	0%
Redefinición extrema del rol	0	0%

El análisis muestra un consenso claro: todos los entrevistados anticipan una transformación estratégica del rol del desarrollador. Si bien algunos matizan el ritmo o la profundidad del cambio, coinciden en que las tareas rutinarias serán progresivamente automatizadas y que las funciones humanas se orientarán cada vez más a la supervisión crítica, el diseño de soluciones y la conexión entre lo técnico y lo estratégico.

Dentro de esos matices, se destacan casos como el de Luis Staudenmann, quien reconoce que las principales transformaciones actuales son de orden técnico y funcional, aunque también proyecta un desplazamiento paulatino hacia competencias más abstractas, como la validación de arquitecturas o el diseño de soluciones robustas. Esta perspectiva híbrida refuerza la idea de que la evolución del rol no es abrupta ni uniforme, pero sí constante y con dirección clara.

En este sentido, el desarrollador del futuro no será definido solo por su dominio técnico, sino por su capacidad de integrar múltiples herramientas, interpretar problemas complejos y contribuir a la toma de decisiones con perspectiva de negocio. No se identificaron posturas que anticipen una redefinición extrema o desaparición del rol, lo que sugiere una visión pragmática: lejos de eliminar la figura del desarrollador, la IA la configura y eleva su nivel de responsabilidad.

#### 6.2.2 Cambios proyectados en la demanda del rol del desarrollador según líderes de la industria tecnológica

Las entrevistas realizadas también permiten identificar cómo proyectan los líderes de la industria tecnológica los cambios en la demanda de desarrolladores frente al avance de la inteligencia artificial. Para este análisis, se definieron cuatro categorías interpretativas principales:

- Aumento neto de la demanda: Se anticipa claramente un crecimiento sostenido en la demanda total de desarrolladores debido al incremento de productos y soluciones digitales impulsadas por IA.
- Reconfiguración interna de la demanda: Se mantiene relativamente estable la cantidad total, pero cambia la composición interna y el tipo de perfiles demandados.
- Reducción neta de la demanda: Se prevé explícitamente una disminución en la cantidad total de desarrolladores debido a la automatización generalizada.
- Ambigüedad en la proyección: Existen dudas o señales mixtas sobre cómo evolucionará la demanda futura, combinando expectativas de aumento inicial con posible disminución futura o incertidumbres sobre el impacto real.

La siguiente tabla resume la clasificación de cada entrevistado junto con testimonios representativos que fundamentan su postura:

Líder tecnológico	Clasificación	Testimonios clave
Camila Moscatelli	Reconfiguración interna de la demanda	<p>“En el corto plazo no creo que haya menos desarrolladores, pero sí menos juniors y más especializados en seguridad, análisis de datos e infraestructura.”</p> <p>“La demanda va a virar claramente hacia perfiles con más capacidad técnica y estratégica.”</p>
Juan Pablo Mazza	Ambigüedad en la proyección	<p>“A corto plazo habrá más demanda, pero a largo plazo podría bajar debido a la enorme productividad por la automatización.”</p> <p>“Todavía no sabemos realmente hasta dónde va a escalar la productividad, por eso es difícil pronosticar con precisión.”</p>
Sebastian Melgar	Ambigüedad en la proyección	<p>“Vamos hacia equipos más chicos y más productivos, lo que podría implicar menos desarrolladores por proyecto.”</p> <p>“A corto plazo esto incrementa oportunidades, pero más adelante esto podría reducir drásticamente la cantidad de desarrolladores necesarios.”</p>
Santiago Perez	Reconfiguración interna de la demanda	<p>“La cantidad total de desarrolladores probablemente se mantenga estable, pero sí veo un reacomodamiento fuerte en el tipo de perfiles requeridos.”</p> <p>“Se necesitarán nuevas habilidades específicas más estratégicas y técnicas.”</p>
Cecilia Mariani	Reconfiguración interna de la demanda	<p>“La demanda total probablemente sea similar, pero claramente veo menos necesidad de perfiles juniors y más demanda de seniors que entiendan el negocio.”</p> <p>“Equipos más pequeños y estratégicos serán más efectivos.”</p>

José Gazzano	Ambigüedad en la proyección	“Quizás haya menos contrataciones por parte de las empresas, pero también más proyectos en paralelo.” “El impacto total aún es incierto, hay que esperar cómo evolucionan las expectativas de costo y tiempo.”
Anthony Figueroa	Ambigüedad en la proyección	"Va a crecer la demanda de software, pero puede disminuir la hacerlo."
Luis Staudenmann	Ambigüedad en la proyección	“Probablemente el desarrollador se enfrente a nuevos desafíos, así que no es fácil saber si el número total va a subir o bajar.”

La siguiente tabla muestra la cantidad y el porcentaje de líderes tecnológicos en cada categoría:

Categoría	Cantidad	Porcentaje
Aumento neto de la demanda	0	0%
Reconfiguración interna de la demanda	3	37,5%
Reducción neta de la demanda	0	0%
Ambigüedad en la proyección	5	62,5%

Los resultados revelan que la mayoría de los entrevistados no proyecta cambios drásticos en la cantidad total de desarrolladores requeridos, sino una transformación en la composición del talento demandado.

El 37,5% de los líderes prevén que la demanda total de desarrolladores no experimentará cambios radicales en términos absolutos, pero sí una transformación significativa en su composición interna. Estos líderes coinciden claramente en anticipar una reducción notable en la necesidad de perfiles junior, acompañada por un marcado aumento en la demanda de profesionales más especializados en IA, experimentados y estratégicos. En este nuevo contexto, será clave la capacidad para aprovechar herramientas inteligentes y aportar valor desde áreas como el diseño lógico, la arquitectura y la comprensión profunda del negocio.

Sin embargo, un 62,5% de los entrevistados manifiesta una postura ambigua respecto a la evolución del volumen de la demanda. Si bien reconocen que la inteligencia artificial puede abrir nuevas oportunidades en el corto plazo, también advierten que a mediano o largo plazo podría producirse una disminución en la cantidad total de desarrolladores requeridos, principalmente debido al aumento en la productividad habilitado por la automatización. Esta incertidumbre se debe, en parte, a la dificultad actual para prever con precisión hasta qué punto las herramientas inteligentes sustituirán tareas tradicionalmente humanas.

En síntesis, aunque existe una visión compartida sobre el cambio en el tipo de talento requerido, no hay consenso sobre la dirección que tomará la demanda total de desarrolladores. Lo que sí se mantiene como constante es la necesidad de adaptación: todos los entrevistados coinciden en que la

capacidad para integrar tecnologías emergentes, repensar el valor que aporta el desarrollador y adquirir una mirada más estratégica serán claves para sostener su relevancia en los próximos años.

## CAPÍTULO 7 - DATOS SECUNDARIOS

### 7.1 - Proyecciones de crecimiento del mercado del software

Diversas firmas de análisis de mercado realizan proyecciones sobre la evolución futura del sector del software a nivel global. Si bien existen diferencias metodológicas y segmentaciones específicas en cada reporte, aquellas fuentes que adoptan una visión integral del mercado permiten estimar la magnitud potencial de este sector en los próximos años.

Según Statista, se espera que el mercado global de software alcance ingresos de aproximadamente 740.89 mil millones de dólares en 2025, impulsado por la creciente demanda de soluciones digitales en sectores como finanzas, salud, educación y comercio. A partir de esa base, Statista proyecta una tasa de crecimiento anual compuesta (CAGR) del 4.87% entre 2025 y 2029, alcanzando un volumen estimado de 896.17 mil millones de dólares en 2029 (Statista, 2024).

Statista elabora sus proyecciones mediante un enfoque cuantitativo basado en modelos estadísticos, que integran datos de múltiples fuentes: reportes financieros de empresas líderes del sector, publicaciones especializadas, estudios de terceros, información de oficinas estadísticas nacionales y tendencias macroeconómicas globales. Además, los modelos se ajustan manualmente mediante el juicio experto de sus analistas. Las proyecciones se construyen por segmentos (por ejemplo, software empresarial o de infraestructura) y consideran tanto el desarrollo tecnológico como la evolución del gasto digital en distintas geografías (Statista, 2023).

Una estimación similar es provista por Mordor Intelligence, que proyecta que el mercado global de software alcanzará los 720 mil millones de dólares en 2025 y crecerá a una CAGR del 11.23%, hasta alcanzar aproximadamente 1.23 billones de dólares en 2030 (Mordor Intelligence, 2024). Esta proyección, levemente más optimista en términos de crecimiento, destaca como principales motores de expansión la adopción acelerada del software en la nube, el auge de soluciones móviles, la necesidad de plataformas de analítica avanzada y el papel de la inteligencia artificial como habilitador transversal en distintos sectores económicos.

El enfoque metodológico de Mordor Intelligence combina investigación secundaria con validación primaria. Por un lado, recopilan información de fuentes públicas y privadas, incluyendo reportes de empresas, datos gubernamentales, documentos regulatorios y literatura académica. Por otro, complementan esta base con entrevistas a expertos de la industria, lo cual permite ajustar las estimaciones según percepciones cualitativas, identificar nuevas tendencias emergentes y validar supuestos de crecimiento. Esta metodología mixta les permite abordar no sólo proyecciones cuantitativas, sino también factores estratégicos y contextuales que podrían influir en la evolución del mercado (Mordor Intelligence, n.d.).

A pesar de diferencias en las cifras específicas, ambas fuentes coinciden en prever un crecimiento sólido y sostenido del mercado global de software en los próximos años, lo que confirma su papel central como componente estratégico en la economía digital contemporánea. En la Ilustración 14. Proyección de crecimiento del mercado global de software (2025–2030) se observan los datos antes mencionados tanto de Statista como Mordor Intelligence.

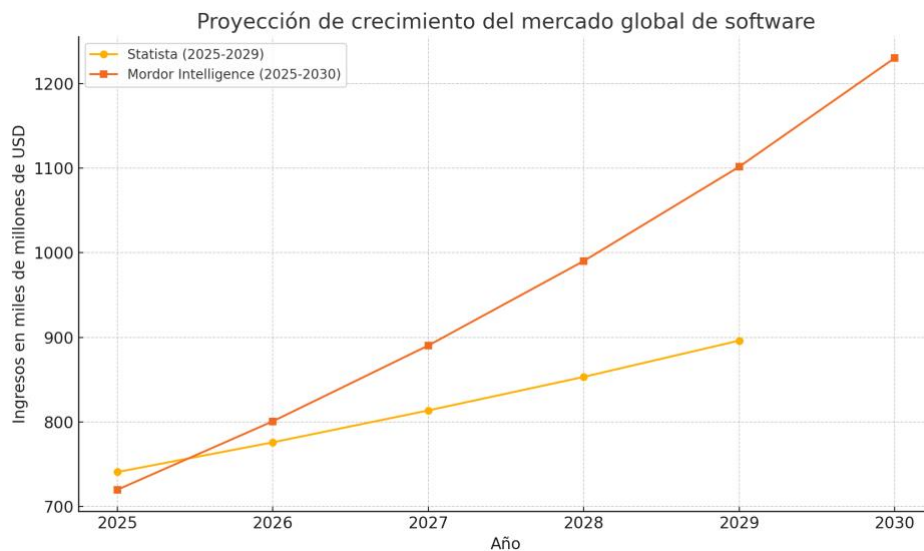


Ilustración 14. Proyección de crecimiento del mercado global de software (2025–2030)

**Nota.** Elaboración propia basada en datos de Statista (Statista, 2024) y Mordor Intelligence (Mordor Intelligence, n.d.).

## 7.2 - Proyecciones de empleo en el sector tecnológico

Diversas organizaciones especializadas realizan estimaciones sobre la evolución futura del mercado laboral en sectores específicos. En el ámbito tecnológico, dos entidades destacadas en Estados Unidos, la U.S. (Bureau of Labor Statistics, 2024) *of Labor Statistics* (BLS) y la *Computing Technology Industry Association* (CompTIA), ofrecen proyecciones fundamentadas que permiten anticipar tendencias y planificar estrategias en relación con la demanda de talento en el sector.

El BLS proyecta que el empleo de desarrolladores de software crecerá un 17.9% entre 2023 y 2033, pasando de 1,692,100 a 1,995,700 puestos, lo que equivale a una tasa de crecimiento anual compuesta (CAGR) del 1.66% (Bureau of Labor Statistics, 2024), ver Ilustración 15. Proyección de empleo de desarrolladores de software en EE. UU. (2024–2034). Estas proyecciones se derivan de un modelo cuantitativo que estima la evolución del empleo en distintos sectores y ocupaciones. El proceso inicia con proyecciones sobre el crecimiento poblacional y la participación laboral, continúa con estimaciones del Producto Interno Bruto (PIB) y su distribución sectorial, y culmina con la conversión de estos datos en empleo esperado por industria y ocupación. En este marco, el impacto de nuevas tecnologías, como la inteligencia artificial, la automatización o el desarrollo de software asistido por máquinas, se incorpora de manera indirecta, a través de supuestos sobre productividad sectorial, sustitución de tareas rutinarias y evolución de la composición del empleo (Bureau of Labor Statistics, 2024).

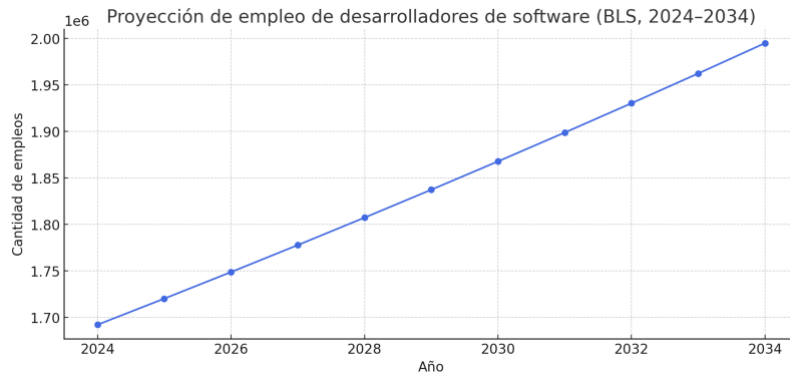


Ilustración 15. Proyección de empleo de desarrolladores de software en EE. UU. (2024–2034)

**Nota.** Elaboración propia basada en datos del Bureau of Labor Statistics (Bureau of Labor Statistics, 2024).

Por ejemplo, si se anticipa que una tecnología emergente permitirá producir más bienes o servicios con menos horas de trabajo, el modelo ajustará a la baja el crecimiento del empleo en ese sector. De igual forma, si se prevé que ciertas ocupaciones experimentarán una transformación significativa debido a la adopción de tecnología, se redistribuyen funciones hacia otros roles con mayor componente creativo, analítico o estratégico. Sin embargo, estas estimaciones no modelan tecnologías específicas ni integran validación cualitativa con referentes del sector, lo que podría subestimar disrupciones aceleradas o no lineales.

CompTIA, por su parte, estima que el empleo tecnológico en Estados Unidos aumentará de 6 millones en 2024 a 7.1 millones en 2034, lo que representa un crecimiento del 18.3% en dicho período y una CAGR aproximada del 1.7% (CompTIA, 2024), ver Ilustración 16. Proyección de empleo total en el sector tecnológico según datos de CompTIA (2024–2034). Esta organización complementa los datos del BLS con encuestas propias y fuentes privadas como Lightcast. A diferencia del BLS, que se centra específicamente en desarrolladores de software y analistas de control de calidad, CompTIA abarca una gama más amplia de roles tecnológicos, incluyendo profesionales de soporte técnico, especialistas en redes, analistas de datos y otros puestos relacionados con TI. Además, CompTIA incorpora datos de ofertas de empleo y encuestas a empleadores para ajustar sus proyecciones, lo que puede explicar las diferencias en las estimaciones entre ambas organizaciones.

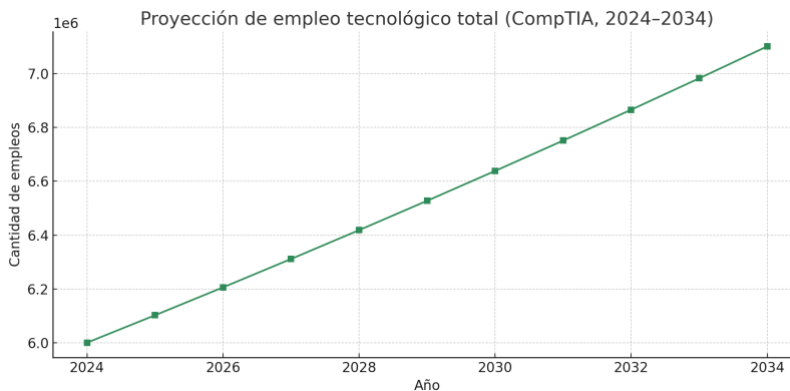


Ilustración 16. Proyección de empleo total en el sector tecnológico según datos de CompTIA (2024–2034)

**Nota.** Elaboración propia basada en datos de CompTIA (CompTIA, 2024).

Ambos informes coinciden en prever un crecimiento sostenido en la demanda de talento tecnológico. No obstante, reconocen que la incorporación de herramientas de inteligencia artificial aumenta la productividad individual, lo cual podría reducir la necesidad de ciertos perfiles mientras incrementa la demanda de roles especializados en integración, supervisión y uso estratégico de sistemas basados en IA. Es importante destacar que, aunque las proyecciones son optimistas, la magnitud real del crecimiento dependerá en gran medida del impacto efectivo de la inteligencia artificial y otras tecnologías emergentes en el mercado laboral. Esta es una variable cuyo efecto es aún difícil de cuantificar con precisión, y es posible que las estimaciones actuales estén subestimando el alcance de su disrupción.

### 7.3 - Voces de expertos de la industria sobre el futuro del rol del desarrollador y su demanda

En este capítulo se recogen opiniones de expertos y referentes mundiales sobre la temática en investigación, enfocándose en cómo la inteligencia artificial (IA) está transformando el rol del desarrollador de software y su demanda en el mercado laboral. Las perspectivas se agrupan en dos secciones: la transformación del rol del desarrollador y el impacto en la demanda de desarrolladores.

#### 7.3.1 - Resumen de voces de expertos

Sam Altman, de código a la estrategia y arquitecto de sistemas



En una entrevista con *Stratechery* en abril de 2025, Sam Altman, CEO de OpenAI, se refirió a la transformación del rol del desarrollador en un contexto dominado por la inteligencia artificial (Altman, 2025). Desde su perspectiva, la programación tradicional dejará de ser el núcleo de la tarea del desarrollador, dando paso a una función más abstracta y orientada al diseño de sistemas. Altman sostiene que los ingenieros de software del futuro no se destacarán por escribir líneas de código, sino por su capacidad de estructurar problemas, definir objetivos y guiar modelos de IA para que implementen soluciones de forma autónoma. En este nuevo paradigma, el valor del desarrollador residirá menos en su conocimiento técnico detallado y más en su habilidad para pensar en términos estratégicos y de producto.

Respecto a las competencias necesarias para los desarrolladores del futuro, Altman enfatiza que será clave saber cómo comunicarse eficazmente con modelos de inteligencia artificial. Esto incluye formular preguntas de forma precisa, entender cómo funciona el modelo y verificar sus respuestas. Según él, escribir código será menos importante que saber cómo guiar al sistema para que lo genere correctamente. También destaca que muchas de las personas más efectivas trabajando con estas herramientas no necesariamente vendrán de la ingeniería tradicional, sino de perfiles capaces de unir conocimientos técnicos con pensamiento creativo, habilidades de diseño y una buena comprensión de productos. En lugar de centrarse en los detalles del código, los nuevos desarrolladores deberán tener una visión más global y enfocarse en cómo crear soluciones útiles trabajando junto a estas herramientas inteligentes.

Sobre la demanda de desarrolladores, Altman sugiere que esta no desaparecerá, sino que cambiará su composición. Habrá menos demanda de programadores tradicionales y más de “arquitectos de sistemas IA”, personas que sepan conectar piezas, validar resultados y crear productos completos en los que la IA cumple un rol operativo. De hecho, considera que la democratización del desarrollo, posibilitada por herramientas como ChatGPT y Copilot, ampliará la cantidad de personas que participan en la creación de software.

## Thomas Dohmke: el desarrollador como guía de la inteligencia artificial



En una conversación con Sequoia Capital, Thomas Dohmke (Sequoia Capital, 2024), CEO de GitHub, se refirió a la evolución del rol del desarrollador en un entorno cada vez más influenciado por herramientas de inteligencia artificial como GitHub Copilot. Según Dohmke, la figura del desarrollador se ampliará notablemente: ya no será una actividad restringida a ingenieros altamente capacitados, sino una competencia que millones de personas podrán adquirir y aplicar. Sostiene que los modelos generativos de código permitirán que individuos sin formación tradicional en programación creen software funcional, lo que democratiza el acceso al desarrollo tecnológico. Este fenómeno, lejos de suponer una amenaza, representa una oportunidad para repensar la manera en que la sociedad entera se involucra en la construcción de soluciones digitales.

Respecto a las competencias que serán más relevantes, Dohmke destaca que la clave estará en la habilidad para colaborar con sistemas inteligentes. Esto implica aprender a comunicarse con asistentes como Copilot, verificar el código generado, y, sobre todo, entender el contexto en el que una solución debe funcionar. La lógica de "programar desde cero" será reemplazada por una lógica de revisión, adaptación y guía. El conocimiento técnico seguirá siendo valioso, pero perderá su rol exclusivo: personas con pensamiento crítico, visión de producto y capacidad de iterar soluciones con ayuda de la IA podrán desempeñar un rol activo en el desarrollo, incluso sin dominar lenguajes de programación tradicionales.

En cuanto a la demanda futura de desarrolladores, Dohmke ofrece una visión contraria a la idea de reemplazo. Según su estimación, para 2030 habrá más de mil millones de personas participando en tareas de desarrollo de software, en gran parte gracias a la accesibilidad que brindan herramientas como Copilot. Este pronóstico no solo sostiene que la demanda continuará, sino que se expandirá de manera significativa. La IA no reducirá la necesidad de desarrollo, sino que habilitará nuevos casos de uso, proyectos y emprendedores. Además, Dohmke destaca que esta expansión conlleva un cambio de paradigma: pasar de pensar en desarrolladores como una élite técnica, a considerarlos como facilitadores de innovación distribuida en todos los sectores sociales y económicos.

## Benedict Evans: más software, nuevos roles y nuevas preguntas



Benedict Evans es un reconocido analista de tecnología, conocido por su trabajo en la firma de capital de riesgo Andreessen Horowitz y por sus publicaciones independientes sobre la evolución de las plataformas digitales, inteligencia artificial y tendencias en software. Su enfoque combina el análisis estratégico con una comprensión profunda de la historia tecnológica, lo que le permite proyectar escenarios futuros con una perspectiva crítica. En su artículo *Working with AI* (Benedict Evans, 2023), Evans analiza el impacto de la inteligencia artificial en el trabajo del conocimiento, centrándose en cómo esta tecnología podría transformar profundamente el desarrollo de software.

Evans sostiene que la IA no eliminará el rol del desarrollador, sino que lo reformuló. A medida que los modelos generativos de código se vuelvan más sofisticados, los desarrolladores dejarán de enfocarse en escribir cada línea manualmente para concentrarse en tareas de mayor nivel como el diseño de sistemas, la estructuración de soluciones y la supervisión del trabajo realizado por la IA. El trabajo técnico será cada vez más complementado por herramientas automatizadas, desplazando el valor hacia la capacidad de dar buenas instrucciones, interpretar resultados, y mantener la coherencia

conceptual de los productos. Evans compara esta transformación con la aparición de las hojas de cálculo: su masificación no eliminó los trabajos relacionados con finanzas o contabilidad, pero sí redefinió cómo se ejercen y quiénes podían desempeñarlos.

Sobre la evolución de la demanda, Evans advierte que la reducción de barreras técnicas derivada del uso de IA no necesariamente implica una caída en la necesidad de desarrolladores. Por el contrario, al abaratar y facilitar el proceso de construcción de software, se ampliarán los casos de uso, las industrias involucradas y la cantidad de personas capaces de generar soluciones digitales. En este sentido, anticipa una expansión significativa del software como herramienta transversal, lo que generará nuevas oportunidades de desarrollo, tanto para perfiles técnicos tradicionales como para nuevos actores que se incorporen mediante herramientas accesibles. La automatización, lejos de sustituir completamente a los profesionales del desarrollo, funcionará como catalizador para una mayor producción de software y una mayor penetración de este en la vida económica y social.

Mark Zuckerberg – ¿El fin del desarrollador semi senior? Una nueva era de desarrollo asistido por IA



Mark Zuckerberg, CEO de Meta, ha sido una de las voces más contundentes en advertir sobre el impacto inminente de la inteligencia artificial en el desarrollo de software. En una entrevista publicada en abril de 2024, sostuvo que para 2025 las herramientas de IA desarrolladas por su compañía y otras similares podrían reemplazar en gran medida a los ingenieros de software de nivel medio (semi senior), al ser capaces de generar código funcional de forma autónoma y eficiente (Tech Gig, 2024).

Si bien reconoció que actualmente estos sistemas pueden ser costosos de operar, anticipó que su rendimiento mejorará rápidamente, lo que les permitirá asumir una proporción significativa de las tareas de codificación rutinarias. Para Zuckerberg, este avance marcará una transformación profunda del rol del desarrollador, quien dejará de centrarse exclusivamente en escribir código para enfocarse en tareas estratégicas como definir objetivos técnicos, validar resultados generados por IA y supervisar decisiones complejas durante el proceso de desarrollo.

Este desplazamiento de las tareas operativas hacia funciones de mayor abstracción sugiere una reconfiguración del perfil profesional del desarrollador. En palabras del propio Zuckerberg, los ingenieros del futuro deberán responder preguntas como “¿qué?”, “¿cómo?”, “¿dónde?” y “¿por qué?”, dejando que la IA ejecute los “cómo” más concretos del código.

Zuckerberg también situó esta evolución dentro de un fenómeno más amplio que ya está impactando a toda la industria: otras empresas líderes, como Google e IBM, están adoptando modelos similares donde la mayoría del nuevo código es generado por IA y luego revisado por desarrolladores humanos. Esta tendencia no sólo redefine el trabajo cotidiano del desarrollador, sino que también podría transformar la estructura misma de la demanda de talento en el sector.

Sundar Pichai – De codificadores a curadores



Sundar Pichai, CEO de Google y Alphabet, ha delineado una visión transformadora sobre el impacto de la inteligencia artificial (IA) en el desarrollo de software. Durante la presentación de resultados del tercer trimestre de 2024, Pichai reveló que más del 25% del nuevo código en Google ya es generado por IA, siendo posteriormente revisado y aprobado por ingenieros humanos. Este

enfoque ha permitido a los desarrolladores de la compañía aumentar su productividad y avanzar más rápidamente en sus proyectos (The Economic Times, 2024).

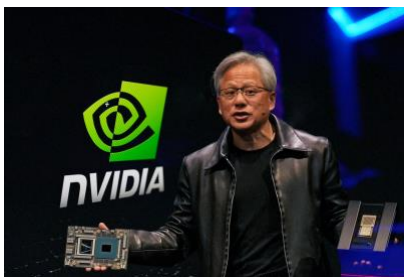
Pichai enfatizó que la IA no solo está redefiniendo los procesos internos de Google, sino que también está impulsando un cambio fundamental en el rol de los desarrolladores. En lugar de centrarse exclusivamente en la escritura de código, los ingenieros ahora asumen funciones más estratégicas, como la supervisión del código generado por IA y la toma de decisiones arquitectónicas clave. Este cambio sugiere una transición del desarrollador tradicional hacia un perfil más orientado a la curaduría y dirección de proyectos tecnológicos.

Además, Pichai ha expresado su optimismo respecto al impacto de la IA en el mercado laboral. Aunque reconoce que la inteligencia artificial transformará profundamente la forma en que se desarrolla software, considera que esta transformación no implicará una eliminación masiva de empleos, sino una redefinición de las habilidades necesarias. En lugar de reemplazar completamente a los desarrolladores, la IA permitirá aumentar su productividad y liberar tiempo para que puedan enfocarse en tareas más estratégicas, creativas y centradas en la resolución de problemas complejos (infoabe, 2023).

Según Pichai, los ingenieros de software del futuro deberán combinar habilidades técnicas con pensamiento crítico, diseño de soluciones y entendimiento del negocio, en lugar de limitarse a la ejecución de instrucciones técnicas. En este contexto, la demanda no necesariamente crecerá en términos absolutos, pero sí se desplazará hacia perfiles más versátiles y con mayores capacidades de abstracción. De este modo, la IA actuaría como un catalizador para el surgimiento de nuevos roles híbridos, tanto en el desarrollo como en áreas complementarias como la ética de la IA, la evaluación de resultados generados por modelos, la ingeniería de prompts y la supervisión de sistemas autónomos.

En palabras de Pichai, la IA es "una de las tecnologías más profundas en las que estamos trabajando hoy" y "afectará todo: cada sector, cada industria y cada tipo de trabajo" (Juan, 2024). Sin embargo, también ha remarcado que esto representa una oportunidad para reimaginar la educación, el trabajo y la creación de software desde una lógica más inclusiva y escalable, donde más personas puedan participar del proceso de desarrollo apoyadas por estas nuevas herramientas.

Jensen Huang – De programadores a usuarios



Jensen Huang, CEO de NVIDIA, ha adoptado una postura radical respecto al futuro de la programación como disciplina profesional. Durante su intervención en la Cumbre Mundial de Gobiernos en Dubái (2024), Huang afirmó que, con los avances recientes de la inteligencia artificial, "ya no es necesario aprender a programar". Según su visión, la IA ha alcanzado tal nivel de sofisticación que el lenguaje humano se está convirtiendo en el nuevo lenguaje de programación, desplazando la necesidad de conocimientos técnicos tradicionales (Juan, 2024).

Huang remarcó que durante décadas se incentivó el aprendizaje de programación como una habilidad fundamental para participar en el mundo digital. Sin embargo, considera que esa etapa está llegando a su fin: "la programación será realizada por la inteligencia artificial", y el valor diferencial de las personas residirá en su capacidad para interactuar con estos sistemas, no en escribir código línea por línea (Juan, 2024).

Esta afirmación implica una redefinición profunda del rol del desarrollador. Si la escritura de código pasa a ser responsabilidad de la IA, entonces los ingenieros deberán adaptarse a un entorno donde su aporte estará más ligado a la definición de objetivos, el diseño de soluciones y la validación de resultados. En otras palabras, el foco ya no estará en el “cómo”, sino en el “qué” y el “por qué”.

Además, Huang sugirió que las nuevas generaciones que deseen trabajar en tecnología deberían considerar disciplinas emergentes como la biología digital, en lugar de la informática tradicional. Desde su perspectiva, la ingeniería del futuro se desarrollará en áreas como la biotecnología, la energía y los materiales, en combinación con sistemas inteligentes que realizarán la mayor parte del trabajo técnico (Juan, 2024).

En este escenario, el rol del desarrollador tradicional se ve fuertemente tensionado: ya no se trata de codificar, sino de colaborar con agentes inteligentes. Lejos de un simple cambio de herramientas, esta visión plantea una transformación estructural en la demanda laboral y en la definición misma de lo que significa “desarrollar software”.

Erik Brynjolfsson – ¿Quién se beneficia realmente? Evidencia empírica sobre IA y productividad en el trabajo técnico



Erik Brynjolfsson, economista especializado en transformación digital y director del Stanford Digital Economy Lab, ha investigado durante décadas el impacto de las tecnologías emergentes en la productividad y el mercado laboral. En colaboración con Danielle Li (MIT Sloan) y Lindsey Raymond (MIT), publicó en 2023 un estudio empírico titulado *Generative AI at Work*, que ofrece evidencia concreta sobre los efectos de la inteligencia artificial generativa en entornos de trabajo técnico (Brynjolfsson, Li, & Raymond, 2023).

El estudio se centra en una herramienta de IA generativa implementada en una empresa de software con más de 5.000 agentes de soporte técnico. Aunque el foco no está en desarrolladores de software, los hallazgos tienen claras implicancias para el debate sobre el futuro del rol del programador.

Los resultados muestran que el uso de IA generativa aumentó la productividad general de los trabajadores en un 14%, con mejoras particularmente pronunciadas entre los perfiles menos experimentados (hasta un 34%). Este efecto se explica por la capacidad de la IA para acelerar el aprendizaje de los trabajadores y ayudarlos a replicar las mejores prácticas del colectivo:

“AI assistance improves performance most for novice and low-skill workers, helping them move more quickly down the experience curve” (Brynjolfsson, Li, & Raymond, 2023).”

En contraste, el impacto sobre los trabajadores más experimentados fue nulo o incluso negativo. El estudio señala que:

“AI assistance does not lead to any productivity increase for the most skilled workers [...] and may decrease the quality of conversations by the most skilled agents” (p. 19).

Estos hallazgos cuestionan la narrativa más extendida en el discurso de líderes de la industria tecnológica, que plantea que los desarrolladores migrarán a roles estratégicos, convirtiéndose en “curadores” o “orquestadores” de código generado por IA. En cambio, la evidencia empírica de Brynjolfsson sugiere que la IA podría reducir la brecha entre junior y senior, automatizando la transferencia de conocimiento y homogeneizando los niveles de productividad.

Los autores aclaran que el estudio no permite concluir cómo afectará esto a la demanda laboral futura, y que hay múltiples escenarios posibles:

“Firms may respond to increasing productivity among novice workers by hiring more of them, de-skilling positions, or seeking to develop more powerful AI systems that can replace lower-skill workers entirely” (p. 3).

Esta observación introduce una tensión relevante con otros expertos, mientras el discurso dominante celebra la evolución del desarrollador hacia un perfil más estratégico, la evidencia todavía es ambigua. La IA no necesariamente refuerza el valor de los perfiles más senior; incluso podría debilitarlo, facilitando la sustitución de ciertas tareas expertas o volviéndose más fácilmente replicables. Esta diferencia entre las visiones normativas (basadas en liderazgo de opinión) y los estudios empíricos plantea interrogantes importantes sobre qué perfiles serán realmente más demandados en los equipos técnicos del futuro.

Tim O’Reilly – La IA como herramienta de ampliación del pensamiento humano



Tim O’Reilly, fundador de O’Reilly Media y figura clave en el pensamiento tecnológico contemporáneo, sostiene que la inteligencia artificial (IA) no está desplazando a los desarrolladores de software, sino transformando profundamente su rol. A su entender, la IA es una extensión de la tendencia histórica de facilitar la programación, permitiendo que más personas creen soluciones digitales con menos barreras técnicas (O’Reilly, 2023). Lejos de eliminar puestos de trabajo, O’Reilly argumenta que este proceso amplía la creatividad humana y crea nuevas oportunidades laborales.

En su artículo *The End of Programming as We Know It*, O’Reilly (O’Reilly, 2023) plantea que estamos ante una transición en la que programar ya no significa únicamente escribir código, sino estructurar ideas para resolver problemas complejos. Este cambio lleva al desarrollador a adoptar un rol más abstracto, donde el pensamiento sistémico, el diseño y la capacidad de articular correctamente un problema pasan a ser tan o más importantes que el dominio de un lenguaje de programación. Apoyado por experiencias concretas con herramientas como GitHub Copilot, describe cómo los ingenieros senior pueden generar funciones completas en minutos, siempre que cuenten con la experiencia necesaria para guiar, validar y refinar las propuestas de la IA. En contraposición, advierte que perfiles menos experimentados tienden a aceptar las salidas de la IA sin suficiente juicio crítico, lo que puede derivar en “código castillo de naipes”, funcional a simple vista, pero propenso a fallar bajo presión.

Otro aspecto clave que resalta es la colaboración entre humanos e IA, donde la supervisión experta humana es imprescindible para que la IA sea útil y confiable. En esta relación, el desarrollador deja de ser un simple ejecutor técnico y pasa a ocupar un rol de director técnico y estratega, capaz de aportar el contexto de negocio y las consideraciones sistémicas que la IA no puede deducir por sí sola (O’Reilly, 2023).

Respecto a la demanda futura de desarrolladores, O’Reilly ofrece una mirada optimista pero exigente. En una entrevista con *Software Engineering Daily*, afirma que cuando la tecnología simplifica una tarea, no destruye el empleo asociado, sino que lo transforma y lo expande (O’Reilly, 2018). Para él, cada revolución en la programación ha traído más personas al proceso de creación de software, y la IA generativa no será la excepción. La diferencia estará en el tipo de profesional que se necesita: desarrolladores capaces de trabajar junto a la IA, aprender continuamente y asumir mayor responsabilidad en el diseño y control de calidad del software. Su advertencia es clara: “No serán los programadores los que quedarán sin trabajo, sino aquellos (en cualquier industria), que no aprendan a convertirse en programadores asistidos por IA” (O’Reilly, 2023).

En suma, la visión de Tim O'Reilly destaca que el rol del desarrollador no desaparece, sino que se reinventa. El foco se desplaza de la ejecución técnica a la definición estratégica y la supervisión inteligente. En este nuevo paradigma, la IA no reemplaza al desarrollador: lo potencia, y, a la vez, exige más de él.

### 7.3.2 - Análisis comparativo de las voces de expertos sobre el futuro del rol del desarrollador y su demanda

Este capítulo sintetiza y compara las perspectivas de los expertos analizados en la sección 7.3.1, con el objetivo de identificar puntos de consenso, tensiones y divergencias sobre dos temas clave: la evolución del rol del desarrollador de software y la demanda futura de estos profesionales, en línea con las preguntas de investigación de esta tesis.

#### 7.3.2.1 - Cambios proyectados en el rol del desarrollador

Los expertos presentan diferentes visiones sobre la magnitud del cambio que la inteligencia artificial generativa implica para el rol del desarrollador. Para sistematizar estas posturas, se los ha clasificado en cuatro categorías:

- Transformación del rol hacia un perfil estratégico y orquestador de IA: el desarrollador deja de centrarse en escribir código para adoptar un rol de estrategia, curador, arquitecto o diseñador de sistemas. Utilizan herramientas de IA para generar gran parte del código, y se enfocan en supervisar, validar y dirigir su integración en soluciones más amplias.
- Cambios parciales o evolución significativa pero técnica: el trabajo cambia, pero se mantiene dentro del marco tradicional del desarrollo. El foco sigue siendo técnico, pero se observa un aumento significativo en la productividad individual gracias al uso de herramientas de IA.
- Sin cambios sustanciales: el rol se mantiene sin modificaciones relevantes.
- Redefinición extrema del rol: el desarrollador como tal deja de ser necesario; la programación se vuelve una tarea accesible a través del lenguaje natural y mayormente delegada a sistemas inteligentes.

La siguiente tabla muestra la categoría en la que se ubica cada autor según su visión sobre el cambio en el rol del desarrollador:

Experto	Clasificación
Sam Altman	Transformación del rol hacia un perfil estratégico y orquestador de IA
Thomas Dohmke	Transformación del rol hacia un perfil estratégico y orquestador de IA
Tim O'Reilly	Transformación del rol hacia un perfil estratégico y orquestador de IA
Sundar Pichai	Cambios parciales o evolución significativa pero técnica
Benedict Evans	Cambios parciales o evolución significativa pero técnica
Erik Brynjolfsson	Cambios parciales o evolución significativa pero técnica
Mark Zuckerberg	Transformación del rol hacia un perfil estratégico y orquestador de IA

	IA
Jensen Huang	Redefinición extrema del rol

La siguiente tabla muestra la cantidad y el porcentaje de expertos en cada categoría:

Categoría	Cantidad	Porcentaje
Transformación del rol hacia un perfil estratégico y orquestador de IA	5	62,5%
Cambios parciales o evolución significativa pero técnica	2	25%
Redefinición extrema del rol	1	12,5%
Sin cambios sustanciales	0	0%

Es relevante destacar que ningún experto sostiene que el rol del desarrollador permanecerá sin cambios, lo que refuerza la idea de una transformación inevitable en el sector.

Entre todas las voces analizadas, Jensen Huang representa la postura más divergente: propone una redefinición extrema del rol del desarrollador, anticipando un escenario en el que la programación ya no sea una habilidad central, sino que se realice a través del lenguaje natural y con mínima intervención humana. Esta visión, más radical y potencialmente pesimista, plantea interrogantes sobre la centralidad futura del desarrollador como figura profesional.

En contraposición, la visión predominante entre los expertos apunta a una transformación del desarrollador hacia un perfil estratégico, en el que adquiere protagonismo como orquestador de sistemas inteligentes. Estos perfiles deberán comprender a fondo el "qué" y el "para qué" del software, dominar herramientas de IA, y supervisar el trabajo automatizado para integrarlo de forma efectiva en soluciones reales. Esta transición exige habilidades más analíticas, comunicacionales y de diseño, y refuerza el valor de quienes puedan combinar pensamiento sistémico con criterio técnico.

#### 7.3.2.2 - Cambios proyectados en la demanda del rol del desarrollador

Las opiniones sobre la demanda futura de desarrolladores también varían significativamente. Se han clasificado en cuatro categorías:

- Aumentará: la demanda crece por expansión de aplicaciones de software e integración de IA.
- Se mantendrá estable: la cantidad se mantiene, aunque cambia el perfil requerido.
- Disminuirá: menor necesidad de programadores humanos por sustitución.
- No concluye / ambigua: evidencia insuficiente para una afirmación definitiva.

La siguiente tabla muestra la categoría en la que se ubica cada autor según su visión sobre la demanda futura de desarrolladores:

Experto	Clasificación	Aclaración contextual
Sam Altman	Aumentará	Visualiza que aumentará la cantidad de personas que participan en la creación de software, aunque no necesariamente en el rol tradicional del desarrollador.

Thomas Dohmke	Aumentará	Concibe al desarrollador como facilitador de innovación, no como una élite técnica, con un perfil más inclusivo y centrado en guiar a la IA.
Tim O'Reilly	Aumentará	Argumenta que la IA ampliará el alcance del desarrollo de software y traerá más personas al proceso creativo, manteniendo al desarrollador como figura clave.
Sundar Pichai	Aumentará	Prevé un cambio en las tareas del desarrollador, pero sin una reducción ni aumento claro en la demanda.
Benedict Evans	Se mantendrá estable	Sostiene que habrá más software por crear, lo que implica una mayor necesidad de desarrolladores en general.
Erik Brynjolfsson	No concluye / ambigua	Su investigación muestra efectos mixtos: los juniors se benefician más, pero no se concluye si eso incrementará o reducirá la demanda.
Mark Zuckerberg	Disminuirá	Asegura que su IA podría reemplazar a desarrolladores semi-senior en el corto plazo, reduciendo la necesidad de estos perfiles.
Jensen Huang	Disminuirá	Considera que la programación será completamente delegada a la IA y que ya no será necesario que las personas aprendan a programar.

La siguiente tabla muestra la cantidad y el porcentaje de expertos en cada categoría:

Categoría	Cantidad	Porcentaje
Aumentará	4	50%
Se mantendrá estable	1	12,5%
Disminuirá	2	25%
No concluye / ambigua	1	12,5%

A diferencia de lo observado respecto al rol del desarrollador, las posturas sobre la demanda futura no siempre son explícitas ni unívocas. En muchos casos, se infiere la clasificación a partir del contexto o de frases indirectas, lo cual introduce un mayor grado de interpretación.

En este escenario, se destaca que la mitad de los expertos analizados considera que la demanda de desarrolladores aumentará, aunque bajo definiciones más amplias y diversas del perfil profesional. Al mismo tiempo, el resto de las voces exhibe posiciones más diversas, incluyendo tanto advertencias sobre sustitución parcial como posturas empíricas que evitan pronósticos concluyentes. Esta heterogeneidad refuerza la necesidad de observar las trayectorias futuras con una mirada abierta, considerando no solo la cantidad, sino también la calidad y configuración de los nuevos roles técnicos.

### 7.3.3 - Reflexión final sobre opinión de expertos

La evidencia analizada permite afirmar que existe un consenso parcial en torno a la transformación del rol del desarrollador: la mayoría de los expertos coinciden en que se volverá más abstracto, estratégico y orientado a la colaboración con sistemas de inteligencia artificial. Las divergencias se

concentran en la proyección sobre la demanda futura: mientras algunos anticipan una expansión de oportunidades, otros alertan sobre la posibilidad de reemplazo parcial o reconfiguración del mercado laboral.

Estas diferencias reflejan no sólo distintos marcos interpretativos (normativos, empíricos, tecnológicos), sino también el carácter altamente contingente del escenario. Por eso, este trabajo sostiene la necesidad de observar, anticipar y adaptarse a los posibles futuros: para los desarrolladores, esto implica adquirir nuevas competencias; para las empresas de servicios tecnológicos, reimaginar sus modelos operativos y de formación interna.

La convergencia de voces con diferentes perspectivas, desde fundadores de compañías de IA, hasta economistas laborales y expertos en infraestructura digital, refuerza la relevancia y urgencia de las preguntas que guían esta investigación.

# CONCLUSIONES Y RECOMENDACIONES

## CAPÍTULO 8 - PRODUCTIVIDAD, ROL Y DEMANDA FUTURA DEL DESARROLLADOR

Las conclusiones descritas en este capítulo, responden a dos de las preguntas centrales de investigación: *¿Cómo está cambiando la productividad del desarrollador de software, su rol actual y la composición de los equipos de desarrollo con la adopción de IA?*, y *¿Cómo impactará la inteligencia artificial en la demanda de desarrollo de software a corto y mediano plazo?*, se basan en entrevistas con líderes de tecnología, el análisis de opiniones de expertos recogidas en medios especializados, y datos secundarios provenientes de organismos reconocidos como la U.S. Bureau of Labor Statistics (BLS) y CompTIA.

Las conclusiones se organizan en conclusiones primarias, las cuales responden directamente a las preguntas de investigación de la tesis, y conclusiones secundarias, que no responden directamente a las preguntas de investigación, pero ayudan a comprender mejor el futuro de la industria y el rol del desarrollador de software.

### 8.1 - Conclusiones primarias

#### 8.1.1 - Aumento de productividad

Diversos estudios recientes conducidos por Microsoft y Google, analizados en el capítulo 3.2 - IA y Productividad en el Desarrollo de Software, coinciden en que la adopción de herramientas de inteligencia artificial generativa en el desarrollo de software está produciendo un aumento significativo en la productividad de los desarrolladores. Estas investigaciones, publicadas en 2023, reportan mejoras del orden del 20% al 25% en tareas como la generación de código, pruebas, la lectura de documentación técnica o la resolución de bugs. Si bien estos resultados ya son alentadores, es importante tener en cuenta que el ritmo de cambio en esta tecnología es vertiginoso, y que tanto las herramientas como sus formas de uso están evolucionando de manera acelerada.

Esta perspectiva cuantitativa encuentra eco en las entrevistas a los ocho líderes de tecnología analizadas en el capítulo 6 6.1 - Resumen de entrevistas, si bien la mayoría no asigna porcentajes exactos, todos coinciden en que la adopción de copilotos y otras herramientas de IA ya produjo mejoras perceptibles en la velocidad de desarrollo, la reducción de retrabajo y la calidad del código. Es decir, la hipótesis de mayor productividad se ha convertido en un hecho observable en la práctica diaria, reforzando la validez externa de los hallazgos reportados por Microsoft y Google. Esto se alinea también con datos de la encuesta a desarrolladores de Stack Overflow presentados en 3.1.3 - Adopción de herramientas AI según Stack Overflow, donde el 72 % de los desarrolladores afirma que las herramientas de IA les ayudan a ser más productivos, lo que refuerza tanto la percepción subjetiva como el impacto medido.

Esta tendencia puede ser interpretada a través del modelo SPACE presentado en el capítulo 1.4.2 - Modelos recientes: SPACE y sus aplicaciones, que amplía la noción de productividad más allá de la simple velocidad de entrega. El salto de productividad del 20%–25 % señalado en los estudios de Microsoft y Google se materializa, ante todo, en la dimensión Actividad, vinculada al aumento de tareas completadas por unidad de tiempo, el mismo se presentó en 3.2 - IA y Productividad en el Desarrollo de Software. Desde la evidencia primaria presentada en 6.1 - Resumen de entrevistas, existen testimonios que refuerzan

mejoras tanto en el ámbito de Actividad, como también, Eficiencia, Flujo, Satisfacción y Bienestar, completando otras dimensiones contempladas en SPACE:

- Camila Moscatelli, Directora de Ingeniería en Rootstrap, sostiene que *“Código que antes era más repetitivo o sencillo, hoy puede generarse automáticamente, y el desarrollador pasa cada vez más a un rol de supervisor, chequeador de código, y no tanto quien lo escribe línea por línea”*.
- Juan Pablo Mazza, CTO de Effectus Software, afirma que *“ya que permite diseñar arquitecturas o flujos conversando directamente con un agente de IA, reduciendo la necesidad de tantas interacciones humanas para validar ideas. Es una forma muy ágil de avanzar en etapas tempranas.”*
- Sebastián Melgar, CEO de Neocoast, destaca que *“Hoy, dentro de IT, no veo un rol específico que esté siendo perjudicado por la IA; de hecho, la mayoría se beneficia gracias a las nuevas herramientas y a la mayor eficiencia que permiten”*.
- Santiago Pérez, Director de Ingeniería en Rootstrap, observa que *“Hoy la IA permite generar especificaciones más detalladas, asistir en la producción de código con cierto grado de supervisión por parte del desarrollador, y automatizar pruebas, tanto unitarias como funcionales. También en etapas como diseño empieza a tener un rol importante”*.

Estas mejoras también se reflejan en percepciones más amplias del sector: un 65 % de los encuestados en Stack Overflow, indicó que las herramientas de IA les permiten aprender nuevas tecnologías más rápido, lo que sugiere una experiencia laboral más fluida, desafiante y enriquecedora.

En resumen, la evidencia sugiere que la inteligencia artificial tiene un potencial disruptivo y transformador sobre la productividad en el desarrollo de software. A medida que su adopción se masifica y madura, ya presente en más del 70 % de los desarrolladores según Stack Overflow, es probable que varios paradigmas tradicionales de trabajo, tamaño de equipo y planificación se vean desafiados o reconfigurados.

Sin embargo, su adopción exitosa no depende solo de la disponibilidad técnica, sino de entender la productividad como un fenómeno multidimensional, en línea con el modelo SPACE, y de alinearla con los valores organizacionales, los procesos existentes y las capacidades del equipo técnico.

#### 8.1.2 - Evolución estratégica del rol del desarrollador

La irrupción de herramientas de inteligencia artificial generativa está impulsando una evolución progresiva, aunque no homogénea, en el rol del desarrollador de software. Lejos de representar una ruptura total con el pasado, este cambio puede interpretarse como una aceleración de un proceso ya en marcha. Tal como se describe en el capítulo 1.2 - Principales responsabilidades del desarrollador, las funciones del desarrollador ya habían comenzado a extenderse más allá de la mera escritura de código, incorporando decisiones relacionadas con el producto, la experiencia de usuario y la estrategia técnica. En este sentido, la IA no transforma el rol desde cero, sino que acentúa su desplazamiento hacia tareas de mayor valor agregado, como la toma de decisiones, la priorización de soluciones y la colaboración con múltiples áreas.

Como concluye el capítulo el capítulo 6.2.1 Cambios proyectados en el rol del desarrollador según líderes de la industria tecnológica, la totalidad de los entrevistados anticipa una transformación estratégica del rol, se espera que el desarrollador combine destreza técnica con capacidad de análisis de negocio y toma de decisiones. Esa visión se refuerza con testimonios del 6.1 - Resumen de entrevistas.

- Camila Moscatelli, Directora de Ingeniería en Rootstrap, subraya que *“Lo que antes tal vez se esperaba de un arquitecto... hoy empiezo a esperarlo cada vez más de la gran mayoría del equipo: esta capacidad de ser crítico con el código, de entender la solución que está construyendo y de vincular los objetivos de negocio y producto con la solución técnica”*.
- Sebastian Melgar, CEO y fundador de Neocoast, agrega: *“Vas a tener un perfil más de orquestador que de programador: sabes usar herramientas, sabes programar, pero usas la IA para resolver el 80 % del problema”*, además menciona *“va a ser cada vez más importante entender el negocio, hablar bien, escribir bien, coordinar equipos y trabajar con diferentes tecnologías”*.
- Juan Pablo Mazza, CTO de Effectus Software exclama *“...uno de los chicos comentaba que ya no siente que está programando tanto, sino que dedica más tiempo a pensar cómo escribir el prompt adecuado para que la herramienta programe por él... se invierte más tiempo en orquestar que en escribir código línea por línea”*.
- Santiago Pérez, Director de Ingeniería en Rootstrap, enfatiza que *“Este tipo de herramientas tienden a nivelar lo técnico; el verdadero diferencial va a estar en habilidades como la comunicación, el trabajo en equipo y en entender bien el problema”*.
- Cecilia Mariani, CEO de Darfowft menciona que *“El verdadero valor aparece cuando el desarrollador entiende el producto y su contexto... no se trata de ejecutar al pie de la letra un requisito, sino de proponer la mejor implementación”*.
- Anthony Figueroa, CTO y fundador de Rootstrap exclama que *“El desarrollador se va a volcar más a la resolución de problemas y toma de decisiones, y menos a la escritura de código repetitivo”*
- Jose Gazzano, CEO y fundador de Seta Workshop dice *“El rol va a ser más lógico, más de diseño... se va a necesitar más entendimiento del negocio y del problema”*

El capítulo 7.3.2.1 - Cambios proyectados en el rol del desarrollador sistematiza las opiniones de referentes internacionales del sector tecnológico y concluye que, si bien hay matices en cuanto al nivel de disrupción esperado, existe consenso sobre una transformación del rol. Gradualmente, la parte del trabajo centrada en escribir código va siendo reemplazada o asistida por herramientas automatizadas, lo que permite al desarrollador dedicar más tiempo a funciones de mayor nivel, como el diseño, la integración y la supervisión de soluciones generadas por inteligencia artificial. En este proceso, el foco se desplaza hacia entender el negocio y el producto para poder brindar las soluciones más efectivas en cada contexto.

Incluso en el acuerdo sobre la transformación del rol del desarrollador de los referentes internacionales del sector tecnológico, existen diferentes voces, las cuales se presentan en el capítulo 7.3.1 - Resumen de voces de expertos. Algunos plantean escenarios altamente disruptivos. *Jensen Huang (CEO de NVIDIA)* representa la visión más radical al afirmar que *“ya no es necesario aprender a programar [...]; la IA programará y las personas deberán definir objetivos y supervisar resultados”*, sugiriendo un futuro donde la codificación tradicional dejaría de ser una habilidad central. Sin embargo, esta postura no es la predominante. La mayoría de los expertos tiende a visiones más graduales y complementarias. Por ejemplo, *Tim O’Reilly* considera que *“programar va a significar estructurar ideas para resolver problemas complejos, más que escribir líneas de código”*, marcando una transición desde la ejecución literal hacia un enfoque más analítico y creativo. En la misma línea, *Thomas Dohmke (CEO de GitHub)* plantea que *“el 80 % del código será generado por IA”*, pero enfatiza que *“el verdadero valor estará en saber cómo pedir lo correcto, cómo evaluar el resultado y cómo integrarlo en un sistema”*. En conjunto, estos aportes reafirman que el rol del desarrollador se está transformando, no por pérdida de valor, sino por redefinición de foco: desde la codificación directa hacia la toma de decisiones, la integración estratégica y la interacción eficaz con sistemas inteligentes.

En síntesis, tanto los líderes de empresas tecnológicas entrevistados, 6.1 - Resumen de entrevistas, como los expertos analizados en fuentes secundarias, 7.3.1 - Resumen de voces de expertos, coinciden, con diferentes matices, en que el rol del desarrollador está transitando una transformación estructural.

#### 8.1.3 - Tendencia a la reducción del tamaño de los equipos

### El incremento de productividad detallado en 8.1 - Conclusiones primarias

8.1.1 - Aumento de productividad, de 20–25 % según estudios conducidos por Microsoft y Google, y confirmado por voces de líderes y expertos, implica que cada desarrollador aporta ahora más valor individual. Si un equipo alcanza el mismo output con menos personas, la lógica de optimización de Scrum cobra aún más relevancia: su “guía base” propone 3-9 miembros para minimizar líneas de comunicación, como se vio en 1.3.2 - Impacto del tamaño y estructura del equipo. La IA amplifica ese principio, habilitando “pods” mas pequeños capaces de entregar lo que antes exigía un equipo completo.

En el capítulo 6.1 - Resumen de entrevistas, los líderes de tecnología aportan evidencia directa:

- Cecilia Mariani, CEO de Darwoft: *“Con IA y componentes reutilizables, equipos más chicos logran la misma entrega: donde antes hablábamos de un scrum de 7-9, podríamos ver 3-5 desarrolladores capaces de producir lo mismo”*.
- Sebastián Melgar, CEO de Neocoast, coincide en que la IA *“podría reducir considerablemente la cantidad de recursos humanos necesarios por proyecto”*.
- Luis Staudenmann, CTO en Condor Technologies: *“Con el aumento de la productividad por desarrollador, deberías necesitar menos gente para hacer lo mismo”*.
- José Gazzano, CEO y fundador de Seta Workshop: *“Equipos más pequeños y estratégicos serán más efectivos”*.

Estas observaciones de campo confirman el postulado de Scrum sobre la relación inversa entre tamaño del equipo y overhead de coordinación y refuerzan la idea, ya discutida en 8.1.2 - Evolución estratégica del rol del desarrollador, de que los perfiles técnicos se vuelven más versátiles y estratégicos a medida que la IA asume tareas operativas.

En suma, dado la teoría Scrum presentada en 1.1.1 - El marco Scrum, la evidencia empírica proveniente de las entrevistas a líderes en 6.1 - Resumen de entrevistas, y resultados cuantitativos de productividad presentados en 3.2 - IA y Productividad en el Desarrollo de Software, se confirma la tendencia de que la IA permite equipos más pequeños, autónomos y multifuncionales. La clave no es encoger por encoger, sino rediseñar la distribución de responsabilidades: menos tiempo en ejecución repetitiva, más en diseño, integración y supervisión de soluciones inteligentes, siempre ajustando el tamaño del equipo a la complejidad y el riesgo del proyecto.

#### 8.1.4 - Ambigüedad en la demanda

A pesar del consenso general en torno al impacto de la inteligencia artificial sobre la productividad, el cambio de rol y la transformación del talento técnico, no existe una visión unificada sobre cómo evolucionará la demanda total de desarrolladores de software. Mientras algunas voces sostienen que la adopción de herramientas inteligentes reducirá la necesidad de contratar tantos perfiles técnicos, otras argumentan que el abaratamiento y la accesibilidad del desarrollo impulsarán la creación de más productos, proyectos y empresas, generando nuevas oportunidades laborales.

En el capítulo 7.2 - Proyecciones de empleo en el sector tecnológico se presento que entidades reconocidas como BLS y CompTIA preveen un crecimiento de entre 17% y 18% en la proxima decada en el empleo de desarrolladores y profesiones tecnológicas. Sin embargo, estas estimaciones se basan en supuestos de productividad histórica e incorporan parcialmente y mediante estimaciones los potenciales cambios en la productividad causados por la IA analizados en 8.1 - Conclusiones primarias

8.1.1 - Aumento de productividad. En dicha linea, de acuerdo con la Paradoja de Jevons (Alcott, 2025), una innovación que abarata la producción, en este caso, software, puede desencadenar tanto efecto sustitución (menos personas por proyecto) como efecto expansión (más proyectos porque el costo relativo baja). El signo final de la demanda depende de cuál de esos dos efectos domine.

Esta tensión se refleja en los testimonios recogidos con los lideres de tecnologia. En el capítulo 6.2.2 Cambios proyectados en la demanda del rol del desarrollador según líderes de la industria tecnológica se consolida las opiniones de los líderes respecto al futuro de la demanda. Un 62,5% se ubica en una posición ambigua, aceptando que el head-count por proyecto podría bajar, pero matizando con la eventual aparición de más iniciativas digitales, mientras que 37,5% prevé una reconfiguración interna: la cantidad total permanecería parecida, pero con menos perfiles junior y más especialistas en IA y negocio.

A continuación se ilustran esas posturas con citas textuales del 6.1 - Resumen de entrevistas:

- Camila Moscatelli – Directora de Ingenieria en Rootstrap: *“En el corto plazo no creo que haya menos desarrolladores, pero sí menos juniors y más especializados en seguridad, análisis de datos e infraestructura.”*.
- Juan Pablo Mazza – CTO y fundador de Effectus Software: *“A corto plazo habrá más demanda, pero a largo plazo podría bajar debido a la enorme productividad por la automatización”*.
- Sebastián Melgar – CEO y fundador de Neocoast: *“Vamos hacia equipos más chicos y más productivos, lo que podría implicar menos desarrolladores por proyecto.”*.
- Santiago Pérez – Director de Ingenieria en Rootstrap: *“La cantidad total de desarrolladores probablemente se mantenga estable, pero sí veo un reacomodamiento fuerte en el tipo de perfiles requeridos.”*.
- Anthony Figueroa – CTO y fundador en Rootstrap: *“Va a crecer la demanda de software, pero puede disminuir la cantidad de personas necesarias para hacerlo.”*.
- Cecilia Mariani - CEO de Darwoft *“La demanda total probablemente sea similar, pero claramente veo menos necesidad de perfiles juniors y más demanda de seniors que entiendan el negocio”*.

Desde la documentación secundaria, el capítulo 7.3.2.2 - Cambios proyectados en la demanda del rol del desarrollador sistematiza las opiniones de ocho referentes globales, y nuevamente se aprecia la tensión en la opinión sobre el futuro de la demanda: 50% espera que la demanda aumente, 12,5% la ve estable, 25% augura una disminución y 12,5% se declara ambigua.

Sus matices quedan claros en los testimonios del capítulo 7.3.1:

- Tim O’Reilly: *“Cuando la tecnología simplifica una tarea, no destruye el empleo asociado, sino que lo transforma y lo expande.”*.
- Sam Altman: anticipa que habrá *“más personas participando en la creación de software, aunque no necesariamente en el rol tradicional del desarrollador.”*.

- Jensen Huang (visión más disruptiva): *“Ya no es necesario aprender a programar; la IA programará y las personas deberán definir objetivos y supervisar resultados.”* .
- Erik Brynjolfsson advierte que la IA *“homogeneiza la productividad”* y que las firmas podrían terminar contratando más *novice workers...* o prescindir de ellos si el sistema los reemplaza.

En este contexto, lo que parece haber es un acuerdo implícito en que la composición de la demanda cambiará significativamente, aun si la cantidad total de puestos se mantiene o incluso crece. Se requerirán menos perfiles puramente operativos, más perfiles técnicos con enfoque estratégico, y nuevas funciones híbridas que integren IA, diseño, negocio y tecnología. Esta transformación implica desafíos tanto para los trabajadores como para las organizaciones, que deberán adaptarse a un mercado con lógicas diferentes a las que dominaron en décadas anteriores.

En síntesis, la ambigüedad en las proyecciones sobre la demanda de talento no es un signo de incertidumbre analítica, sino una expresión de la complejidad de un proceso en curso. Entender estas tensiones es clave para diseñar estrategias de formación, planificación y desarrollo organizacional que puedan sostenerse en escenarios múltiples, y no solo en un futuro único o lineal.

## 8.2 - Conclusiones secundarias

### 8.2.1 - Reconfiguración del talento: seniority, competencias y responsabilidades

Las mejoras de productividad, el cambio de rol del desarrollador y la reducción del tamaño de los equipos están provocando una reconfiguración del talento que demandan las organizaciones. Como se repasa en el capítulo 1.2 - Principales responsabilidades del desarrollador, el rol del desarrollador gradualmente ha dejado de ser un rol exclusivamente técnico y donde la única actividad es desarrollar código. La IA ha acentuado este cambio, las habilidades técnicas aisladas ya no son un diferencial. se buscan profesionales capaces de orquestar soluciones con IA, enlazar lo técnico con la estrategia y decidir con autonomía en contextos de negocio dinámicos.

Los líderes entrevistados en 6.1 - Resumen de entrevistas describen con claridad qué perfil se valora:

- *Camila Moscatelli – Rootstrap: “Lo que antes tal vez se esperaba de un arquitecto... hoy empiezo a esperarlo cada vez más de la gran mayoría del equipo: esta capacidad de ser crítico con el código, de entender la solución que construye y de vincular los objetivos de negocio y producto con la solución técnica”.*
- *Juan Pablo Mazza – Effectus: “Uno de los chicos comentaba que ya no siente que está programando tanto, sino que dedica más tiempo a pensar cómo escribir el prompt adecuado para que la herramienta programe por él... se invierte más tiempo en orquestar que en escribir código línea por línea”.*
- *Sebastián Melgar – Neocoast: “Va a ser cada vez más importante entender el negocio, hablar bien, escribir bien, coordinar equipos y trabajar con diferentes tecnologías”.*
- *Santiago Pérez – Rootstrap: “Este tipo de herramientas tienden a nivelar lo técnico; el verdadero diferencial va a estar en habilidades como la comunicación, el trabajo en equipo y en entender bien el problema”.*
- *Cecilia Mariani – Darwoft: “La demanda total probablemente sea similar, pero veo menos necesidad de perfiles juniors y más demanda de seniors que entiendan el negocio”.*
- *José Gazzano – Seta Workshop: “Ajustamos el foco en nuestras capacitaciones; antes un junior que siguiera la especificación alcanzaba. Hoy también tiene que hacerse preguntas, analizar, buscar distintas soluciones”.*

En cuanto al nivel de experiencia, la investigación primaria confirma una tendencia a valorar perfiles con visión integral, pero sin concluir que los juniors vayan a desaparecer. De hecho, los estudios de Microsoft vistos en 3.2.1 - Estudio de Microsoft (2023): IA y productividad en una muestra diversa de desarrolladores muestran que los mayores saltos de productividad se dieron en desarrolladores con poca experiencia, mientras que el trabajo de Google visto en 3.2.2 - Estudio de Google: IA y su impacto en el desarrollo empresarial registró mejoras comparables, e incluso superiores, en equipos senior. La evidencia cuantitativa, por tanto, respalda lo que expresan los líderes: no existe una inclinación unívoca hacia más o menos seniority; el impacto depende del contexto y de cómo se integre la IA en el flujo de trabajo.

Del análisis de los ocho testimonios se agruparon las menciones en cinco clústeres principales, cada cluster contiene una competencia que fue mencionada en al menos cuatro entrevistas:

<b>Clúster de competencia</b>	<b>Descripción resumida</b>	<b>Ejemplo literal</b>
Comunicación y trabajo en equipo	Saber explicar, coordinar y negociar con perfiles no técnicos.	<i>“Habilidades como la comunicación y el trabajo en equipo pueden convertirse en verdaderas diferencias”</i> —Santiago Pérez
Entendimiento del negocio y visión de producto	Traducir objetivos comerciales en decisiones técnicas.	<i>“...vincular los objetivos de negocio y producto con la solución técnica”</i> — Camila Moscatelli
Pensamiento crítico y supervisión de IA	Evaluar, validar y corregir el output automatizado.	<i>“...ser crítico con el código, entender la solución que construye...”</i> —Camila Moscatelli
Adaptabilidad y aprendizaje continuo	Mantenerse al día ante la rápida evolución de herramientas.	<i>“Cada vez van a pesar más la adaptabilidad, flexibilidad...”</i> —Juan Pablo Mazza
Prompt-engineering y orquestación de sistemas IA	Formular indicaciones eficaces y combinar múltiples herramientas.	<i>“...dedica más tiempo a pensar cómo escribir el prompt adecuado... se invierte más tiempo en orquestar...”</i> —Juan Pablo Mazza

Estas competencias se mencionan de manera transversal también por los expertos del 7.3.1 - Resumen de voces de expertos, reforzando la convergencia entre teoría y práctica.

En definitiva, la inteligencia artificial no sólo redefine las herramientas que se utilizan en los equipos técnicos, sino también quiénes los componen y qué se espera de cada integrante. La capacidad de adaptación, el aprendizaje continuo y la visión sistémica se consolidan como atributos centrales para mantenerse relevante en un entorno en constante transformación.

### 8.2.2 - Democratización del software

La aparición de herramientas de inteligencia artificial generativa, plataformas visuales de creación y asistentes de código está reduciendo de forma acelerada las barreras de entrada al desarrollo de software. Esta tendencia, conocida como democratización del software, implica que cada vez más personas, incluidos profesionales sin formación técnica formal, puedan crear soluciones digitales. Al mismo tiempo, el menor costo relativo de construir software permite que más empresas accedan a tecnología propia, expandiendo su uso incluso en sectores donde antes era económicamente inviable. En el capítulo 7.1 - Proyecciones de crecimiento del mercado del software se presentan datos sobre el futuro del mercado global del software. Según Statista, la tasa de crecimiento anual compuesta (CAGR) del 4.87% entre 2025 y 2029, mientras que Mordor Intelligence, proyecta CAGR del 11.23 desde 2025 a 2030. En ambos casos la tasa de crecimiento es mas acelerada que la mayoría de las

industrias, e indican que el software superara en facturación a industrias tradicionales, lo que refuerza la idea de que la demanda de nuevas aplicaciones se amplía al ritmo de esta democratización.

La visión de los líderes entrevistados 6.1 - Resumen de entrevistas confirma el fenómeno:

- *Camila Moscatelli – Rootstrap: “Hoy un product designer puede generar un MVP bastante completo sin esperar a que desarrollo le asigne tiempo; eso cambia por completo la cadencia del equipo.”.*
- *Anthony Figueroa – Rootstrap: “Con las nuevas herramientas no-code y la IA, cualquier persona del área de negocio puede automatizar un proceso o armar un prototipo funcional.»*
- *Sebastián Melgar – Neocoast: «La barrera de entrada bajó muchísimo: vemos diseñadores y PMs escribiendo pequeños scripts y conectando APIs gracias a copilotos.”.*

Las voces de expertos internacionales 7.3.1 - Resumen de voces de expertos van en la misma dirección:

- *Tim O’Reilly: “El desarrollo de software se está convirtiendo en una habilidad transversal; la tecnología expande el empleo porque pone la creación en manos de mucha más gente.”.*
- *Sam Altman: “Habrá más personas participando en la creación de software, aunque no necesariamente bajo el rol tradicional del desarrollador.”.*
- *Jensen Huang: “Ya no será necesario aprender a programar; la IA programará y las personas deberán definir objetivos y supervisar resultados.”.*

Lejos de amenazar al desarrollador profesional, la democratización amplía el ecosistema de creación: las áreas de negocio, diseño o producto pueden materializar ideas con rapidez, mientras los perfiles técnicos siguen siendo esenciales para garantizar escalabilidad, seguridad y calidad. El resultado es un proceso de desarrollo más participativo, integrado y ágil, capaz de acelerar la innovación y responder con mayor rapidez a los desafíos del entorno digital.

### 8.2.3 - Actualización de la formación académica

La transformación del rol del desarrollador descrita en 8.1.2 - Evolución estratégica del rol del desarrollador, hacia perfiles más estratégicos, colaborativos y capaces de interactuar críticamente con sistemas de inteligencia artificial no sólo tiene consecuencias para la industria del software, sino también para el sistema educativo que forma a estos profesionales. En el capítulo 8.2.1 - Reconfiguración del talento: seniority, competencias y responsabilidades se describen los cambios esperados en competencias y responsabilidades de los desarrolladores desde la expectativa de líderes y expertos en tecnología. Frente a este nuevo escenario, resulta fundamental que las carreras universitarias, en particular las ingenierías en sistemas e informática revisen y actualicen sus planes de estudio.

En ese sentido, la formación universitaria enfrenta el desafío de incorporar competencias transversales que históricamente han tenido menor peso en los planes de estudio de ingeniería. Como menciona, Luis Staudenmann, CTO de Condor Technologies, y también profesor en el ITBA, *“Yo creo que sí. Definitivamente creo que [los programas] se tienen que adaptar, y mucho”.* Esta actualización curricular no solo facilita una mejor inserción laboral de los egresados, sino que también prepararía a los futuros profesionales para un entorno de trabajo en constante transformación, donde el aprendizaje continuo y la capacidad de adaptación serán tan importantes como la base técnica.

## CAPÍTULO 9 - RECOMENDACIONES PARA CONSULTORAS DE TECNOLOGÍA

En el CAPÍTULO 8 - PRODUCTIVIDAD, ROL Y DEMANDA FUTURA DEL DESARROLLADOR se sintetizan los hallazgos obtenidos a partir de la evidencia empírica, las entrevistas a líderes del sector y las opiniones de expertos. Allí se muestra, en primer lugar, que la adopción de inteligencia artificial eleva de forma verificable la productividad de los equipos de desarrollo; en segundo término, que el rol del desarrollador evoluciona hacia un perfil más estratégico y orquestador de la IA; y, finalmente, que la eficiencia alcanzada con estas herramientas habilita la conformación de equipos más pequeños, multidisciplinarios y orientados a resultados. No obstante, las proyecciones sobre la demanda futura de talento siguen siendo ambiguas, con un optimismo apenas moderado, debido a la tensión entre el aumento de la productividad individual y la posible redistribución de tareas entre perfiles.

A partir de esas conclusiones, este capítulo presenta un conjunto de recomendaciones estratégicas para las empresas de consultoría tecnológica que deseen anticiparse a la nueva dinámica del mercado y transformar la disrupción de la IA en una ventaja competitiva sostenible.

### 9.1 - Resumen de recomendaciones

Antes de desarrollar en profundidad cada una de estas recomendaciones, a continuación se presenta un resumen ejecutivo que permite visualizar el conjunto de iniciativas propuestas y su objetivo principal.

#### Estrategia y posicionamiento

1. **Posicionamiento “AI-Expert” y socio estratégico:** Construir una identidad diferencial basada en capacidades concretas en IA, y establecer vínculos de largo plazo como socios de negocio, más allá del rol de proveedor técnico.
2. **Nuevos modelos de pricing para capturar el valor de la IA:** Reformular esquemas de precios y contratos para reflejar el valor estratégico generado con IA, saliendo de la lógica exclusivamente basada en horas.
3. **Demostrar experiencia real en IA:** Implementar IA en procesos internos, desarrollar materiales comerciales especializados y generar evidencia concreta de casos de uso exitosos.
4. **Alianzas estratégicas con el ecosistema IA:** Establecer vínculos con plataformas, universidades y startups del ecosistema para acelerar capacidades y mantenerse actualizado.

#### Organización del delivery y estructura de equipos

5. **Equipos pequeños con expertise en IA:** Diseñar células de trabajo pequeñas, multidisciplinarias y orientadas a resultados, que maximicen el impacto del uso de IA y reduzcan tiempos de desarrollo.
6. **Reducir los tiempos de entrega y salida a producción:** Mejorar los procesos internos y aprovechar herramientas de IA para responder a la creciente demanda de ciclos de desarrollo más cortos.
7. **Generalistas sobre especialistas:** Promover desarrolladores con visión amplia, capaces de orquestar herramientas y colaborar en equipos multidisciplinarios, por sobre perfiles hiperespecializados.
8. **Roles especialistas en IA:** Incorporar perfiles técnicos con foco en IA, como MLOps, Data Scientists, Data Engineers o AI Engineers, que puedan liderar iniciativas complejas dentro de los equipos.

9. **Arquetipo del desarrollador del futuro:** Reconfigurar el perfil del desarrollador hacia un rol más estratégico, capaz de integrar IA, comprender el negocio y colaborar en entornos ágiles.
10. **Desarrollo de competencias clave para desarrolladores:** Establecer un mapa claro de habilidades técnicas y blandas necesarias para liderar el desarrollo de productos en esta nueva etapa.
11. **Alfabetización en IA en toda la organización:** Promover una cultura organizacional que comprenda las capacidades, limitaciones y oportunidades de la IA en todos los niveles y áreas.

## 9.2 - Desarrollo de Recomendaciones

A continuación, se desarrollan en profundidad cada una de las recomendaciones resumidas previamente. Para cada caso, se presenta el fundamento que las respalda, su relevancia frente a los desafíos actuales del sector, y ejemplos o lineamientos que pueden guiar su implementación por parte de empresas de consultoría tecnológica.

### 9.2.1 - Posicionamiento “AI-Expert” y socio estratégico

En marketing se distinguen los puntos de paridad, atributos que todo proveedor debe garantizar, de los puntos de diferenciación, aquello que realmente separa una oferta de la competencia (Mixpanel, 2024). La evidencia presentada confirma que la IA se ha convertido en punto de paridad: más del 70 % de los desarrolladores ya utiliza herramientas basadas en IA de acuerdo a la entrevista realizada por Stack Overflow presentadas en el capítulo 3.1.3 - Adopción de herramientas AI según Stack Overflow, y los estudios de Microsoft (3.2.1 - Estudio de Microsoft (2023): IA y productividad en una muestra diversa de desarrolladores) y Google (3.2.2 - Estudio de Google: IA y su impacto en el desarrollo empresarial) muestran incrementos de productividad del 20%-25%. Así, decir “usamos IA” sólo habilita la conversación inicial; ya no genera una ventaja competitiva per se.

Para sostener ese piso, las consultoras deben asegurarse de que sus desarrolladores integren IA en su flujo diario, copilotos de código, MCPs, etc, y dominen conceptos como LLMs, RAG, arquitecturas multi-agente y fine-tuning ligero. Tal como advierte Camila Moscatelli en las entrevistas presentadas en 6.1 - Resumen de entrevistas, *“el desarrollador pasa cada vez más a un rol de supervisor, chequeador de código, y no tanto quien lo escribe línea por línea”*. Esta evolución coincide con la importancia que la literatura otorga a los sistemas multi-agente en la práctica actual presentado en 2.3 - El auge de los agentes inteligentes y sistemas multiagente, a la explosión de la IA generativa presentado en 2.2 - El boom de la IA generativa, y a la creciente adopción en la industria de soluciones IA presentado en 2.2.2 - Impacto de la IA generativa en la industria. El conocimiento funcional de estas herramientas también debe abarcar a diseñadores, Product Managers, QA Engineers y Project Managers, quienes estiman costes de inferencia, afinan pruebas generadas por IA o coordinan la orquestación de agentes.

La confianza del cliente se fortalece cuando percibe que el equipo no solo domina la tecnología, sino que sabe aplicarla para generar impacto de negocio. En ese punto, la IA deja de ser una palabra de moda y se convierte en una ventaja efectiva y medible. Esto se alinea con lo presentado en el capítulo 6.1 - Resumen de entrevistas, donde Cecilia Mariani plantea: *“la verdadera oportunidad de diferenciación aparece cuando la consultora no se presenta sólo como experta técnica, sino como un aliado estratégico que entiende el negocio”*. En la misma línea, José Gazzano destaca que ser partner estratégico implica también *“acompañar desde lo humano”*, pues *“el componente relacional sostiene el vínculo a largo plazo con el cliente”*.

### 9.2.2 - Nuevos modelos de pricing para capturar el valor de la IA

Las mejoras de productividad que habilita la IA presentadas en 3.2 - IA y Productividad en el Desarrollo de Software, no se traducen automáticamente en una mayor rentabilidad para las consultoras. No es posible simplemente cobrar más por hora porque se es más productivo, ya que el uso de IA se ha convertido en un punto de paridad (Mixpanel, 2024): el cliente lo espera como parte del servicio, no como un valor adicional.

Además, al completarse los proyectos en menos tiempo, si no se modifican los esquemas comerciales, la única forma de mantener la facturación será vendiendo más proyectos en menos tiempo. Por eso, en un contexto donde la IA ya es un punto de paridad y no un diferencial competitivo por sí misma, es fundamental repensar cómo se cobra el trabajo realizado. Este concepto se introduce por Jose Gazzano cuando dice: *“Los tiempos de ejecución se acortan y los equipos son más pequeños. Si no repensamos la forma de cobrar, solo nos quedará vender más proyectos en menos tiempo”*.

El modelo tradicional de contratación, conocido como Time & Materials (T&M) (Adobe, n.d.), implica que el cliente paga por el tiempo trabajado (horas o días) de cada persona asignada al proyecto. Este modelo es transparente y flexible, pero tiene una limitación importante: asocia el valor percibido al tiempo invertido, no al impacto generado.

Para capitalizar mejor las ventajas de la IA, muchas consultoras comienzan a explorar esquemas complementarios o alternativos que permiten vincular los ingresos con el valor entregado. Algunas ideas aplicables en contextos distintos son:

- Precios combinados por resultado: se acuerda un monto base por el servicio, y un adicional si se alcanza un objetivo específico que se definió con el cliente (por ejemplo, reducir los tickets de soporte o acelerar la incorporación de usuarios).
- Bonificaciones por impacto: si la solución entregada permite reducir costos, aumentar ventas o mejorar eficiencia en un área concreta, la consultora recibe una bonificación proporcional.
- Retainers mensuales para evolución de soluciones IA: en lugar de cerrar el proyecto al entregar un MVP, se establece una suscripción mensual que cubre mejoras continuas, ajustes de modelos y optimización de rendimiento de la IA implementada.

Estos esquemas permiten que el cliente sienta que paga por un resultado concreto, y no solo por horas de trabajo. Además, aumentan el margen cuando se trabaja de forma más eficiente gracias a IA, ya que el ingreso no está limitado por el tiempo consumido.

Un segundo punto clave en este nuevo contexto es la posibilidad de vender perfiles especializados a un precio diferencial. Roles como *Data Scientists*, *Data Engineers* o *AI Developers* (desarrolladores con conocimientos en IA y uso de modelos como LLMs, RAG, embeddings, etc.) tienen alta demanda en el mercado y escasa oferta. Esto genera una oportunidad para la consultora: si demuestra especialización, puede cotizar estos perfiles a valores más altos, reflejando tanto su escasez como el impacto potencial que generan.

En resumen, adoptar IA no solo implica cambiar cómo se desarrolla, sino también cómo se presenta y cobra ese trabajo. Vender impacto en lugar de horas, y mostrar especialización real en IA, son caminos para capturar el valor generado y construir relaciones comerciales más sostenibles y rentables.

### 9.2.3 - Demostrar experiencia real en IA

La rápida adopción de soluciones basadas en inteligencia artificial en la industria queda en evidencia como se presenta en el capítulo 2.2.2 - Impacto de la IA generativa en la industria, donde el 65 % de las compañías ya la utiliza en al menos una función con IA. Esto ha elevado las expectativas de los

clientes: ya no basta con promesas o argumentaciones genéricas, sino que se requiere evidencia concreta de que la consultora domina la IA y la traduce en impacto de negocio. En un mercado exigente y altamente competitivo, demostrar expertise real constituye el factor decisivo para ganar confianza, justificar las tarifas profesionales y sostener relaciones de largo plazo.

Para convertir esa promesa en pruebas tangibles, la organización puede articular una cartera integrada que incluya:

- Casos de éxito con clientes documentados con métricas y testimonios verificables.
- Pruebas de concepto (PoC) internas que resuelvan problemáticas frecuentes y estén siempre listas para demostraciones.
- Uso interno sistemático de IA en procesos clave, desarrollo, ventas, operaciones, que evidencie adopción genuina y aprendizajes propios.
- Kit de ventas especializado (white papers, demos guiadas, playbooks de gobierno y ética) que traduzca la metodología de IA en un relato claro y replicable para el cliente.
- Participación y organización de eventos de IA donde expertos externos ofrezcan charlas o workshops, reforzando la posición de la consultora como referente en la materia.

En conjunto, estos frentes convierten el conocimiento en resultados visibles y posicionan a la empresa como un socio confiable, capaz de liderar proyectos de IA con credibilidad y consistencia. Los mismos representan algunos ejemplos, no siendo las únicas formas de mostrar conocimiento.

#### 9.2.4 - Alianzas estratégicas con proveedores de modelos y plataformas de infraestructura

Si bien esta tesis no exploró directamente el vínculo entre consultoras tecnológicas y proveedores de plataformas, el análisis general del mercado permite reconocer una tendencia clave: a medida que la inteligencia artificial se convierte en un componente estructural de los productos digitales, la relación con quienes ofrecen acceso a modelos, infraestructura y herramientas especializadas adquiere un valor estratégico creciente.

Por un lado, los principales desarrolladores de modelos fundacionales (LLMs), como OpenAI, Anthropic, entre otros, concentran el desarrollo de tecnologías que luego son integradas por miles de consultoras y empresas de software. A través de sus APIs, SDKs o acceso preferencial en fase beta, estas compañías determinan qué capacidades estarán disponibles primero, en qué condiciones y con qué costo.

Por otro lado, los hyperscalers como Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP) no sólo ofrecen la infraestructura necesaria para escalar estas soluciones, sino que también integran modelos directamente en sus ecosistemas, por ejemplo, Bedrock de AWS, Azure OpenAI Service o Vertex AI de Google, y brindan recursos de cofinanciación, soporte técnico y co-marketing para empresas que se suman a sus programas de partners. Además, existen proveedores clave en torno al manejo de datos y pipelines de machine learning, como Databricks o Snowflake, que ofrecen soluciones de orquestación de datos, entrenamiento distribuido y observabilidad de modelos.

Para una consultora tecnológica, establecer alianzas estratégicas con estos proveedores puede ser una ventaja relevante, tanto en términos técnicos como comerciales. El acceso anticipado a nuevas funcionalidades, los beneficios económicos por volumen o especialización, y el posicionamiento derivado de ser reconocido como “partner oficial” son elementos que fortalecen la propuesta de valor ante clientes exigentes. En el caso de algunos proveedores, incluso pueden acercar clientes a partners con relaciones más avanzadas.

Naturalmente, la decisión de con quién aliarse dependerá del tamaño de la consultora, su foco tecnológico y su estrategia de crecimiento. Empresas más pequeñas pueden optar por construir con modelos open source y mantenerse agnósticas de plataforma, priorizando flexibilidad y control de costes. Consultoras medianas y grandes pueden beneficiarse de programas de partners de cloud providers, o proveedores de manejo de datos y pipelines, accediendo a créditos de infraestructura, formación exclusiva o soporte de marketing. En todos los casos, evaluar y actualizar periódicamente estas alianzas debe ser parte de la estrategia general de IA de la organización.

Frente a un mercado en rápida evolución, donde la diferenciación no proviene solo del talento interno sino también de la capacidad de integrar tecnologías de frontera, las alianzas inteligentes con los actores del ecosistema IA pueden convertirse en un acelerador real de competitividad y escala.

#### 9.2.5 - Equipos pequeños con expertise en IA

Uno de los desafíos clave para las empresas de consultoría tecnológica en el contexto actual es mantener la competitividad en un mercado donde la inteligencia artificial ya no es opcional, sino una expectativa básica, como ya se presentó en 9.2.3 - Demostrar experiencia real en IA. En este escenario, repensar la conformación de los equipos de trabajo se vuelve central para seguir brindando soluciones de calidad, sostener márgenes saludables y responder con agilidad a las demandas del cliente. La adopción de equipos más chicos con expertise en IA es una de las formas más efectivas de lograrlo: equipos más pequeños, más autónomos y con mayor capacidad de adaptación, apalancados en herramientas de IA. Esto se desprende del análisis presentado en 8.1.3 - Tendencia a la reducción del tamaño de los equipos y 8.1.2 - Evolución estratégica del rol del desarrollador, donde los líderes y expertos en tecnología concluyen en la posibilidad de equipos más pequeños, y la importancia de competencias en IA.

Tal como se presenta en el capítulo 1.1.1 - El marco Scrum, Scrum sugiere equipos de entre 5 y 9 personas como estructura óptima para mantener la comunicación fluida y la entrega continua de valor. En el pasado, alcanzar esa eficiencia con menos personas implicaba asumir riesgos o sobrecargas. Sin embargo, gracias a los diferentes avances brindados por la IA que han sido descritos, hoy es posible mantener o incluso aumentar la productividad con menos integrantes. Como afirma Cecilia Mariani, *“con IA y componentes reutilizables, equipos más chicos logran la misma entrega: donde antes hablábamos de un scrum de 7 a 9 personas, podríamos ver equipos de 3 a 5 desarrolladores capaces de producir lo mismo”*. Esto no solo reduce el costo directo del proyecto, sino que también mejora los márgenes cuando se adoptan esquemas comerciales más flexibles, como se menciono anteriormente.

Otra de las recomendaciones estratégicas identificadas en esta tesis es la priorización de perfiles generalistas por sobre los especialistas, tema que se desarrollará con mayor profundidad más adelante. En el contexto de un equipo más chico esto se traduce en una mayor presencia de roles versátiles, capaces de operar en múltiples frentes del desarrollo. Por ejemplo, que los desarrolladores automaticen parte del QA y definan criterios de aceptación mediante copilotos, que los diseñadores generen flujos de usuario utilizando herramientas de vibe coding o prototipos low-fi sin depender de equipos técnicos, o que los QA Engineers incorporen scripting y automatización sin apoyo exclusivo de otros roles. Esta expansión de capacidades técnicas amplía la autonomía del pod y reduce los tiempos de entrega, sin comprometer la calidad. En esta línea, Juan Pablo Mazza, CEO y fundador de Effectus menciona que *“hoy la IA es gran aliada para generar tests automáticos y diseñar casos de prueba”*, mientras que Cecilia Mariani, CEO de Darwoft menciona que *“con automatización apoyada en IA, no necesitás veinte testers manuales”*.

En este modelo, también cobra importancia que todos los roles tengan cierto nivel de dominio en IA. En algunos casos, puede ser recomendable incluir un experto en IA dentro del equipo, encargado de guiar técnicamente la incorporación de IA, acompañar la experimentación y elevar la propuesta de

valor del equipo frente al cliente. Este tipo de perfil no solo mejora la calidad técnica, sino que también fortalece el posicionamiento estratégico de la consultora.

Adoptar la lógica de equipos chicos con expertise en IA no es simplemente una cuestión de estructura interna. Es una forma concreta de responder a los cambios del mercado, entregar soluciones más rápidas y robustas, y hacerlo con equipos que requieren menos coordinación y pueden adaptarse con más velocidad. Para las consultoras, representa una oportunidad para elevar la calidad del delivery, mejorar sus márgenes, y consolidar una propuesta de valor moderna y alineada con las expectativas reales de sus clientes.

#### 9.2.6 - Reducir los tiempos de entrega y salida a producción

Las expectativas de los clientes han comenzado a cambiar de forma acelerada en relación con los tiempos de desarrollo y despliegue de productos digitales. Ya no se espera únicamente que una consultora entregue software funcional, sino que lo haga en menos tiempo, con ciclos de iteración más cortos y mayor capacidad de liberar versiones tempranas y mejoras continuas. Esta presión se da tanto para productos nuevos (como MVPs) como para funcionalidades en plataformas en producción. Como se presento en el capítulo 6.1 - Resumen de entrevistas, Camila Moscatelli menciona: *“creo que hay un cambio de paradigma mental y expectativas... cosas que antes llevaban más tiempo, ahora tienen que llevar mucho menos”*. En esa misma línea, Cecilia Mariani afirma: *“Nuestro objetivo es acortar el ciclo de un MVP: hoy lo hacemos en tres meses; queremos ver si, combinando IA y componentes reutilizables, podemos reducirlo”*.

Para responder a esta nueva realidad, las consultoras deberán optimizar sus procesos operativos y adoptar tecnologías que les permitan acortar los tiempos de entrega sin comprometer la calidad. Esto incluye automatizar tareas repetitivas, fomentar la autonomía de los equipos, incorporar herramientas de inteligencia artificial desde etapas tempranas del desarrollo, y utilizar componentes reutilizables para evitar reconstruir soluciones desde cero. Al mismo tiempo, es clave adoptar métricas que midan el tiempo de ciclo completo y no solo los entregables funcionales. Acortar los tiempos de salida a producción no es únicamente una cuestión técnica: se ha convertido en una ventaja competitiva tangible, y en muchos casos, en una condición necesaria para sostener relaciones comerciales en un mercado cada vez más exigente.

#### 9.2.7 - Generalistas sobre especialistas

Las consultoras deberán priorizar perfiles generalistas, en el caso de los desarrolladores, que sean capaces de manejar varios lenguajes, por ejemplo dos como principales y otros que puedan abordar rápidamente con ayuda de asistentes de IA. Sumado a poder asumir actividades tradicionalmente asignadas a distintos roles. Esto incluye por ejemplo generar pruebas de calidad mediante herramientas inteligentes, crear diseños preliminares con IA o redactar historias de usuario asistidas por modelos generativos. La versatilidad técnica, combinada con la habilidad de emplear la IA para ampliar su alcance, permitirá a los equipos reaccionar con agilidad ante nuevos requisitos o cambios de tecnología.

Aunque seguirán existiendo muchos desafíos que demanden especialistas con profundidad técnica o en áreas en específico, los equipos se estructurarán alrededor de generalistas de diversas disciplinas, desarrolladores, diseñadores, product owners y otros perfiles, que cubran un espectro amplio de responsabilidades y reduzcan los tiempos de aprendizaje al incorporar nuevas herramientas. Este enfoque favorece la flexibilidad operativa y facilita la evolución constante de los servicios ofrecidos a los clientes.

Algunos testimonios de líderes de tecnología que respaldan esta recomendación y presentados en el 6.1 - Resumen de entrevistas son:

- Juan Pablo Mazza, CEO y fundador de Effectus: *“Mucha gente ya usa IA para QA, refactorización o generación de documentación... estamos todos en ese camino”*.
- Santiago Pérez, Director de Ingeniería en Rootstrap: *“La IA permite automatizar pruebas unitarias y funcionales, y empieza a tener un rol en diseño”*.
- Cecilia Mariani, CEO de Darwoft: *“Las herramientas ya permiten que personas sin perfil técnico profundo construyan funcionalidades básicas”*.

#### 9.2.8 - Roles especialistas en IA

Para ofrecer un servicio sólido y generar confianza en los clientes, y en línea con el posicionamiento como *AI Expert* y partner tecnológico recomendado en 9.2.1 - Posicionamiento “AI-Expert” y socio estratégico, resulta indispensable que las consultoras inviertan en contar con perfiles de especialización profunda en inteligencia artificial. Dentro del marco teórico, en el capítulo 2.3 - El auge de los agentes inteligentes y sistemas multiagente se profundizó en el auge de soluciones agentes y multiagentes, mientras que en el capítulo 2.4 - Uso de la Inteligencia Artificial en diferentes verticales de negocio, se profundizó en soluciones y casos de éxito en diferentes verticales aplicando IA. Para brindar dichas soluciones es indispensable contar con expertos dentro de disciplinas IA, entre ellos destacan Data Scientists y Data Engineers con sub-especialidades críticas: expertos en MLOps que diseñan y mantienen pipelines de despliegue continuo, profesionales de forecasting y optimización que ajustan modelos para maximizar impacto de negocio, especialistas en data compliance y seguridad que aseguran la gobernanza adecuada de los datos, arquitectos de soluciones que integran componentes de IA con infraestructuras híbridas, entre otros. Estos roles aportan el conocimiento avanzado necesario para garantizar la calidad técnica, mitigar riesgos regulatorios y escalar iniciativas de IA de forma sostenible, brindando el servicio adecuado a los clientes.

Además, los especialistas actúan como multiplicadores de conocimiento dentro de la organización. Desarrollan materiales de aprendizaje, documentan procesos y acompañan al resto del equipo en la aplicación práctica de técnicas de IA, asegurando que las buenas prácticas se propaguen rápidamente, y el conocimiento en IA se desarrolle transversalmente en toda la organización, como fue antes también recomendando. Su liderazgo técnico también permite ofrecer servicios diferenciales, como auditorías de modelos o afinamiento de sistemas predictivos, que posicionan a la consultora un paso adelante de competidores con un dominio superficial de la inteligencia artificial.

#### 9.2.9 - Arquetipo del desarrollador del futuro

La velocidad con la que la inteligencia artificial está transformando el desarrollo de software vuelve imprescindible que las organizaciones definan un arquetipo del desarrollador del futuro: una visión compartida sobre qué conocimientos, habilidades y actitudes serán necesarios para continuar generando valor en un entorno cada vez más asistido por IA. Este arquetipo permite a los líderes de las distintas áreas, junto con los equipos de recursos humanos, alinear sus decisiones estratégicas en torno al talento: desde cómo formar a los equipos, hasta cómo seleccionarlos, integrarlos y acompañar su evolución dentro de la organización. Esto se alinea con las voces de los líderes entrevistados, quienes anticipan una transformación profunda del rol del desarrollador hacia funciones de mayor abstracción, juicio estratégico y colaboración con herramientas inteligentes. Para esto, es recomendado visitar las competencias más mencionadas por líderes de tecnología en 8.2.1 - Reconfiguración del talento: seniority, competencias y responsabilidades, sin embargo, las mismas pueden variar dependiendo de la organización, sus objetivos, clientes y servicios brindados.

Contar con un arquetipo claro habilita trabajar de forma más coherente en distintos frentes de gestión del talento: actualizar planes de entrenamiento o inversión en formación según las nuevas habilidades requeridas, realizar contrataciones ponderando los aspectos más relevantes definidos por el arquetipo del rol, y diseñar procesos de onboarding que introduzcan desde el primer día las competencias clave que el nuevo contexto exige. Si bien este capítulo se centra en el perfil del desarrollador, la lógica puede extenderse a otros roles mencionados en la tesis, como Product Designer, Product Owner, Project Manager o QA, todos ellos impactados transversalmente por la adopción de inteligencia artificial.

#### 9.2.10 - Desarrollo de competencias clave para desarrolladores

Tal como se detalla en el capítulo 1.2 - Principales responsabilidades del desarrollador, el rol del desarrollador ya no se limita únicamente a escribir código, sino que implica comprender el negocio, colaborar con otros equipos, y tomar decisiones que impactan en el producto final. En ese marco, la incorporación de herramientas de inteligencia artificial no transforma esa definición, sino que acelera y acentúa un recorrido ya en marcha: el de un profesional con habilidades técnicas sólidas, pero también estratégicas y de colaboración. En línea con esta evolución, y con lo propuesto en el capítulo 9.2.9 - Arquetipo del desarrollador del futuro, se recomienda que cada empresa defina su propio arquetipo, como instrumento que oriente la formación, el reclutamiento y la evaluación de talento.

En cuanto a las principales competencias a desarrollar, como se vio en el capítulo 8.2.1 - Reconfiguración del talento: seniority, competencias y responsabilidades, los líderes de tecnología entrevistados coinciden en que ciertas competencias se vuelven prioritarias en un entorno atravesado por IA. Entre ellas, destacan: la comunicación efectiva, el entendimiento del negocio y visión de producto, la capacidad de supervisar y complementar a los sistemas inteligentes, el aprendizaje continuo y adaptabilidad, y la habilidad para diseñar prompts efectivos o articular soluciones basadas en IA.

Como recomendación final, se sugiere que cada consultora defina un mapa de competencias clave alineado a su arquetipo del desarrollador, que contemple diferentes niveles de dominio, acciones concretas de formación (como talleres, simulaciones o coaching especializado), e indicadores que permitan medir su aplicación en proyectos reales. De este modo, se transforma el diagnóstico conceptual en una hoja de ruta práctica que potencie el desarrollo del talento en un contexto de cambio permanente.

#### 9.2.11 - Alfabetización en IA en toda la organización

La alfabetización organizacional en inteligencia artificial se define como la base común de conocimientos que permite a cualquier colaborador, sea técnico o no, comprender los principios, las limitaciones éticas y los casos de uso de la IA, interpretar sus resultados con juicio crítico y comunicarlos con claridad. El impacto transversal de la IA, analizado en el capítulo 2.2 - El boom de la IA generativa, y su creciente protagonismo en entornos corporativos como se vio en el capítulo 2.2.2 - Impacto de la IA generativa en la industria, son una clara muestra de la necesidad de extender esta alfabetización más allá del ámbito técnico.

Este proceso de formación debe incluir los conceptos fundamentales explicados en 2.1 - Conceptos clave en Inteligencia Artificial, como los distintos tipos de IA, el rol de los modelos fundacionales o las nociones de entrenamiento, prompting y sesgos algorítmicos. Incorporar este lenguaje compartido no solo facilita la colaboración entre áreas, sino que permite que perfiles no técnicos contribuyan activamente en la ideación, supervisión y evaluación de soluciones basadas en IA.

Desde el trabajo de campo, los testimonios recogidos en el capítulo 6.1 - Resumen de entrevistas refuerzan esta visión. Sebastián Melgar plantea que *“todos deberán adaptarse para seguir siendo relevantes: desarrolladores, diseñadores, project managers y QA incluidos”*, mientras que Juan Pablo Mazza remarca que *“no adoptar estas herramientas hoy es quedar fuera del ritmo de la industria”*. Estas ideas se alinean con los expertos citados en 7.3.1 - Resumen de voces de expertos, como Tim O’Reilly, quien observa que la creación de software se está volviendo una habilidad cada vez más horizontal, y Sam Altman, quien proyecta un involucramiento más amplio en el uso de IA, incluso fuera del rol de desarrollador. Así, el cruce entre teoría, entrevistas y bibliografía secundaria refuerza que una organización competitiva será aquella donde la comprensión de la IA esté distribuida en todos los equipos.

Como acción concreta, se recomienda que cada empresa desarrolle un plan continuo de alfabetización que incluya diagnósticos iniciales, contenidos básicos y aplicados, espacios de práctica entre pares, y embajadores internos que faciliten su adopción por área. Esta capacidad, más que un evento puntual, debe consolidarse como una competencia organizacional estratégica, que refuerce la implementación efectiva de cualquier iniciativa vinculada a IA.

## BIBLIOGRAFÍA

- JetBrains. (2025). *The State of Developer Ecosystem*. n.p.: JetBrains.
- Alcott, B. (2025). *Jevons' Paradox*. London: Earthscan.
- Scrum Guides. (2020). *Guía de Scrum: La guía definitiva de Scrum*. From The Scrum Guide: <https://scrumguides.org>
- Forsgren, N., Storey, M.-A., & Humble, J. (2021). *The SPACE of Developer Productivity*. n.p.: Microsoft Research.
- JetBrains. (2024). *Developer Ecosystem Survey 2024*. n.p.: JetBrains.
- Stack Overflow. (2024). *Developer Survey 2024*. From Stack Overflow: <https://survey.stackoverflow.co/2024/ai>
- Microsoft. (2023). *Autogen*. From GitHub: <https://github.com/microsoft/autogen>
- Paradis, E., Grey, K., Madison, Q., Nam, D., Macvean, A., Meimand, V., . . . Ferrari-Church, B. (2024). *How much does AI impact development speed? An enterprise-based randomized controlled trial*. From arXiv: <https://arxiv.org/abs/2410.12944>
- Bit2Brain. (2023). *Historia de la IA*. From Bit Brain: <https://bit2brain.com/historia-de-la-ia/>
- Statista. (2023). *Statista Market Insights: Elements & Methodology*. From Statista: <https://cdn.statcdn.com/static/img/outlook/methodology/methodology-en.pdf>
- Statista. (2024). *Software – Worldwide*. *Statista Market Insights*. From Statista: <https://www.statista.com/outlook/tmo/software/worldwide>
- Mordor Intelligence. (2024). *Global Software Market - Growth, Trends, Forecasts (2025 - 2030)*. From Mordor Intelligence: <https://www.mordorintelligence.com/industry-reports/global-software-market>
- Mordor Intelligence. (n.d.). *About us*. From Mordor Intelligence: <https://www.mordorintelligence.com/about>
- Bureau of Labor Statistics. (2024). *Software Developers, Quality Assurance Analysts, and Testers*. U.S. Department of Labor. From Bureau of Labor Statistics: <https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm>
- CompTIA. (2024). *Tech Employment Outlook*. CompTIA.
- O'Reilly, T. (2023). *The End of Programming as We Know It*. *O'Reilly Radar*. New York.
- Altman, S. (2025, Abril 2). *An Interview with OpenAI CEO Sam Altman About Building a Consumer Tech Company*. From Stratechery: <https://stratechery.com/2025/an-interview-with-openai-ceo-sam-altman-about-building-a-consumer-tech-company/>
- Sequoia Capital. (2024, Octubre 16). *Thomas Dohmke: Building GitHub Copilot and the future of software development [Podcast]*. From Sequoia Capital: <https://www.sequoiacap.com/podcast/training-data-thomas-dohmke/>
- The Economic Times. (2024, Octubre 31). *Jobs at risk? Sundar Pichai's comment raises alarm for software engineers*. From The Economic Times: <https://m.economictimes.com/news/international/global-trends/jobs-at-risk-sundar-pichais-this-comment-raises-alarm-for-software-engineers-google-ai-25-percent-new-code/articleshow/114806887.cms>
- Juan, I. D. (2024, Febrero 28). *CEO de NVIDIA dice que ya no es necesario saber programación y aconseja qué se debería estudiar*. From Infobae: <https://www.infobae.com/tecno/2024/02/28/ceo-de-nvidia-dice-que-ya-no-es-necesario-saber-programacion-y-aconseja-que-se-deberia-estudiar/>
- Infobae. (2023, Mayo 13). *El CEO de Google, Sundar Pichai, afirmó que la Inteligencia Artificial es uno de los cambios más profundos de la historia: "Afectará todo"*. From Infobae: <https://www.infobae.com/tecno/2023/05/13/el-ceo-de-google-sundar-pichai-afirmo-que-la-inteligencia-artificial-es-uno-de-los-cambios-mas-profundos-de-la-historia-afectara-todo/>
- Brynjolfsson, E., Li, D., & Raymond, L. (2023, Abril 23). *Generative AI at Work*. From arXiv: <https://arxiv.org/abs/2304.11771>

- Tech Gig. (2024, Abril 8). *Meta's AI can replace mid-level coders in 2025: Zuckerberg*. From Tech Gig: <https://content.techgig.com/technology/metas-ai-can-replace-mid-level-coders-in-2025-zuckerberg/articleshow/117235296.cms>
- Orosz, G. (2024). *The Pragmatic Engineer*. From The state of the tech engineering market in 2024: <https://newsletter.pragmaticengineer.com/p/state-of-eng-market-2024>
- Orosz, G. (2024). *ZIRP software engineers*. From The Pragmatic Engineer: <https://newsletter.pragmaticengineer.com/p/zirp-software-engineers>
- Generative Programmer. (2024). *AI Coding Assistants Landscape*. From Generative Programmer: <https://generativeprogrammer.com/p/ai-coding-assistants-landscape>
- Tooltester. (2024). *ChatGPT & Co. vs. Google: The AI Search Engine Showdown*. From Tooltester: <https://www.tooltester.com/en/blog/chatgpt-vs-google/>
- Bit Brain. (2023). *Historia de la IA*. From Bit Brain: <https://bit2brain.com/historia-de-la-ia/>
- GitHub. (n.d.). *GitHub Copilot*. From GitHub: <https://github.com/features/copilot>
- OpenAI. (2022). *ChatGPT: Overview*. From OpenAI: <https://openai.com/blog/chatgpt>
- Cursor. (n.d.). *Cursor – The AI-first Code Editor*. From Cursor: <https://www.cursor.com/>
- Windsurf. (n.d.). *Windsurf AI*. From Windsurf: <https://codeium.com/windsurf>
- Tabnine. (n.d.). *AI Assistant for Software Developers*. From Tabnine: <https://www.tabnine.com/>
- Anthropic. (2023). *Introducing Claude*. From Anthropic: <https://www.anthropic.com/index/introducing-claude>
- Meta AI. (2023). *LLaMA: Open and efficient foundation language models*. From Meta AI: <https://ai.meta.com/blog/llama-open-source-foundation-language-model/>
- Stability AI. (2022). *Stable Diffusion*. From Stability AI: <https://stability.ai/blog/stable-diffusion-public-release>
- DeepSeek AI. (2023). *DeepSeek-V2 Technical Report*. From DeepSeek AI: <https://github.com/deepseek-ai/DeepSeek-V2>
- McKinsey & Company. (2024). *El estado de la IA a principios de 2024: la adopción de la IA generativa aumenta y comienza a generar valor*. From McKinsey & Company: <https://www.mckinsey.com/locations/south-america/latam/hispanoamerica-en-potencia/el-estado-de-la-ia-a-principios-de-2024-la-adopcion-de-la-ia-generativa-aumenta-y-comienza-a-generar-valor/es-CL>
- Hernández, G. M. (2019). *Métricas de productividad para equipo de trabajo de desarrollo ágil de software: una revisión sistemática*. From Redalyc: <https://www.redalyc.org/journal/3442/344262226007/html/>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2023). *Attention is all you need*. From arXiv: <https://arxiv.org/pdf/2304.10557>
- OpenAI. (2021). *Creating images from text*. From OpenAI: <https://openai.com/dall-e>
- WiserNotify. (2024). *ChatGPT Users*. From WiserNotify: <https://wisernotify.com/blog/chatgpt-users/>
- Google. (2023). *Google*. From Bard (ahora Gemini): Alternativa basada en modelos de la familia PaLM y Gemini: <https://blog.google/technology/ai/introducing-bard-google-ai/>
- Google & Ipsos. (2023). *Our Life with AI: How people view and use artificial intelligence*. From Google Public Policy: [https://static.googleusercontent.com/media/publicpolicy.google/en//resources/our\\_life\\_with\\_ai\\_google\\_ipsos\\_report.pdf](https://static.googleusercontent.com/media/publicpolicy.google/en//resources/our_life_with_ai_google_ipsos_report.pdf)
- Wang, Q., Wang, B., Lu, Y., Li, Z., Xie, Z., Hu, J., . . . Liu, Y. (2024). *Multi-Agent Foundation Models*. From arXiv: <https://arxiv.org/abs/2403.07769>
- CrewAI. (2024). *CrewAI – Multi-agent orchestration framework for LLMs*. From CrewAI: <https://crewai.io/>
- LangChain. (2023). *LangChain documentation*. From LangChain: <https://docs.langchain.com>
- Abdul Rahman, H. S., Michael, P. T., & Lee, A. H. (2023). *Artificial Intelligence-Enabled Intelligent Assistant for Personalized and Adaptive Learning in Higher Education*. From Abdul Rahman, H. S.; Michael, P. T.; Lee, A. H.: <https://arxiv.org/abs/2309.10892>

- Sánchez, M. R., Torres, J. P., & Fernández, L. A. (2023). *Personalización del aprendizaje con inteligencia artificial en la educación superior*. From ResearchGate: [https://www.researchgate.net/publication/376577076\\_PERSONALIZACION\\_DEL\\_APRENDIZAJE\\_CON\\_INTELIGENCIA\\_ARTIFICIAL\\_EN\\_LA\\_EDUCACION\\_SUPERIOR](https://www.researchgate.net/publication/376577076_PERSONALIZACION_DEL_APRENDIZAJE_CON_INTELIGENCIA_ARTIFICIAL_EN_LA_EDUCACION_SUPERIOR)
- Anantrasirichai, A., Zhang, Y., & Bull, D. (2025). *Artificial Intelligence in Creative Industries: Advances Prior to 2025*. From arXiv: <https://arxiv.org/abs/2501.02725>
- Fernández, J., Liu, M., & Rossi, P. (2025). *Generative Artificial Intelligence in Creative Contexts: A Systematic Review*. From AI & Society: <https://link.springer.com/article/10.1007/s11301-025-00494-9>
- Smith, J., Wang, T., & Rodríguez, M. (2020). *Advancing from Predictive Maintenance to Intelligent Maintenance with AI and IIoT*. From arXiv: <https://arxiv.org/abs/2009.00351>
- Patel, R., Müller, L., & Zhang, K. (2024). *State-of-the-Art Review: The Use of Digital Twins to Support Artificial Intelligence-Guided Predictive Maintenance*. From arXiv: <https://arxiv.org/abs/2406.13117>
- GitHub. (2023). *Research: Quantifying GitHub Copilot's impact in the enterprise with Accenture*. From GitHub Blog: <https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-in-the-enterprise-with-accenture/>
- Amazon Web Services. (n.d.). *Amazon CodeWhisperer*. From Amazon Web Services: <https://aws.amazon.com/codewhisperer/>
- Kalliamvakou, E., Singer, L., DeVito, C., Allamanis, M., Kokkalis, N., & Bacchelli, A. (2024). *Using AI-Based Coding Assistants in Practice: State of Affairs, Perceptions, and Ways Forward*. From ResearchGate: <https://www.researchgate.net/publication/381372621>
- Cui, Z., Demirer, M., Jaffe, S., Musolff, L., Peng, S., & Salz, T. (2025). *The Effects of Generative AI on High-Skilled Work: Evidence from Three Field Experiments with Software Developers*. From SSRN: <https://ssrn.com/abstract=4945566>
- Coursera. (2023). *¿Qué es la transformación digital? Definición, estrategia y ejemplos*. From Coursera: <https://www.coursera.org/mx/articles/digital-transformation>
- IBM. (n.d.). *¿Qué es la Industria 4.0?* From IBM: <https://www.ibm.com/mx-es/topics/industry-4-0>
- Boston Consulting Group. (2022). *The rates of success, goals, and future priorities of digital transformations, by sector*. From Boston Consulting Group: <https://media-publications.bcg.com/BCG-IT-Services-the-Rates-of-Success-Goals-and-Future-Priorities-of-Digital-Transformations-by-Sector-April-2022.pdf>
- Whatfix. (2023). *Digital transformation & tech adoption by sector*. From Whatfix: <https://whatfix.com/blog/digital-transformation-by-sector/>
- Business Research Insights. (2024). *IT services market size & growth forecast (2023–2032)*. From Business Research Insights: <https://www.businessresearchinsights.com/es/market-reports/it-services-market-111832>
- Rootstrap. (n.d.). *Rootstrap*. From Rootstrap: <https://www.rootstrap.com/>
- Finmark. (2023). *Seed, Series A, B, C, D Funding: A Guide to Startup Funding Rounds*. From Finmark: <https://finmark.com/seed-series-a-b-c-d-funding/>
- Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Hoboken: Pearson.
- edX. (2024). *edX: Access 2500+ Online Courses from 140 Institutions*. From edX: <https://www.edx.org/>
- OpenAI. (2020, Abril 30). *Introducing Jukebox*. From OpenAI: <https://openai.com/research/jukebox>
- Google Research. (2023, Enero 27). *MusicLM: Generating Music From Text*. From Google Research: <https://google-research.github.io/seanet/musiclm/examples/>
- MidJourney. (2023). *MidJourney*. From MidJourney: <https://www.midjourney.com/>
- JetBrains. (n.d.). *IntelliJ IDEA: The Capable & Ergonomic Java IDE*. From JetBrains: <https://www.jetbrains.com/idea/>
- Microsoft. (n.d.). *Visual Studio Code – Code Editing. Redefined*. From Visual Studio Code: <https://code.visualstudio.com/>

- Atlassian. (2024). *Jira Software – Project and issue tracking*. From Atlassian: <https://www.atlassian.com/software/jira>
- Atlassian. (2024). *Trello – Manage Your Team’s Projects From Anywhere*. From Atlassian: Trello – Manage Your Team’s Projects From Anywhere
- JetBrains. (2024). *PyCharm: The Python IDE for Professional Developers*. From JetBrains: <https://www.jetbrains.com/pycharm/>
- JetBrains. (2010). *WebStorm: The Smartest JavaScript IDE*. From JetBrains: <https://www.jetbrains.com/webstorm/>
- Accenture. (2024). *Annual Report FY2024*. From Accenture: [https://investor.accenture.com/~/\\_media/Files/A/Accenture-IR-V3/home/accenture-fiscal-2024-annual-report.pdf](https://investor.accenture.com/~/_media/Files/A/Accenture-IR-V3/home/accenture-fiscal-2024-annual-report.pdf)
- Globant. (2024). *Annual Report 2024*. From Globant: <https://investors.globant.com/2025-02-28-Globant-Files-Annual-Report-for-2024>
- Hitachi. (2021, Julio 14). *Hitachi Completes Acquisition of GlobalLogic*. From Hitachi: <https://www.hitachi.com/New/cnews/month/2021/07/210714.html>
- Stock Analysis. (2024). *Perficient (PRFT) revenue and employees*. From Stock Analysis: <https://stockanalysis.com/stocks/prft/revenue/>
- Thoughtworks. (2023, Febrero 28). *Informe anual 2022*. From Thoughtworks Investor Relations: <https://investors.thoughtworks.com/static-files/5792afee-2258-4dc4-a606-ae10d9caf870>
- Rootstrap. (2022). *Emeritus – Engineering management for edtech scale-up*. From Rootstrap: <https://www.rootstrap.com/featured-cases/emeritus>
- Rootstrap. (2021). *MasterClass - Scaling product growth through staff augmentation*. From Rootstrap: <https://www.rootstrap.com/featured-cases/masterclass>
- Rootstrap. (2019). *Google – Scaling a custom CMS in weeks*. From Rootstrap: <https://www.rootstrap.com/case-studies/google-opus>
- Effectus Software. (n.d.). *Effectus Software*. From Effectus Software: <https://www.effectussoftware.com/>
- Neocoast. (n.d.). *Neocoast*. From Neocoast: <https://www.neocoast.com/>
- Darwoft. (n.d.). *Darwoft*. From Darwoft: <https://www.darwoft.com/>
- Seta Workshop . (n.d.). *Seta Workshop* . From Seta Workshop : <https://setaworkshop.com/>
- Condor Technologies. (n.d.). *Condor Technologies*. From Condor Technologies: <https://condortech.com.ar/>
- Mixpanel. (2024). *Puntos de paridad vs. puntos de diferencia*. From Mixpanel: <https://mixpanel.com/blog/points-of-parity-vs-points-of-difference/>
- Bureau of Labor Statistics. (2024). *Employment Projections: Methods Overview*. U.S. Department of Labor. From Bureau of Labor Statistics: <https://www.bls.gov/emp/documentation/projections-methods.htm>
- Benedict Evans. (2023). *Working with AI*. From Benedict Evans: <https://www.benedict-evans.com/benedictevans/2023/7/2/working-with-ai>
- Hadida, S. (2020). *La agilidad en las organizaciones: trabajo comparativo entre metodologías ágiles y de cascada en un contexto de ambigüedad y transformación digital*. From Universidad del CEMA: <https://www.econstor.eu/bitstream/10419/238381/1/756.pdf>
- Anshu, K., Anurag, Y., Naveen, G., & Ashok, S. (2023). An empirical study to analyze the adoption and implementation of Agile methodologies in software development. *Innovations in Systems and Software Engineering*, pp. 1–19. From Innovations in Systems and Software Engineering: <https://doi.org/10.1007/s13198-023-01958-5>
- Tahir, G., Zohreh, T., Halimah, Z., & Hamid, K. (2020). *The role of project manager in agile software teams: A systematic literature review*. From IEEE Access: <https://doi.org/10.1109/ACCESS.2020.3004450>
- Radu, F., Vladislav, S., & Dag I.K., S. (2023). *On the roles of software testers: An exploratory study*. From Journal of Systems and Software: <https://doi.org/10.1016/j.jss.2023.111742>

- Agile Business Consortium. (n.d.). *Roles and responsibilities – Team leader*. From Agile Business Consortium: <https://www.agilebusiness.org/dsdm-project-framework/roles-and-responsibilities.html>
- Anders, B., Marta K., L., Lone, N., Per A., N., & Jesper S., P. (2018). *The role of UX professionals in agile development: A case study from industry*. From ACM Digital Library: <https://doi.org/10.1145/3240167.3240213>
- Lucy Ellen, L., Tapio, K., Tomi, K., Teemu, S., Ville, H., Jari, I., & Casper, L. (2019). *DevOps in practice: A multiple case study of five companies*. From Information and Software Technology: <https://doi.org/10.1016/j.infsof.2019.06.010>
- Adobe. (n.d.). *Adobe*. From What is Time and Materials Contract: <https://www.adobe.com/acrobat/business/hub/what-is-time-and-materials-contract.html>

## Anexo 1 – Guía de preguntas para entrevistas

Este anexo contiene la guía de preguntas utilizada durante las entrevistas realizadas en el marco del presente estudio. La entrevista adoptó un formato semi-estructurado, lo que implica que las preguntas funcionaron como guía orientativa para la conversación.

No todas las preguntas fueron necesariamente abordadas en cada entrevista, y en algunos casos surgieron temas adicionales relevantes según el perfil del entrevistado y el desarrollo de la conversación. Además, algunas preguntas específicas fueron planteadas únicamente a líderes vinculados a empresas de servicios de tecnología o consultoría de software.

### Guía de preguntas

#### **Introducción breve del entrevistado**

- ¿Podrías contar brevemente sobre tu experiencia profesional en el sector tecnológico, especialmente en relación con el desarrollo de software?

#### **Impacto de la inteligencia artificial en la industria y diferentes roles**

- Desde tu perspectiva, ¿cuáles crees que son los principales cambios que está provocando la inteligencia artificial en el desarrollo de productos digitales actualmente?
- ¿En qué áreas específicas del desarrollo de productos digitales crees que la inteligencia artificial podría generar mayor impacto o automatización?
- Considerando todo el proceso de desarrollo de productos digitales, ¿qué roles podrían cambiar más significativamente, fusionarse, reducirse o incluso desaparecer como resultado del avance de la inteligencia artificial? ¿Observas potencial para que surjan roles completamente nuevos?
- ¿Qué roles están siendo más afectados y cuáles menos?
- ¿Observas alguno de estos cambios en los equipos o proyectos con los que trabajas hoy?

#### **Cambios esperados en el rol del desarrollador de software**

- ¿Cómo imaginas que cambiará el rol tradicional del desarrollador de software en los próximos 1, 3 y 5 años, considerando el avance actual de la inteligencia artificial?
- Pensando en las tareas habituales del desarrollo de software, ¿cuáles crees que pasarán a ser más relevantes o críticas, y cuáles serán menos relevantes o incluso obsoletas debido a la inteligencia artificial?
- ¿En qué medida crees que herramientas como asistentes de código (GitHub Copilot) o IDEs impulsados por IA (como Cursor) están generando cambios concretos en la forma en que trabaja el desarrollador de software hoy en día?
- ¿En qué medida crees que estas herramientas están cambiando la productividad de los desarrolladores?
- ¿Y en otros roles del equipo (por ejemplo QA, PM, diseño)?

#### **Competencias claves para el desarrollador del futuro**

- ¿Qué habilidades o competencias, tanto técnicas (duras) como personales (blandas), crees que serán clave para que los desarrolladores mantengan su relevancia profesional en este nuevo contexto?
- ¿Cómo debería evolucionar el perfil profesional de los desarrolladores para adaptarse al crecimiento del uso de IA generativa y asistentes de código?
- ¿Identificas algún desafío particular en la formación o capacitación de profesionales tecnológicos en este contexto?
- ¿Qué tipo de formación deberían fomentar las empresas de tecnología para preparar a sus equipos frente a este cambio?

#### **Impacto de la inteligencia artificial en la demanda de desarrollo de software**

- Considerando la industria tecnológica global, ¿cómo crees que la inteligencia artificial afectará al tamaño general del mercado o demanda global de servicios de desarrollo de software (1, 3, 5 y 10 años)? ¿Por qué?
- En cuanto a la demanda específica de profesionales técnicos (desarrolladores y otros roles relacionados al desarrollo de software), ¿cómo visualizas que la inteligencia artificial impactará en la cantidad y tipo de profesionales requeridos en el (1, 3, 5 y 10 años)?
- ¿Qué sectores o tipos específicos de desarrollo de software (por ejemplo, aplicaciones móviles, desarrollo web, software empresarial) crees que verán más afectada su demanda debido a la creciente adopción de la inteligencia artificial?
- ¿Qué industrias o sectores económicos consideras que podrían beneficiarse más del uso de IA y, por ende, impulsar más la demanda de servicios relacionados con inteligencia artificial y desarrollo tecnológico? (Ejemplos: entretenimiento, educación, salud, etc.)
- ¿Notas cambios en las expectativas de los clientes en cuanto a velocidad, costo o calidad del desarrollo a partir del uso de IA?

**Estrategias recomendadas para empresas del sector** *(solo para entrevistados de empresas de servicios de tecnología o consultoría)*

- Desde tu experiencia, ¿qué estrategias recomendarías a las empresas de desarrollo de software para posicionarse exitosamente en el mercado ante el avance de la inteligencia artificial?
- ¿Qué iniciativas específicas (en cuanto a oferta de servicios, capacitación del equipo o posicionamiento comercial) deberían implementar estas empresas para aprovechar las oportunidades que ofrece el contexto actual y futuro de la IA en el desarrollo tecnológico?
- ¿Qué empresas o tipos de empresas ves que están adaptándose mejor al contexto de IA? ¿Por qué?

**Cierre de la entrevista**

- ¿Hay algún otro aspecto relevante sobre este tema que no hayamos abordado y que consideres importante mencionar?

## Anexo 2 – Transcripción de entrevistas

En este anexo se presentan las transcripciones completas de las entrevistas realizadas a líderes de la industria tecnológica, en el marco del estudio sobre el impacto de la inteligencia artificial en el desarrollo de software y el rol del desarrollador. Los entrevistados ocupan posiciones de liderazgo en empresas de desarrollo de productos digitales, consultoría tecnológica o áreas vinculadas al talento técnico.

Las entrevistas fueron conducidas bajo un formato semi-estructurado, lo que permitió una conversación flexible orientada por una guía de temas previamente definida (ver Anexo 1 – Guía de preguntas para entrevistas). Las transcripciones fueron obtenidas mediante herramientas de transcripción automática y posteriormente editadas para mejorar su legibilidad. Algunas expresiones reflejan el lenguaje oral y espontáneo de los participantes.

### Transcripción entrevista a Camila Moscatelli

**Matias Mansilla:** Bien. Hola, Camila. Vamos a arrancar por una primer pregunta, te pido contarme brevemente sobre tu experiencia profesional en el sector de tecnología.

**Camila Moscatelli:** Bien. Hola, Matías. Bien, hago un recap a un nivel un poco más alto. Trabajo en el sector hace unos 10 años aproximadamente. Toda mi carrera profesional se fue desarrollando en la misma empresa en la que trabajo actualmente, donde arranqué como desarrolladora de software, trabajando principalmente en el desarrollo de productos mobile, con un foco bastante grande en el desarrollo de aplicaciones. A partir de ahí fui desarrollando mi carrera: empecé a asumir roles más vinculados a la gestión, pasando por posiciones de liderazgo de equipo y de personas. Hacía eso en paralelo con el trabajo en proyectos, y ya en los últimos tiempos me fui dedicando en su mayoría a la gestión y el liderazgo del departamento de ingeniería, en roles como Engineers Manager y Directora, que es el rol que ocupo actualmente.

**Matias Mansilla:** Perfecto, gracias. Ahora vamos a ver algunas preguntas más ligadas al impacto de la inteligencia artificial en la industria del software y en diferentes roles. Para arrancar, desde tu perspectiva, ¿cuáles son los principales cambios que está provocando la inteligencia artificial en la industria del software?

**Camila Moscatelli:** Bien. Creo que si pienso como el impacto más grande y los principales cambios que veo hoy en día, creo que hay como un cambio de paradigma mental y expectativas incluso en relación con lo que es el desarrollo de productos de software en general, donde hoy en día creo que hay también una expectativa como de mucha más agilidad en todo lo que es el mundo de software, y obviamente también de evitar tareas muy repetitivas aprovechando justamente lo que es el mundo de IA. Entonces creo que hay una expectativa y un cambio como en ese sentido en lo que es, cosas que antes llevan más tiempo, ahora tienen que llevar mucho menos, y es uno de los impactos más claros que veo en este momento, y que eso permite también justamente es llegar al mercado con cosas mucho más rápido que antes, en cierto punto, y creo que a nivel de lo que es el desarrollo de productos per sé, creo que eso ya venía pasando un poco igual, pero justamente todo mundo de IA creo que está poniendo mucho más todavía el foco en usuarios, en entender los usuarios, en entender la importancia ahora también de consumir mucho más data que antes y entender mucho más data que antes.

**Camila Moscatelli:** Lo que necesitan los usuarios y ese foco total en la personalización de lo que hacés para el usuario al que apuntás, va muy en línea con lo que es el desarrollo.

**Matias Mansilla:** ¿Creés que ese cambio que mencionás, la expectativa de ser más ágiles, de salir más rápido, de tener mayor foco en los usuarios, ya lo estás viendo que suceda en los proyectos que lidera Rootstrap, o es algo que ves más como una expectativa todavía?

**Camila Moscatelli:** Creo que es más una expectativa y un deseo, en cierto punto, pero no sé si es algo que ya esté pasando tanto en el mundo del software. Creo que todavía estamos en un periodo de entendimiento, de hasta dónde va esto y hasta dónde no va. Y también, como cierto temor, creo, en algunas áreas. Esto también trae un montón de nuevos desafíos a nivel de: si estás consumiendo datos de usuarios, ¿cómo lo estás haciendo éticamente?, ¿qué tanto estás considerando la seguridad? Creo que todavía hay un poco de tironeo entre extremos, en el sentido de hasta qué punto avanzar. En algunas circunstancias hay más resistencia, y en otras, una actitud más de “veamos con calma” en lugar de lanzarse directamente.

**Matias Mansilla:** ¿En...?

**Camila Moscatelli:** Sí, a todo. Y creo que eso puede estar impactando un poco en que todavía no se esté viendo tanto. Pero creo que es un camino hacia el que vamos igual.

**Matias Mansilla:** ¿En qué etapas del proceso de desarrollo de un producto digital o de software, desde que se arranca hasta que está en producción y se continúa trabajando, ves que hay un mayor impacto hoy en día?

**Camila Moscatelli:** Creo que, en cierta forma, hay impacto en todo el proceso. Tal vez, por cercanía, se ve un impacto muy grande en lo que es el proceso de desarrollo. Pero, pensando incluso en etapas iniciales, también hay impacto en la necesidad de entender el negocio y en la bajada de requerimientos, como para poder aprovechar herramientas de IA durante el delivery de un producto. En la etapa de desarrollo, puntualmente, creo que es donde veo el impacto más tangible hoy en día. Justamente en términos de agilidad, de traducción de esos requerimientos a código funcional. Y también veo impacto en el testing y la calidad de esos productos. Creo que, en ambos sentidos, hay un impacto grande. Aunque todavía estamos en un momento de aprendizaje sobre cómo usar la IA en estas etapas más tempranas.

**Camila Moscatelli:** Entonces, hay una mejora en agilidad, pero también existen riesgos a nivel de calidad. Estamos en ese proceso de aprendizaje sobre cómo balancear ambas cosas, en cierta medida. También veo mejoras importantes en lo que es el testing de código, en lo que se desarrolla. Creo que ahí también hay un impacto.

**Matias Mansilla:** Perfecto.

**Camila Moscatelli:** Un impacto claro.

**Matias Mansilla:** Entonces, en el ciclo, ¿hay más impacto en programación y en testing?

**Camila Moscatelli:** Hoy en día, sí.

**Matias Mansilla:** Bien. Relacionado con esto que estás mencionando: ¿qué roles del proceso de desarrollo creés que están siendo más afectados o están cambiando más drásticamente? ¿O si imaginás incluso roles que cambien totalmente, que se fusionen, que desaparezcan o nuevos roles que surjan?

**Camila Moscatelli:** No sé si soy tan drástica, pero sí creo que hay un nivel de adaptación por el que todos los roles que participan hoy en día en un proceso de desarrollo van a tener que pasar. Por ejemplo, creo que en el rol de desarrollador hay un impacto muy grande, porque tareas que antes

formaban parte del core del día a día, hoy pueden ser soportadas por herramientas de IA. Código que antes era más repetitivo o sencillo, hoy puede generarse automáticamente, y el desarrollador pasa cada vez más a un rol de supervisor, chequeador de código, y no tanto quien lo escribe línea por línea. Entonces, creo que en ese punto ya hay un impacto bastante claro en esos roles, y también creo que a futuro va a ser cada vez más grande. Pero para mí no se pierde la necesidad del desarrollador. Si bien hay una necesidad de adaptarse, seguís necesitando una persona que tenga un entendimiento profundo de lo que significa el proceso de desarrollo de software: lo que implica realmente armar una arquitectura que sea escalable, entender un problema y traducirlo en una solución técnica que soporte una necesidad de negocio y producto.

Hay algo ahí parecido a la diferencia entre un perfil junior y uno senior. Se necesita igual esa experiencia.

Entonces, yo creo que el rol sigue siendo necesario, pero se adapta. Tal vez esa parte más metodológica es la que pierde un poco de valor, y sube el valor de otras habilidades. Pero, al final del día, al menos por ahora, se sigue necesitando al desarrollador.

**Matias Mansilla:** Claro. Nuevamente te pregunto: ¿esto quiere decir que ya lo estás viendo que sucede fuertemente, o hay una mezcla de visión a futuro y expectativa?

**Camila Moscatelli:** Creo que es una mezcla. Hay algunas cosas que ya se empiezan a ver, sobre todo si querés entregar hoy un diferencial o un valor distinto. Necesitás poner el foco en otras cosas. Y justamente, como ya hoy en día tenemos herramientas que ayudan a agilizar la escritura de código, creo que es aún más necesario que nuestros developers tengan el foco puesto en asegurar que la solución que estamos entregando resuelva efectivamente la necesidad del negocio, y no tanto simplemente en escribir código. Y eso sí lo veo hoy en día.

**Matias Mansilla:** Perfecto. Vamos ahora a mirar un poco más hacia adelante. ¿Cómo imaginás el rol del desarrollador de acá a 1, 3, 5 o 10 años?

**Camila Moscatelli:** A largo plazo me cuesta muchísimo imaginarlo, porque incluso si me preguntás dentro de unos meses, puede que te dé otra respuesta. Pero voy a tratar de compartir lo que hoy en día imagino. En el corto plazo, no sé si espero un cambio tan grande en el rol del desarrollador, más allá de los puntos que venía mencionando. Lo que sí creo es que lo que antes tal vez se esperaba de un arquitecto o de un desarrollador muy senior, hoy empiezo a esperarlo cada vez más de la gran mayoría del equipo: esta capacidad de ser crítico con el código, de entender la solución que está construyendo, de vincular los objetivos de negocio y producto con la solución técnica que se está dando.

**Camila Moscatelli:** Ahí es donde veo el cambio, y en el corto plazo también el impacto. Y cuando digo corto plazo, me refiero al próximo año, incluso tal vez a los próximos tres años, donde ese impacto probablemente se irá intensificando gradualmente. Pero no sé si veo un cambio tan drástico. Lo que sí veo es una necesidad creciente de expertos en ciertas áreas, que vienen de la mano de los desafíos que trae la IA. Siempre hubo, por ejemplo, expertos en seguridad, pero ahora, con los nuevos desafíos asociados a la seguridad que aparecen, va a haber cada vez más necesidad de ese perfil para el desarrollo de soluciones. Lo mismo ocurre con el entendimiento y análisis de datos. Tal vez hay una necesidad creciente de expertise desde ese lugar. Lo veo como un cambio en el background del desarrollador: antes, ese background quizás estaba más

vinculado a un lenguaje de programación específico, y ahora tiende a orientarse más hacia estos otros aspectos.

**Matias Mansilla:** Claro. Me respondiste en cierta medida la próxima pregunta que te iba a hacer, que estaba relacionada con las tareas que imaginás como más importantes y las competencias clave del desarrollador.

Quizás podemos resumir tu respuesta en algunos títulos que ya fuiste mencionando, y si te queda alguno más por agregar, te escucho.

**Camila Moscatelli:** Ahí creo que ya mencioné bastantes habilidades, y en cierto punto, algunas que son más cercanas al negocio. Tal vez cuando hablaba del foco en producto y de esa mentalidad más parecida a la de un supervisor, y también la parte más técnica, de background técnico. Pero si pienso en *skills* que considero súper necesarias, diría que la agilidad para incorporar cosas nuevas es una habilidad base. El mundo de la IA es muy cambiante, y justamente esa adaptabilidad, esa agilidad para aprender y estar constantemente al día con los nuevos cambios, para mí es una necesidad básica para cualquiera que quiera mantenerse en esta industria hoy en día. Siempre lo fue, en cierta forma, pero creo que ahora lo es aún más, con la velocidad con la que ocurren los cambios en esta área. También pienso mucho en lo que es el manejo de clientes, y en cómo lograr traducir esas necesidades del cliente en requerimientos de negocio y producto. Y después, eventualmente, traducir eso a código, de forma cada vez menos manual.

**Matias Mansilla:** Perfecto. Me salgo un poco de la agenda, pero todo lo que venís mencionando, y vos misma lo dijiste, suena muchas veces a tareas que uno espera de un perfil senior o de un arquitecto. ¿Dónde ves al perfil junior?

**Camila Moscatelli:** Es una excelente pregunta. Creo que, en cierta forma, el junior puede seguir siendo alguien que maneje y conozca soluciones de IA, pero que aún necesita mucha supervisión para poder considerar su código como entregable. Justamente, uno de los puntos más polémicos de todo esto es cómo alguien junior va a aprender cuando muchas cosas ya vienen dadas por las herramientas. Antes, el camino natural era aprender haciendo: entendiendo el proceso de desarrollo de software, enfrentando desafíos, escribiendo código, remangándose y metiéndose en la práctica. Hoy en día, al tener muchas cosas resueltas por herramientas de IA, creo que el camino va a ser muy distinto para los developers juniors. Ese aprendizaje tal vez se va a dar más en el proceso de revisión por parte de alguien más, y no tanto en el ejercicio directo de tener que construir todo desde cero.

**Matias Mansilla:** Claro. ¿Y esto implica que imaginás una menor necesidad de perfiles juniors?

**Camila Moscatelli:** Puede ser, sí.

**Matias Mansilla:** Yendo más a herramientas existentes hoy en día como Copilot, IDEs como Cursor o Windsurf, ¿en qué medida creés que están afectando al desarrollador? Ya hablamos un poco de los cambios en el rol del desarrollador, pero ¿qué tanto impacto creés que están teniendo estas herramientas en la productividad? ¿Poco, algo, mucho? ¿Qué tanto están afectando hoy y cuál es tu perspectiva para los próximos meses o años?

**Camila Moscatelli:** Creo que todavía no están afectando tanto, pero principalmente por un tema de adopción general. No estamos aún en un nivel alto de adopción, pensando también en otras empresas y en

conversaciones que tengo con otros colegas del área. Todavía hay cierta resistencia por momentos, incluso, y creo que se le podría sacar muchísimo más provecho del que actualmente se está obteniendo. Entonces, el impacto se ve un poco limitado por eso. El efecto en productividad, por lo tanto, tampoco es tan claro todavía. Incluso, a nivel de expectativas del cliente, creo que si el uso de estas herramientas estuviera mucho más extendido, habría un nivel más alto de exigencia en términos de velocidad y de cómo se entrega el servicio en sí mismo.

**Camila Moscatelli:** Entonces, creo que todavía no estamos ahí a nivel industria. Si bien se ha empezado a usar muchísimo más, y hay mucho impulso y difusión en torno a los asistentes de código, creo que todavía no están tan adoptados de forma general. Igual, incluso con ese nivel de adopción parcial, creo que sí hay una mejora en los casos donde se usan. Y, a medida que se adopten más, se va a empezar a ver de forma generalizada un cambio en lo que es productividad.

Ahí es donde estas tareas más sencillas, que antes tal vez te llevaban mediodía o más, hoy en día pueden llevarte solo un par de horas. Y eso cambia muchísimo a nivel de productividad y de en qué se enfoca el tiempo que se gana, básicamente.

**Matias Mansilla:** En resumen, en el mediano plazo, visualizas un cambio de productividad grande?

**Camila Moscatelli:** Sí. Todavía veo poco impacto directo, pero creo que estamos empezando a ver algunos cambios. Se están usando estas herramientas, aunque aún de forma limitada, y por eso los efectos en productividad todavía no son del todo evidentes. De todas formas, creo que estamos en el camino correcto. También hay una mejora constante en esas herramientas. Si mirás cómo estaban hace cuatro meses, la diferencia es muy grande. A medida que sigan mejorando, creo que el impacto se va a dar de manera más natural.

**Matias Mansilla:** Perfecto. Bien, pasemos a hablar un poco más sobre las expectativas en torno a la demanda del rol del desarrollador y del software en general. Con todos estos cambios que se están dando, ¿creés que van a afectar positiva o negativamente la demanda de trabajo en el sector del software?

**Camila Moscatelli:** ¡Qué pregunta! En el corto plazo, diría que de forma positiva. Creo que va a haber más demanda, porque esto abre un espacio de innovación y creatividad en distintas industrias, que van a buscar incorporar soluciones con inteligencia artificial en sus sistemas para mejorar su eficiencia. Para ser más óptimos, para entender mejor y para ofrecer mejores soluciones a sus usuarios, creo que en el corto plazo eso va a generar un incremento en la necesidad de desarrolladores. A largo plazo, me cuesta tomar una postura firme. Puede ser que la demanda baje eventualmente. Va a depender mucho de hacia dónde evolucione todo esto. Pero sí creo que podría empezar a disminuir la necesidad de desarrolladores a medida que las herramientas de IA sigan creciendo en capacidad y en el nivel de soluciones que ofrecen. Tal vez empiecen a predominar roles más enfocados en la gestión del proyecto y en el análisis de la solución técnica, y no sea necesario tener tanta gente directamente “metiendo mano” en el código.

**Matias Mansilla:** Claro, o sea, visualizás un escenario donde, por un lado, con menos personas se puede hacer más gracias a estas herramientas, pero al mismo tiempo hay un boom donde muchas más empresas quieren hacer cosas nuevas, lo cual empuja la demanda. Entonces, hay una especie de contrapeso entre eficiencia y expansión. ¿Creés que cuando ese boom se estabilice, va a pesar más la eficiencia?

**Camila Moscatelli:** Sí, exactamente.

**Matias Mansilla:** Claro. ¿Imaginas que algunos roles van a tener más demanda y otros menos?

**Camila Moscatelli:** Sí, creo que en línea con lo que venía diciendo: los roles más cercanos al negocio y al producto van a sostener o incluso aumentar su demanda, porque se van a necesitar habilidades más relacionadas al análisis de datos y a la capacidad de traducir esos datos en soluciones técnicas. También el rol del desarrollador, aunque actualizado, creo que va a seguir siendo necesario. Pero más como un orquestador de soluciones, alguien que supervise o integre herramientas, no tanto que escriba todo el código desde cero.

En cambio, donde sí veo potencialmente una baja en la necesidad es en el rol de QA, sobre todo en el QA manual.

Estoy pensando en un escenario donde la IA pueda simular cada vez mejor el comportamiento de los usuarios. Incluso a nivel de testing funcional, podrían aparecer agentes que testeen las aplicaciones automáticamente.

En ese contexto, el rol de QA manual podría empezar a tener cada vez menos impacto y valor, al menos en los términos tradicionales en que lo entendemos hoy.

**Camila Moscatelli:** ...a nivel de necesidad, y cómo podría disminuir.

**Matias Mansilla:** Bien, perfecto. Ahora, pensando no tanto en la demanda de desarrolladores o profesionales, sino en el tamaño del mercado del software en general: ¿creés que ese mercado va a crecer más o menos en los próximos años?

**Camila Moscatelli:** ¿Cuando hablás de tamaño del mercado te referís a...?

**Matias Mansilla:** A la cantidad de software desarrollado en el mundo, y a la necesidad general de soluciones de software.

**Camila Moscatelli:** Ah, perfecto. Creo que esto abre puertas para muchas cosas nuevas que antes ni siquiera eran parte de la conversación. En ese sentido, sí veo un incremento en el desarrollo dentro de la industria, porque se están abriendo nuevas posibilidades. Hay cosas que antes ni siquiera pensábamos en resolver con software, y hoy en día sí es posible. Entonces, en ese sentido, sí esperaríamos que el mercado crezca.

**Matias Mansilla:** ¿Conocés alguna industria o vertical que creas que va a estar más impactada por los cambios que trae la inteligencia artificial?

**Camila Moscatelli:** Creo que todas, en mayor o menor medida. Pero hay algunas más claras. Por ejemplo, pienso en soluciones con IA que tienen un gran impacto en términos de automatización y eficiencia. Cualquier sector que tenga que ver con logística o con procesos va a estar muy impactado por estos avances.

**Matias Mansilla:** Bien, o sea, ¿ves un impacto más directo o transversal?

**Camila Moscatelli:** Creo que hay un impacto directísimo, pero también transversal. Incluso en industrias que no están directamente vinculadas a mi trabajo, he visto ejemplos, sobre todo en investigación, donde el impacto es muy grande. En el sector salud, por ejemplo, hay avances donde cosas que antes eran imposibles de imaginar sin soluciones de este tipo, hoy ya son realidad. Desde la detección de enfermedades hasta la generación

de mejores planes de salud, hay un mundo completamente nuevo abierto gracias a estas tecnologías. Ahí creo que hay un impacto claro y profundo, que realmente puede cambiar las cosas.

**Matias Mansilla:** Perfecto. Desde tu rol, tenés una visibilidad muy grande de las expectativas tanto de clientes actuales como de potenciales que llegan a Rootstrap. ¿Notas cambios en cuanto a las expectativas de velocidad, costos y tiempos por parte de estos clientes, en base a la inteligencia artificial y los cambios que está generando en el mundo del software?

**Camila Moscatelli:** Creo que todavía no es tan concreto. Similar a lo que mencionaba antes: como estas herramientas todavía no están tan adoptadas, y además hay cierto miedo por desconocimiento, no veo aún una expectativa generalizada en ese sentido. Todavía no está del todo instalada la idea de que la IA ya debería estar transformando la velocidad o el costo de los proyectos.

Antes, la presión de los clientes solía ser simplemente: “esto tiene que salir más rápido”. Ahora, sí creo que empiezo a ver una necesidad, por parte de algunos clientes, de incorporar algo de IA dentro de sus soluciones. Pero no necesariamente con la expectativa de que eso genere una mejora directa en la eficiencia o en la velocidad de entrega. Igualmente, creo que eso sí se va a dar cada vez más con el tiempo, aunque hoy todavía no lo veo tan claro en la práctica.

**Matias Mansilla:** Perfecto. Las preguntas que te voy a hacer ahora están orientadas a recomendaciones para empresas de consultoría de software, considerando que vos tenés un rol de dirección en una de ellas. La primera es: ¿qué estrategias recomendarías a empresas de desarrollo de software para posicionarse en este nuevo entorno en los próximos años?

**Camila Moscatelli:** No voy a revelar todos los secretos del éxito... ¡pero puedo dar algunas recomendaciones!

Creo que hay una necesidad clara de empezar a incorporar la IA en todo lo que hace una empresa de servicios hoy en día, para no quedarse fuera, básicamente. Y eso implica tanto usarla para beneficio interno, para ganar eficiencia operativa, como también asegurarse de tener un equipo preparado para asumir desafíos relacionados con IA. La IA no va a desaparecer, y parte de estar preparado para lo que viene es contar con un equipo realmente listo para eso.

Creo que una de las claves está en la capacitación. No solo en el uso de IA, sino en capacitar al equipo en general, no únicamente a los expertos, porque, como veníamos hablando, el rol del desarrollador en su totalidad está cambiando. Entonces, no se trata solo de tener personas especializadas en datos, en cloud o en IA per se, sino de capacitar a todo el equipo para que pueda incorporar estas herramientas en sus soluciones y en su día a día.

Ahí es donde creo que la capacitación se vuelve una pata muy importante.

Por otro lado, también mencionaría la necesidad de contar con una visión experta a nivel estratégico. Para poder tomar ciertas decisiones, e incluso para llegar a determinados clientes o comprender mejor ciertas problemáticas, es necesario tener personas que aporten esa mirada especializada. Alguien que pueda asegurar, por ejemplo, un nivel adecuado de confiabilidad en el uso de datos o en las soluciones de IA. Ese perfil experto también puede ser un referente clave dentro de la empresa, y estar al tanto de las tendencias en un mundo que cambia tan rápidamente.

Y finalmente, creo que otra cuestión estratégica a considerar es que no se trata solo de IA. Todo lo que rodea e implica a la IA también es importante: hablo de infraestructura, de cloud, datos, etc. Hay que entender el ecosistema completo.

**Camila Moscatelli:** Creo que ahí hay un punto muy importante a considerar dentro de una estrategia de posicionamiento. No puedes ofrecer soluciones de IA sin tener también un posicionamiento fuerte en otras áreas que las soportan. Entonces sí, creo que es necesario madurar esas otras partes, tanto del proceso como de los servicios que una empresa brinda para estar realmente preparada.

**Matias Mansilla:** Perfecto. ¿Conoces alguna empresa que creas que se está adaptando bien a este nuevo contexto?

**Camila Moscatelli:** Sí, por cercanía, creo que Qubika ha hecho un buen trabajo en ese sentido. Se ha posicionado de forma sólida en el mundo de datos, y ha venido creciendo muchísimo en su equipo de AI y Data en el último tiempo. Han hecho un buen trabajo tanto en posicionarse en el mercado y captar clientes, como en construir un equipo multidisciplinario que pueda enfrentar distintos desafíos dentro de este mundo.

**Matias Mansilla:** Buenísimo. Ya estamos cerrando lo que son las recomendaciones para consultoras. ¿Hay algún otro aspecto relevante que te gustaría mencionar antes de terminar la entrevista, relacionado con todo lo que venimos hablando?

**Camila Moscatelli:** No, simplemente que hay que subirse al tren. Esto no se va a frenar.

**Matias Mansilla:** Bien, perfecto. Muchísimas gracias, Camila.

**Camila Moscatelli:** ¡Un placer!

## Trascripción entrevista a Sebastian Melgar

**Matias Mansilla:** Buenas, Sebastián. Bienvenido. Para empezar, ¿podrías contarme brevemente a qué te dedicás y cuál es tu experiencia?

**Sebastian Melgar:** Gracias por la invitación, Mati. Soy fundador y actualmente CEO de Neocoast, una empresa que presta servicios de desarrollo de software para clientes, principalmente en Estados Unidos. Ofrecemos un servicio 360 que incluye desarrollo, diseño, project management, QA e inteligencia artificial.

**Matias Mansilla:** Perfecto. Ahora pasemos a la temática de la entrevista. La primera pregunta, más general, es: ¿cómo creés que la inteligencia artificial está impactando hoy el proceso de desarrollo de productos digitales?

**Sebastian Melgar:** Desde hace un par de años la IA está presente en prácticamente todas las conversaciones cuando arrancamos un proyecto. Entre 2016 y 2018 el gran foco era “mobile”; hoy casi no se menciona. El tema dominante es la IA, y veo dos tipos de necesidades: empresas con objetivos claros que buscan eficientar procesos internos o resolver un problema concreto del mercado y start-ups que persiguen financiación y necesitan sumar IA a su producto para presentarlo a VC o fondos de private equity. A estos clientes hay que guiarlos, explicarles dónde genera valor la IA y cómo integrarla coherentemente a su propuesta. Son, en definitiva, dos enfoques muy diferentes.

**Matias Mansilla:** Perfecto. Ya enfocándonos en los procesos y roles, ¿en qué etapas específicas del desarrollo de un producto digital, y sobre qué roles, creés que la IA está teniendo mayor impacto?

**Sebastian Melgar:** Buena pregunta. Hoy, dentro de IT, no veo un rol específico que esté siendo perjudicado por la IA; de hecho, la mayoría se beneficia gracias a las nuevas herramientas y a la mayor eficiencia que permiten. Donde sí hay impacto es en el plano financiero: un proyecto que antes requería cinco desarrolladores ahora puede abordarse con dos, lo que probablemente reduzca la demanda de talento y presione los precios a la baja en el corto y mediano plazo. Pensando hacia adelante, algunos puestos. por ejemplo, muchos roles de QA manual, podrían desaparecer si no evolucionan. En síntesis, hoy ningún rol se ve directamente afectado, pero todos deberán adaptarse para seguir siendo relevantes a mediano plazo: desarrolladores, diseñadores, project managers y QA incluidos.

**Matias Mansilla:** Perfecto. Para resumir: hoy no observás un impacto negativo, ¿verdad?

**Sebastian Melgar:** Exacto, lo veo como un impacto positivo.

**Matias Mansilla:** ¿Qué tan tangible es ese beneficio de las nuevas herramientas para los distintos roles?

**Sebastian Melgar:** Aún no lo medimos con métricas formales, pero la percepción interna es clara: ganamos eficiencia y calidad. El verdadero problema lo tendrán las empresas que prohíben el uso de estas herramientas—algo comparable a rechazar la maquinaria durante la Revolución Industrial. Quienes se resistan sentirán pronto un impacto negativo. En cambio, las organizaciones que adoptamos la IA ya vemos ventajas, aunque reconozco que a mediano plazo podría convertirse en una amenaza si no seguimos evolucionando.

**Matias Mansilla:** Perfecto. Sigamos por ese camino. ¿Cómo visualizás el rol del desarrollador a corto, mediano y largo plazo, digamos, en 1, 3 y 5 años?

**Sebastian Melgar:** De acá a un año no creo que cambie demasiado respecto a hoy, pero sí habrá un requisito excluyente: todo desarrollador deberá manejar herramientas de IA con soltura, al mismo nivel que se exige inglés. De hecho, ya estamos revisando nuestras pruebas técnicas para evaluar cómo usan la IA en lugar de prohibirla; sería incoherente prohibir algo que necesitan en el trabajo diario. En unos 3 años: Espero una disrupción mayor. Surgirán herramientas capaces de generar un MVP, diseño, web y app incluidos, de punta a punta. Hoy existen versiones beta aún inmaduras, pero en tres años veremos soluciones probadas y ampliamente adoptadas. Eso alterará las reglas: los clientes ya no llegarán con “hagamos un proof of concept” porque podrán hacerlo solos con IA. A 5 años y más allá los desarrolladores tendrán que ser cada vez más “generalistas” y capaces de tocar varios stacks, porque las herramientas de IA simplificarán lo repetitivo y lo muy especializado. Veo al talento migrando hacia sistemas legados complejos, por ejemplo, plataformas enterprise enormes, donde la sustitución completa por IA es inviable a corto plazo. En ese terreno, dominar múltiples tecnologías y saber integrar IA a procesos antiguos será clave. En síntesis, la IA no eliminará al desarrollador, pero obligará a ampliar su rango de habilidades y a centrarse en problemas que las herramientas no puedan resolver por sí solas.

**Matias Mansilla:** Claro.

**Sebastian Melgar:** Los desarrolladores van a tener que codear en .NET, PHP, o en la tecnología que sea.

**Matias Mansilla:** Justamente, esa es la siguiente pregunta: ¿cuáles creés que van a ser las tareas más relevantes en la vida de un desarrollador?

**Sebastian Melgar:** Es una pregunta difícil para mí porque ya no desarrollo en el día a día, entonces me cuesta imaginar cómo es exactamente el flujo actual. Pero lo que sí tengo claro es que el uso de herramientas de IA va a ser mucho más intensivo. Antes, cuando programaba, vivía entre Google y Stack Overflow; cada duda o error me llevaba ahí. Hoy, la IA es la nueva fuente de asistencia, y se va a usar para todo.

En cuanto a tareas, creo que el desarrollador va a seguir escribiendo código, pero con una diferencia clave: va a tener que hacerlo en múltiples lenguajes y tecnologías. El concepto de ser “experto en una sola stack” se va a diluir. Así como una persona puede hablar español e inglés en paralelo, los developers van a tener que ser capaces de trabajar con Python, PHP, .NET, React o Rails según el contexto, apoyados por IA que ayude a salvar las diferencias entre lenguajes.

**Matias Mansilla:** Y siguiendo esa línea, ¿cuáles creés que van a ser las competencias más importantes para los desarrolladores?

**Sebastian Melgar:** Sin dudas, una de las habilidades clave va a ser el manejo de prompts: saber interactuar con herramientas de IA de forma lógica y eficiente. Ya en nuestra época, el que mejor sabía formular una búsqueda en Google resolvía los problemas más rápido. Ahora pasa lo mismo, pero con IA: quien mejor sepa “prontear” va a obtener mejores resultados.

Además, creo que las soft skills van a volver a cobrar mucha relevancia. Las habilidades técnicas se van a estandarizar o resolver en parte con IA, pero la diferencia entre un buen profesional y uno promedio estará en lo humano: comunicación clara, capacidad de respuesta, empatía, iniciativa. Todo eso va a marcar la diferencia en un entorno donde lo técnico ya no será tan diferenciador como antes.

**Matias Mansilla:** Perfecto. Y siendo Neo una empresa que muchas veces forma talento, ¿qué desafíos ves hoy para capacitar desarrolladores en este nuevo mundo atravesado por la inteligencia artificial?

**Sebastian Melgar:** Es una gran pregunta. Cada vez es más difícil entrar al mercado —y ya lo estamos viendo. En el pasado, entre 2016 y 2020, la demanda de desarrolladores era tan alta que si alguien tenía un par de años de estudio lo contratábamos, lo formábamos internamente y en seis meses ya estaba trabajando en proyectos reales. Hoy la necesidad no es tan urgente, y eso impacta directamente en las oportunidades para quienes están comenzando.

En cuanto a la IA, en Neo ya estamos incorporando en el onboarding el uso de herramientas específicas de inteligencia artificial. Queremos que los nuevos talentos aprendan desde el inicio a apoyarse en estas tecnologías para ser más eficientes y rápidos. Es mucho más intuitivo aprender a usarlas desde el comienzo que tener que cambiar de paradigma después de años de trabajar de otra forma.

Igualmente, seguimos enseñando ciertas tecnologías que, a futuro, quizás pierdan protagonismo. Eso nos va a exigir atravesar una transición para que lo que formamos hoy no quede desalineado con lo que el mercado va a requerir en tres años.

**Matias Mansilla:** Genial. Ahora una pregunta más general. Si pensás en la industria del software en su conjunto, ¿cómo creés que va a evolucionar el tamaño del mercado y la demanda de software en los próximos 3 a 5 años? No me refiero al rol del desarrollador, sino al tamaño de la industria y sus oportunidades.

**Sebastian Melgar:** Buena pregunta, ahí me siento más cómodo porque es algo en lo que pienso todos los días. Justamente ayer tuvimos una jornada estratégica en Neo y hablamos sobre esto. Es fácil caer en una visión pesimista y pensar que todo va hacia abajo, pero a mí me gusta ver el vaso medio lleno y buscar cómo adaptarnos en un entorno que cambia constantemente.

A corto plazo, lo que vemos es que muchas empresas están lanzando MVPs generados con herramientas de low-code o IA, y aunque algunos funcionen bien de cara al usuario, muchos de esos productos están mal contruidos por dentro. Técnicamente son débiles. Entonces, cuando esas start-ups consigan una Serie A y necesiten escalar o sostener su producto, van a buscar ayuda profesional para reestructurar lo que armaron rápido y mal. Ahí vemos una demanda de reconstrucción en el corto plazo.

A mediano y largo plazo, creemos que el mercado de los MVPs tradicionales va a desaparecer casi por completo para empresas como la nuestra. Los proyectos iniciales se resolverán con IA sin necesidad de contratar a una agencia externa. Solo algunos casos concretos, clientes con mucho capital o escépticos de la IA, seguirán buscándolo. En cambio, el verdadero valor va a estar en ayudar a grandes empresas con sistemas legados complejos, que necesitan eficiencia y modernización.

Ese cambio va a generar un reposicionamiento del talento: los desarrolladores que hoy hacen MVPs van a migrar a proyectos más robustos, y muchas empresas grandes van a aprovechar ese talento nearshore para reducir costos. Incluso si no adoptan IA de forma masiva, van a decir: “¿Cómo hacemos más eficiente nuestro gasto?” y van a apostar más por outsourcing y nearshoring.

Veo algo parecido a lo que pasó hace años entre EE. UU. e India. Al principio, el trabajo remoto desde India se limitaba a tareas simples o fuera de horario, pero con el tiempo evolucionó hacia trabajos más lógicos y complejos. Hoy creo que está ocurriendo un fenómeno similar, pero a escala más global.

**Matias Mansilla:** Vos te enfocaste en una parte del mercado, los MVPs. Pero si pensás en toda la industria del software, ¿creés que su tamaño va a aumentar o disminuir?

**Sebastian Melgar:** Soy una persona optimista, así que eso puede jugarme en contra, pero sinceramente no creo que el mercado vaya a disminuir. Tal vez muchos dirían que sí, que va a caer, pero yo pienso que tanto las empresas como los negocios se van a reinventar. Hoy tener un software a medida sigue siendo un commodity caro, y eso va a cambiar. La IA va a permitir que más empresas, incluso pequeñas, puedan automatizar procesos y acceder a soluciones tecnológicas antes impensadas. Por eso, aunque el tipo de demanda va a cambiar, la industria en sí se va a mantener. No veo un crecimiento fuerte, pero tampoco una caída; más bien, una transformación.

**Matias Mansilla:** Perfecto. Y si volvemos al rol del desarrollador específicamente, ¿cómo ves la demanda futura de trabajo en desarrollo?

**Sebastian Melgar:** Creo que se va a mantener, pero los desarrolladores van a tener que adaptarse a nuevas tareas y formas de trabajar. Eso implica también repensar cómo se enseña. Hoy tenemos carreras de ingeniería en sistemas que duran siete años, con muchísima lógica y matemáticas, pero quizás en pocos años muchas de esas habilidades se resuelvan con IA. Tal vez se necesiten carreras más cortas, más aplicadas, que vayan directo al grano. Este cambio debería empezar desde las aulas y abarcar todo el sistema educativo. Hay que mirar hacia atrás para rediseñar la forma en que preparamos a los desarrolladores del futuro.

**Matias Mansilla:** Y desde el lado de tus clientes, ¿ya notás cambios en sus expectativas respecto a precios, tiempos o duración de los proyectos por el uso de inteligencia artificial?

**Sebastian Melgar:** Sí, aunque depende del cliente. Increíblemente, todavía hay quienes no quieren usar IA, y existen, aunque cueste creerlo. Pero en general, sí, hay una expectativa creciente de mayor eficiencia. No nos lo dicen directamente, pero lo vemos en la comparación de cotizaciones: muchos mandan el mismo proyecto a cinco o seis empresas, y a veces reciben presupuestos que cuestan la quinta parte de lo que nosotros estimamos. Eso genera presión.

Los clientes saben que las herramientas nuevas permiten hacer más con menos, y eso afecta sus expectativas sobre tiempo y costo. En nuestro caso, como no somos una empresa cara, seguimos siendo competitivos. Pero notamos que, si bien crecimos en experiencia, no pudimos aumentar nuestras tarifas, y eso ya es una forma de ajuste. Es probable que otras empresas más grandes sí hayan tenido que bajar sus precios. Nosotros, en cierto modo, lo hicimos de forma indirecta al aumentarlos.

**Matias Mansilla:** Perfecto, súper claro. Y para ir cerrando: desde tu rol como líder en una consultora de software, ¿qué recomendaciones le darías a otros líderes de la industria?

**Sebastian Melgar:** La clave es la adaptabilidad. En este negocio, y más aún en tecnología, hay que estar todos los días revisando qué cambió: si OpenAI lanzó una nueva versión, si una empresa china replicó un modelo, si apareció una nueva herramienta. Todo avanza rapidísimo y no podés quedarte un año haciendo lo mismo.

Además, como líderes, tenemos que identificar claramente cuál es nuestro valor agregado. Si no lo hacemos, nos volvemos intercambiables, un conjunto más de freelancers, y eso, tarde o temprano, también va a desaparecer. Hay que estar al día, diferenciarse, y abrazar plenamente la inteligencia artificial. Ser escéptico o resistirse a integrarla es, para mí, totalmente contraproducente.

**Matias Mansilla:** Perfecto. Y por último, ¿hay algún otro aspecto que te parezca relevante compartir sobre todo lo que estuvimos conversando?

**Sebastian Melgar:** Solo agregaría que este tipo de entrevistas están buenísimas, sobre todo cuando se hacen públicas y permiten escuchar otras miradas. Hablar con gente que ocupa distintos roles, con perspectivas distintas, es algo que valoro mucho. Siempre trato de mantenerme en contacto con otros profesionales, escuchar cómo ven las cosas y enriquecer así mi visión. El networking es fundamental: a veces alguien te dice algo que no habías pensado y te cambia el enfoque. En un mundo tan dinámico como este, eso puede marcar una gran diferencia. Así que, felicitaciones por la iniciativa.

## Trascripción entrevista a Juan Pablo Mazza

**Matias Mansilla:** Hola, Juan Pablo. Bienvenido y gracias por participar. Para comenzar, ¿podrías contarme brevemente tu experiencia en el mundo del software y a qué te dedicas hoy?

**Juan Pablo Mazza:** Claro. Empecé en la industria en 2014 como desarrollador en TopTier, una firma de servicios de software, donde trabajé dos años y medio enfocado en tecnologías front-end mobile. Después fundé Effectus Software, agencia en la que actualmente soy CTO.

**Matias Mansilla:** En pocas palabras, ¿qué hace Effectus y cuál es su tamaño?

**Juan Pablo Mazza:** Effectus es una agencia de diseño UX/UI y desarrollo de software. Somos unas 42 personas y trabajamos directamente para clientes en Estados Unidos.

**Matias Mansilla:** Perfecto. Ahora pasamos a las preguntas específicas sobre la temática. Empezamos con algo amplio: ¿cuál creés que es el impacto actual de la inteligencia artificial en el desarrollo de productos digitales?

**Juan Pablo Mazza:** Está teniendo un impacto muy tangible. En nuestro trabajo, la IA asiste en muchas tareas: testing, refactorización, documentación... todas se han visto fuertemente impactadas. En cuanto al desarrollo en sí, la IA ocupa un rol de copiloto, como bien lo refleja el nombre del producto de Microsoft, ayudando a validar ideas, buscar inspiración en soluciones y mejorar la eficiencia general.

**Matias Mansilla:** ¿En qué áreas específicas notás mayor impacto?

**Juan Pablo Mazza:** Principalmente en QA: para generar tests automáticos y diseñar casos de prueba, la IA es una gran aliada. También en la fase creativa, ya que permite diseñar arquitecturas o flujos conversando directamente con un agente de IA, reduciendo la necesidad de tantas interacciones humanas para validar ideas. Es una forma muy ágil de avanzar en etapas tempranas.

**Matias Mansilla:** ¿Y en cuanto a la codificación?

**Juan Pablo Mazza:** También. Podés resolver una parte significativa del trabajo, incluso la mitad, con un buen prompt. Si estás prototipando o resolviendo algo simple, es ideal. Ahora bien, en productos complejos o ya avanzados, la utilidad baja un poco porque hay mucho conocimiento tácito que es difícil transmitirle a la IA.

**Matias Mansilla:** Fuiste bordeando esta pregunta, pero si vamos específicamente a los roles que participan en el desarrollo de un producto digital, ¿cuáles creés que están siendo más afectados y cuáles menos?

**Juan Pablo Mazza:** El rol de QA es claramente uno de los más afectados. Muchas de sus tareas son automatizables, y eso genera un cambio fuerte. Igual, esas personas tienen muchas habilidades que pueden redirigir, reenfoándose en tareas que agregan más valor y dejando otras en manos de la IA. En cuanto al desarrollador, si bien ahora cuenta con más herramientas, hay un punto donde la programación se empieza a "cajanegrizar". Algunas soluciones requieren entender menos para obtener resultados, pero cuando surgen bugs complejos o situaciones imprevistas, ese conocimiento superficial no alcanza. Ahí es donde el buen ingeniero sigue siendo clave, por su capacidad técnica y de razonamiento profundo.

**Matias Mansilla:** ¿Muchas de estas cosas ya las ves en el día a día de tu empresa o son proyecciones a futuro?

**Juan Pablo Mazza:** Vemos de ambos casos. Por ejemplo, nuestros ingenieros ya están usando herramientas como Cursor para asistir su trabajo. También usamos IA al inicio de los sprints para diseñar casos de prueba y en la etapa de code review: primero hay una revisión automatizada y luego la pasa un revisor humano. Y en proyectos nuevos, como MVPs, usamos IA para generar una primera versión de código a partir de la firma de funciones o estructuras. Todo eso ya está pasando.

**Matias Mansilla:** Perfecto. Cambiando un poco el foco: si pensás en el rol del desarrollador, ¿cómo lo imaginás dentro de 1, 3 y 5 años?

**Juan Pablo Mazza:** Me imagino un perfil donde la principal habilidad será la adaptabilidad. Las herramientas cambian todo el tiempo, cada semana surge algo nuevo, así que la curiosidad y la capacidad de aprendizaje continuo van a ser esenciales. Eso también va a liberar espacio para desarrollar otras habilidades, como la comunicación, la comprensión del negocio y la colaboración. En definitiva, visualizo un desarrollador más integral, que combine lo técnico con lo humano y estratégico.

**Matias Mansilla:** ¿Y qué tareas del día a día creés que van a ganar relevancia y cuáles van a perder protagonismo?

**Juan Pablo Mazza:** Van a ser más relevantes las tareas relacionadas con el entendimiento del negocio del cliente: comprender la visión del producto, sus objetivos y los problemas reales que enfrenta. Eso le va a permitir al desarrollador proponer mejoras y cambios concretos. Muchas tareas técnicas van a estar asistidas por IA, así que el diferencial estará en tener una mentalidad más orientada al producto que al código en sí.

**Matias Mansilla:** Perfecto. Yendo específicamente a los asistentes de código o IDEs, como mencionaste Cursor, ¿qué impacto creés que están teniendo en el trabajo del desarrollador?

**Juan Pablo Mazza:** Hace poco hablábamos de eso en la oficina. Uno de los chicos comentaba que ya no siente que está programando tanto, sino que dedica más tiempo a pensar cómo escribir el prompt adecuado para que la herramienta programe por él. En ese sentido, el foco del desarrollador empieza a desplazarse: se invierte más tiempo en orquestar que en escribir código línea por línea. Es un trade-off: dedicás un tiempo a jugar con el prompt para ver si obtenés lo que necesitás, y si no, decidís cortar ahí y avanzar manualmente, escribiendo el código vos mismo.

**Matias Mansilla:** ¿Creés entonces en esta idea que se viene discutiendo en la industria del desarrollador como orquestador de herramientas?

**Juan Pablo Mazza:** Sí, 100%. Pero también creo que hay que ponerle un límite a esa orquestación. Está bien aprovechar estas herramientas, porque son claramente disruptivas, pero hay situaciones, especialmente en productos muy personalizados, donde el juicio de un experto sigue siendo indispensable. La IA todavía no puede reemplazar eso.

**Matias Mansilla:** Ya hablando de productividad, ¿en qué medida creés que estas herramientas, especialmente los IDEs, están cambiando la productividad de los desarrolladores hoy? ¿Y cómo pensás que va a evolucionar ese impacto?

**Juan Pablo Mazza:** Creo que estamos frente a una verdadera revolución. No me animo a dar un porcentaje porque no lo tengo medido, pero el impacto es enorme. Hoy, con herramientas de IA,

podés crear, modificar o eliminar componentes tan rápido que es posible tener un MVP funcional en un solo día, algo impensado hace poco tiempo.

Además, con plataformas que integran distintos sistemas, como el protocolo MCP, que permite que la IA se comunique entre Figma, el IDE y otras herramientas, ves resultados que realmente sorprenden. Se siguen sumando capacidades que incrementan la productividad de forma muy tangible.

**Matias Mansilla:** O sea que para vos la curva de productividad sigue creciendo, sin un techo claro.

**Juan Pablo Mazza:** Exacto. Si tuviera que adivinar, diría que la curva es exponencial, pero sinceramente, no sé dónde está el límite. Cada mes aparecen nuevas herramientas o mejoras que superan lo anterior.

**Matias Mansilla:** Vamos ahora a competencias. Ya fuiste mencionando algunas, pero si tuvieras que resumir: ¿cuáles son las competencias técnicas y blandas que debe tener un desarrollador hoy en un mundo atravesado por la IA? ¿Y cuáles creés que va a necesitar en el futuro?

**Juan Pablo Mazza:** Cada vez van a pesar más las competencias blandas, como adaptabilidad, flexibilidad y comunicación. Son claves para mantenerse relevante y atractivo para los clientes. Hoy hay una explosión de emprendedores no técnicos que buscan en los desarrolladores un partner que los acompañe, que les explique, que les traduzca ideas a soluciones.

En cuanto a las habilidades duras, están cambiando. Ya no es tan importante ser experto en un lenguaje particular, sino saber programar, comprender arquitectura, lógica y patrones. A partir de ahí, la IA te puede asistir para codificar en cualquier lenguaje. La clave es entender cómo funciona todo y usar bien las herramientas.

**Matias Mansilla:** Supongamos que hoy estás empezando como desarrollador junior, por ejemplo, en tu primer trabajo en Top Tier. ¿Cuáles serían tus primeros pasos?

**Juan Pablo Mazza:** Lo primero sería familiarizarme con las herramientas más básicas como ChatGPT. Empezar a construir el hábito de pensar con la IA, es decir, razonar y validar ideas con ella en lugar de hacerlo todo mentalmente. Después, probar asistentes como Cursor u otros IDEs con AI integrada. Muchas tienen versiones gratuitas, y cuanto antes te acostumbres, mejor. No adoptar estas herramientas hoy es quedar fuera del ritmo de la industria.

**Matias Mansilla:** Al principio mencionaste la importancia del juicio experto. ¿Ves un riesgo en que quienes están empezando no desarrollen ese pensamiento crítico? ¿Y cómo lo abordarías desde tu rol como líder?

**Juan Pablo Mazza:** Sí, es una preocupación real. Al priorizar tanto la productividad, se pierde parte del proceso reflexivo que antes generaba mucho aprendizaje. La velocidad actual puede impedir que los juniors profundicen. Por eso estamos buscando nuevas formas de aprendizaje.

Una de las iniciativas que implementamos son los tech round table meetings: cuando alguien del equipo descubre algo interesante, lo comparte en una reunión corta y lo debatimos. Eso nos da al menos un punto de partida para investigar más.

Notamos que había resistencia a dar charlas técnicas por la preparación y exposición pública que implican, así que optamos por formatos más íntimos, en grupos reducidos. Así logramos mayor participación y entusiasmo del equipo.

**Matias Mansilla:** Mirando hacia adelante, ¿cómo ves la evolución de la industria del software en su conjunto? Pensando en 1, 3 y 5 años: ¿creés que va a crecer, mantenerse o achicarse? ¿Y por qué?

**Juan Pablo Mazza:** En el corto y mediano plazo, creo que va a crecer, sobre todo por el auge de productos basados en inteligencia artificial. Es un campo en plena exploración, y hoy se busca aplicar IA a todo tipo de procesos: cadenas de suministro, sistemas internos, automatizaciones. Muchas de esas iniciativas funcionarán, otras no, pero el movimiento es muy fuerte.

Ahora bien, más a largo plazo, digamos dentro de 5 a 10 años, imagino que la demanda de proyectos simples va a disminuir, porque ya estarán cubiertos por soluciones preexistentes o automatizadas. La IA se irá enfocando en problemas más complejos y específicos, lo que hará que los productos de software que desarrollemos sean también más desafiantes y sofisticados que los de hoy.

**Matias Mansilla:** Hablamos de la demanda de software en general. ¿Y qué pensás sobre la demanda de desarrolladores o perfiles técnicos?

**Juan Pablo Mazza:** Creo que el perfil clásico del ingeniero muy técnico, más introvertido, va a perder peso. Lo que se va a buscar es un perfil más equilibrado, con habilidades blandas fuertes y una gran capacidad de aprendizaje. No va a desaparecer la demanda, pero va a estar enfocada en ese tipo de desarrollador.

**Matias Mansilla:** ¿Y en términos de cantidad? ¿Te imaginás más o menos desarrolladores en el futuro?

**Juan Pablo Mazza:** Me animaría a decir que menos. Incluso si la demanda de software crece, la mejora en productividad, gracias a IA y nuevas herramientas, va a hacer que se necesiten menos personas para hacer el mismo trabajo.

Leí hace poco algo que dijo el CEO de Shopify: que para contratar a alguien había que demostrar que ese puesto no podía ser reemplazado por IA. Tal vez esa no sea una práctica generalizada, pero creo que esa mentalidad se va a extender, especialmente entre grandes empresas. Así que no me imagino un aumento en la demanda de talento en términos de volumen.

**Matias Mansilla:** En resumen, ¿creés que la “carrera” entre la demanda de software y el aumento de productividad la gana esta última?

**Juan Pablo Mazza:** Sí, sin dudas. Con menos personas, se va a poder hacer más.

**Matias Mansilla:** ¿Y hay sectores o roles específicos donde veas que la demanda va a aumentar?

**Juan Pablo Mazza:** Sí. Ya lo estamos viendo en sectores como salud, entretenimiento y educación. Por ejemplo, en salud el diagnóstico asistido por IA avanza mucho, y en entretenimiento se genera contenido automáticamente todo el tiempo. Son áreas que se ven muy beneficiadas por esta revolución y que van a seguir demandando soluciones tecnológicas.

En cambio, sectores más manuales o tangibles, como la construcción, van a tener una adopción más limitada o más lenta. La revolución ahí no va a ser tan profunda.

**Matias Mansilla:** En la actualidad, ¿notás algún cambio en las expectativas de tus clientes en cuanto a costos, velocidad o tiempos de desarrollo?

**Juan Pablo Mazza:** Sí, definitivamente. Últimamente muchos leads nos preguntan cómo estamos adoptando inteligencia artificial y cómo eso impacta en nuestra productividad. Ya es una pregunta habitual en las conversaciones iniciales.

Muchos clientes, especialmente startups, esperan construir MVPs mucho más rápido gracias al uso de herramientas de IA y no-code. En general, buscan reducir tiempos y costos en las primeras etapas, y

luego, si el producto tiene tracción, están dispuestos a invertir más en futuras iteraciones. Pero la expectativa inicial es clara: lograr más con menos, y en menos tiempo.

**Matias Mansilla:** Para ir cerrando, como CTO de una empresa de consultoría en desarrollo de software, ¿qué estrategias recomendarías a otras empresas del sector para posicionarse exitosamente en este nuevo contexto?

**Juan Pablo Mazza:** Si tuviera que empezar de nuevo, trataría de enfocarme en un nicho específico, sobre todo por industria y no tanto por stack tecnológico. Hoy los servicios genéricos están siendo cubiertos por productos basados en inteligencia artificial, así que especializarse es clave.

Además, armaría una estrategia de comunicación y marketing centrada en IA, para capitalizar esta ola y posicionarte como experto o referente. Hoy, si una empresa de desarrollo no puede ofrecer servicios vinculados a IA, ya sea generarla, integrarla o consumirla de otro proveedor, está mal posicionada. La IA ya no es un diferencial, es una condición de base.

**Matias Mansilla:** ¿Y qué recomendaciones les darías a empresas que ya existen y tienen un tamaño similar al de la tuya?

**Juan Pablo Mazza:** Les diría que fomenten el hábito del uso de estas herramientas dentro del equipo, tanto en ingeniería como en los equipos de soporte. Hay que tener curiosidad, ganas de explorar y ser claros en la comunicación interna: esto es una revolución, y adaptarse ya no es opcional.

**Matias Mansilla:** ¿Hay alguna empresa que tomes como modelo en este aspecto?

**Juan Pablo Mazza:** No conozco tanto la interna de otras empresas, así que no podría mencionar una en particular. Lo que sí veo es que muchas personas, de distintas organizaciones, ya usan herramientas como Cursor, o la IA de Notion para documentación. En general, creo que la mayoría ya está aprovechando la IA para tareas como QA, refactorización o generación de documentación. Estamos todos en ese camino.

**Matias Mansilla:** Y para terminar, ¿hay algún otro aspecto relevante sobre la temática que te interese compartir?

**Juan Pablo Mazza:** Solo una reflexión final: muchas cosas que hoy todavía podemos entender y auditar, con el tiempo se van a volver caja negra. Eso nos obliga a repensar cómo vamos a seguir aprendiendo y entendiendo lo que construimos. Es un buen momento para reflexionar y buscar formas de no perder esa capacidad crítica.

**Matias Mansilla:** Excelente, muchas gracias Juan Pablo.

**Juan Pablo Mazza:** Por favor, un gusto.

## Trascripción entrevista a Santiago Perez

**Matias Mansilla:** Bienvenido, Santiago. Gracias por participar de la entrevista. Arranco con una primera pregunta: ¿podés contarme brevemente tu experiencia en el rubro? En pocas palabras, ¿desde cuándo trabajás en tecnología, qué estás haciendo actualmente y dónde?

**Santiago Perez:** Claro. Actualmente soy Director de Ingeniería en Rootstrap, rol que ocupó desde hace cinco años. Mi experiencia en software comenzó en 2012, cuando empecé trabajando en QA automatizado. Después pasé por roles de desarrollo de software durante algunos años, y luego me fui enfocando en liderazgo de equipos y posiciones más orientadas a management.

**Matias Mansilla:** Perfecto. Y en tu rol actual como Director de Ingeniería, ¿cuántas personas liderás y cuáles son tus principales responsabilidades?

**Santiago Perez:** Hoy lidero un equipo de aproximadamente 10 personas, compuesto por managers, tech leads, team leads y principal engineers. En cuanto a responsabilidades, junto al VP de Ingeniería y otros directores, trabajo en definir y ejecutar la estrategia general del área. Además, en un plano más táctico, me ocupo de la gestión directa de varias cuentas dentro de la empresa.

**Matias Mansilla:** Vamos con una primera pregunta más vinculada a la temática central. En el contexto del desarrollo de software y productos digitales, ¿cómo creés que está impactando la inteligencia artificial?

**Santiago Perez:** Creo que está impactando en todas las etapas del proceso, desde la definición del producto hasta la generación de código. Donde más se nota es justamente en la creación de código, pero también hay un impacto relevante en el testing. Hoy la IA permite generar especificaciones más detalladas, asistir en la producción de código con cierto grado de supervisión por parte del desarrollador, y automatizar pruebas, tanto unitarias como funcionales. También en etapas como diseño empieza a tener un rol importante. En resumen, creo que está generando valor y soporte a lo largo de todo el ciclo de desarrollo.

**Matias Mansilla:** Bien, perfecto. En esa línea, ¿creés que hay algunos roles que están siendo más impactados que otros?

**Santiago Perez:** Me parece que el impacto es significativo en todos los roles, pero tal vez donde más se nota, y donde puede haber cambios más marcados en el futuro, es en el área de QA. No creo que los roles de testing vayan a desaparecer, pero sí van a necesitar apoyarse mucho más en herramientas de inteligencia artificial. Todo lo que sea repetitivo va a poder automatizarse con mayor facilidad, sin requerir el mismo nivel de habilidades técnicas que antes. Desde el lado del desarrollo, la IA también ayuda bastante, especialmente en tareas repetitivas o de bajo alcance. Si bien impacta, no veo que esté en un punto tal que pueda reemplazar el rol del desarrollador. Más bien lo complementa y potencia, pero siempre dentro de ciertos límites.

**Matias Mansilla:** Claro, en parte ya respondiste la siguiente pregunta, pero quizás quieras complementarla. ¿Los cambios que mencionaste los estás viendo hoy en tu día a día o estás pensando más en un escenario futuro, ya sea cercano o a largo plazo?

**Santiago Perez:** No, ya los estamos viendo hoy. En el área de desarrollo, por ejemplo, venimos experimentando con asistentes de código y estamos obteniendo muy buenos resultados. Ese impacto ya se percibe en el día a día de los desarrolladores, y justamente por eso queremos seguir

profundizando su uso, porque lo consideramos productivo y positivo. Incluso en conversaciones recientes con clientes, vimos un cambio en sus expectativas: muchos están buscando reducir la inversión en testing manual y priorizar la contratación de perfiles que puedan aprovechar herramientas de automatización. En esos casos, la IA aplicada al testing ya se percibe como una forma clara de agregar valor.

**Matias Mansilla:** Perfecto. Cambiando un poco el enfoque, ¿cómo te imaginás el rol del desarrollador de acá a 1, 3 o 5 años?

**Santiago Perez:** Es difícil proyectarlo, porque la evolución de la inteligencia artificial en los últimos años ha sido tan acelerada que incluso pensar a tres años parece lejano. De todas formas, me imagino que la IA va a seguir siendo una herramienta central, y que saber usarla correctamente va a ser cada vez más importante para cualquier desarrollador. Lo que me cuesta es anticipar con precisión cómo va a evolucionar su impacto más a largo plazo.

**Matias Mansilla:** ¿Creés que algunas tareas del día a día del desarrollador van a volverse más relevantes o menos relevantes en ese futuro?

**Santiago Perez:** Sí, sin dudas. Creo que muchas tareas repetitivas o relativamente predecibles van a poder ser delegadas casi por completo a la inteligencia artificial. Eso va a hacer que esas tareas pierdan relevancia y también que se invierta menos tiempo en ellas. De todas formas, siempre va a ser necesario un control humano con conocimientos técnicos, porque no creo que llegemos a un punto de confianza total como para prescindir de la supervisión.

**Matias Mansilla:** ¿Y cuáles serían esas tareas más específicas que ves perdiendo relevancia?

**Santiago Perez:** Por ejemplo, el testing unitario o el armado básico de ciertas interfaces. Ya estamos bastante cerca de que la IA no solo genere interfaces que se vean bien, sino que estén técnicamente bien construidas y sean escalables. En el pasado se podían generar maquetas visualmente atractivas pero poco sostenibles. Hoy eso está mejorando, y creo que va a seguir evolucionando hacia una automatización cada vez más robusta. Son esas primeras capas del desarrollo, las más repetitivas o estructurales, las que veo con mayor potencial de ser delegadas a la IA en el corto plazo.

**Matias Mansilla:** Si vamos específicamente al mundo de los asistentes de código o IDEs con inteligencia artificial, ¿qué tanto creés que estas herramientas están cambiando el día a día del desarrollador?

**Santiago Perez:** Creo que ya pasaron a ser una herramienta más del día a día. Así como hace años se volvió impensable programar sin un IDE y dejar de usar un simple editor de texto, hoy tener un IDE con inteligencia artificial empieza a ser el nuevo estándar. Es parte del “nuevo normal” en la forma de trabajar que se está consolidando.

**Matias Mansilla:** Y si hablamos de productividad, ¿cuánto creés que está cambiando con el uso de estas herramientas?

**Santiago Perez:** Mi visión es que depende mucho del contexto y del tipo de trabajo que tenga cada desarrollador. Nosotros internamente venimos haciendo pruebas y experimentos, y hemos visto mejoras interesantes en productividad. Cuando las tareas son repetitivas o bien definidas, el impacto es muy alto. En cambio, si se trata de

tareas complejas, con múltiples pasos o mucha toma de decisiones, el impacto es menor. Aun así, creo que siempre hay una mejora, aunque el nivel varíe según el tipo de trabajo.

**Matias Mansilla:** Entonces, por lo que me decís, esto no es algo que visualizás a futuro, sino que ya está ocurriendo. ¿Cómo creés que va a evolucionar esa curva de mejora en productividad? ¿Será lenta, moderada o acelerada?

**Santiago Perez:** Es difícil predecirlo con exactitud. No tengo una bola de cristal, pero sí veo que los avances no se están estancando; al contrario, se están perfeccionando. Sin embargo, hay aspectos que requieren más tiempo, no solo para que las tecnologías maduren, sino también para que las respuestas de las herramientas sean lo suficientemente rápidas como para no interrumpir el flujo de trabajo. Por ejemplo, si tengo que esperar varios minutos a que la herramienta me devuelva una solución, eso corta mi ritmo y mi forma de procesar lo que estoy construyendo. Entonces, aunque veo progreso, me cuesta imaginar una curva de mejora explosiva a corto plazo. Pero, insisto, no tengo certezas absolutas.

**Matias Mansilla:** Bien, perfecto. Pasando al tema de competencias: tanto las técnicas como las blandas, ¿cuáles creés que van a ser más importantes para los desarrolladores de acá en adelante, en este contexto de inteligencia artificial?

**Santiago Perez:** En cuanto a las competencias técnicas, sin dudas saber usar bien estas herramientas es clave. No solo utilizarlas, sino entender cuándo y cómo usarlas para realmente potenciar la productividad y aprovechar mejor el tiempo. Del lado de las competencias blandas, creo que van a cobrar todavía más relevancia. Este tipo de herramientas tienden a nivelar ciertas capacidades técnicas, por lo que habilidades como la comunicación, el trabajo en equipo, el liderazgo o la relación con clientes pueden convertirse en verdaderos diferenciales. En muchos contextos, van a marcar la diferencia entre un buen profesional y uno excelente.

**Matias Mansilla:** Perfecto. Pensando en alguien que recién está comenzando en el mundo del software, ¿qué le recomendarías hoy para prepararse en este nuevo contexto?

**Santiago Perez:** Le recomendaría empezar cuanto antes a familiarizarse con las herramientas de inteligencia artificial, entenderlas y aprender a usarlas a su favor. Creo que pueden ser una gran ayuda tanto para aprender como para explorar distintos caminos. Dicho eso, también considero fundamental mantener una formación más estructurada, ya sea formal o informal, fuera del ámbito de la IA. La inteligencia artificial no reemplaza la comprensión profunda de cómo funcionan las cosas. Puede ser tentador aceptar lo que una herramienta te da como válido sin entenderlo, pero es clave no usarla a ciegas. Comprender el trasfondo sigue siendo necesario para crecer como desarrollador.

**Matias Mansilla:** ¿Identificás desafíos para las empresas en lo que respecta a la captación y formación de talento?

**Santiago Perez:** Sí, especialmente en la formación. Como la evolución tecnológica es constante, es difícil determinar con claridad hacia dónde enfocar los esfuerzos formativos. Eso representa un gran desafío.

En cuanto a la captación, también hay dificultades, sobre todo en los procesos de hiring. Podés evaluar bien cómo una persona utiliza ciertas herramientas, pero eso no siempre refleja con precisión el nivel de conocimientos de base que tiene. Y ese desbalance puede complicar las decisiones de contratación.

**Matias Mansilla:** Perfecto. Pasemos ahora a una mirada más general. ¿Cómo creés que va a evolucionar la demanda de software en los próximos años? Otra forma de preguntar lo mismo: ¿pensás que la industria del software va a crecer o disminuir?

**Santiago Perez:** Es una buena pregunta. La verdad, no tengo del todo claro si el “tamaño de la torta” va a crecer o no. Lo que sí creo es que dentro de esa torta va a haber cambios significativos: en cómo se desarrollan los productos, en los enfoques y en los intereses de quienes participan. Pero no sé si eso se traducirá necesariamente en más demanda global de software.

**Matias Mansilla:** ¿Y si en lugar de enfocarte en el interés, pensás en términos de necesidades? Es decir, si ahora se puede desarrollar software que antes no era posible, ¿eso podría generar un aumento de la demanda?

**Santiago Perez:** Es una buena perspectiva, no lo había pensado así. Desde el interés, como te decía, no veo un cambio tan claro. Pero es cierto que la posibilidad de resolver problemas de nuevas formas podría generar nuevas oportunidades. Igual, sigo sin tenerlo del todo claro. Es una cuestión abierta para mí.

**Matias Mansilla:** Bien. Y en lo que respecta a la demanda de profesionales técnicos, por ejemplo desarrolladores, pensando a 3 o 5 años, ¿cómo creés que va a evolucionar?

**Santiago Perez:** Creo que la demanda puede volverse más específica. Es decir, orientada a profesionales que sepan trabajar con este tipo de herramientas, ya sea para generar datos, usarlos correctamente o sacar provecho de ciertos análisis. En cambio, los roles más fácilmente automatizables podrían volverse menos requeridos.

**Matias Mansilla:** Claro. ¿Y en términos netos, pensás que va a haber más o menos demanda de profesionales técnicos?

**Santiago Perez:** No me imagino un crecimiento en cantidad. Más bien visualizo un acomodamiento interno, donde lo que cambia es qué se necesita y para qué tareas, no necesariamente la cantidad total de personas.

**Matias Mansilla:** ¿Entonces lo ves más como una reconfiguración que como una disminución?

**Santiago Perez:** Sí, lo veo como una reconversión. La demanda se mantendría similar, pero se transformaría.

**Matias Mansilla:** ¿Y en qué te basás principalmente para sostener esa idea?

**Santiago Perez:** Sinceramente, en este momento me baso más en una intuición que en datos concretos. Pero mi razonamiento parte de lo que pasó en otras industrias con grandes cambios tecnológicos. Cuando hubo procesos de automatización, ciertos trabajos manuales desaparecieron, pero surgieron otros más técnicos, necesarios para operar, mantener o mejorar esas nuevas máquinas o procesos. No veo un motivo claro por el cual debería aumentar significativamente la cantidad total de personas requeridas, pero sí veo cómo la naturaleza de esos roles puede transformarse. Igual reconozco que puedo estar equivocado; son escenarios con muchas incertidumbres.

**Matias Mansilla:** Perfecto. Desde tu experiencia, ¿se te viene a la mente alguna vertical de negocio que pensás que va a estar más impactada por los cambios que está generando la inteligencia artificial, en términos de necesidad o demanda de software?

**Santiago Perez:** Imagino que las verticales que ya manejan una gran cantidad de datos serán las más impactadas, porque tienen una base sólida para generar mayor demanda. Pienso, por ejemplo, en finanzas o e-commerce, donde ya existen grandes volúmenes de datos que pueden utilizarse para generar insights, hacer predicciones o mejorar el forecasting. En contraste, hay sectores como la salud que también tienen muchos datos, pero enfrentan desafíos importantes en términos de privacidad, lo que hace más compleja su utilización.

**Matias Mansilla:** Perfecto. Y desde tu experiencia actual, ¿notás cambios en las expectativas de los clientes? Por ejemplo, en cuanto a tiempos, costos o velocidad, a partir del conocimiento de que hoy los desarrolladores están utilizando cada vez más herramientas de inteligencia artificial.

**Santiago Perez:** Sí, eso fue cambiando con el tiempo. Al principio había cierta precaución, especialmente de nuestro lado como empresa de servicios, por temas de privacidad. Pero con el tiempo empezó a haber una expectativa más clara de mayor rendimiento, e incluso preguntas sobre qué herramientas se están utilizando. A veces, eso lleva a una simplificación del trabajo que no siempre es justa, porque aunque algunas cosas parecen simples gracias a estas herramientas, el trabajo detrás sigue siendo complejo. Y sí, lo estoy viendo cada vez más con distintos clientes.

**Matias Mansilla:** ¿Y para vos, como líder de tecnología, es un desafío explicar esas diferencias a los clientes, o es algo trivial?

**Santiago Perez:** Es un desafío, sobre todo cuando del otro lado no hay una contraparte técnica. A veces el cliente prueba una herramienta y ve que puede lograr algo rápido, como un prototipo, y piensa que eso ya es un producto terminado. Ahí es donde se vuelve difícil explicar que ese resultado inicial no está listo para producción: no es escalable, no es seguro, ni está preparado para mantenerse en el tiempo. Entonces sí, cuando el interlocutor no es técnico, explicar ese diferencial puede ser complejo.

**Matias Mansilla:** Perfecto. Yendo al último bloque de preguntas: como líder de tecnología en una empresa de consultoría de software, ¿qué recomendaciones le harías a otras empresas o líderes del sector frente a los cambios que está generando la inteligencia artificial?

**Santiago Perez:** Además de contar con talento técnicamente sólido, creo que es clave que ese talento esté abierto al cambio y dispuesto a hacer las cosas de forma diferente. En este momento, experimentar es fundamental: probar nuevas herramientas, ver qué funciona y qué no, e ir construyendo conocimiento organizacional que pueda transformarse en una ventaja competitiva.

**Matias Mansilla:** ¿Y recomendarías alguna iniciativa específica más allá de esa experimentación?

**Santiago Perez:** Por ahora, lo principal para mí es justamente eso: experimentar. Es la acción más inmediata y valiosa en este contexto. Y, por supuesto, asegurarse de tener personas preparadas para aprovechar ese proceso.

**Matias Mansilla:** ¿Existe alguna empresa que consideres referente en este ámbito, que esté adaptándose de forma ejemplar al nuevo contexto de inteligencia artificial?

**Santiago Perez:** No tengo un ejemplo claro en mente. No se me ocurre ahora una empresa de consultoría que pueda tomar como modelo concreto.

**Matias Mansilla:** Perfecto. Y para cerrar: ¿hay algún otro aspecto relevante que te parezca importante mencionar sobre esta temática, algo que no hayamos tocado?

**Santiago Perez:** No, creo que abordamos los principales puntos. Estamos en un momento de muchísimos cambios, muy rápidos, y eso exige que también nos adaptemos con rapidez. Esa agilidad para adaptarse creo que va a marcar la diferencia.

**Matias Mansilla:** Perfecto, bien. Estamos listos. Muchas gracias por la participación.

## Trascripción entrevista a José Gazzano

**Matias Mansilla:** Buenas tardes, José. Para empezar, ¿podrías presentarte brevemente? ¿Qué hacés hoy y cuál es tu experiencia en el mundo del software?

**José Gazzano:** Claro. Soy José y llevo unos 13 años en la industria del software. Desde hace 8 años lidero una empresa de productos digitales y desarrollo a medida, combinando un rol de liderazgo tecnológico con funciones de dirección ejecutiva.

**Matias Mansilla:** Perfecto. Pasemos a la temática central. ¿Cómo creés que la inteligencia artificial está impactando hoy el desarrollo de productos digitales?

**José Gazzano:** Creo que estamos ante un cambio sin vuelta atrás, tanto en las dinámicas de trabajo diarias como en el resultado final. El mayor impacto lo veo en la alianza entre desarrolladores y herramientas de IA, que permite alcanzar objetivos que antes requerían más tiempo o procesos engorrosos. En definitiva, la IA hace que el desarrollo sea más ágil y sensiblemente más rápido.

**Matias Mansilla:** Perfecto. ¿Y en qué etapas específicas del desarrollo de productos digitales estás viendo mayor impacto de la inteligencia artificial?

**José Gazzano:** Sin dudas en las fases iniciales, cuando llega un nuevo proyecto o una idea que requiere un relevamiento técnico. La IA ha sido clave para acelerar la creación de pruebas de concepto y validar rápidamente si un producto es técnicamente viable. Antes, ese tipo de análisis solía evitarse porque requería mucho tiempo y era difícil que el cliente quisiera asumir ese costo. Hoy, con estas herramientas, validar si las piezas tecnológicas encajan es mucho más ágil. Luego viene el proceso de refinamiento, claro, pero para una POC ganamos muchísimo tiempo. También veo mucho valor en la etapa de testing. Incluso con poco esfuerzo, es posible generar una capa inicial de tests automatizados que cubra los aspectos básicos, algo que antes se descartaba fácilmente en proyectos con presupuestos acotados o necesidades de velocidad. Ahora, aunque haya que profundizar después, arrancar con una base sólida de tests lleva mucho menos tiempo. Diría que esos dos momentos, la prueba de concepto y el testing inicial, son donde el impacto de la IA es más inmediato y accesible, incluso sin entrar aún en estrategias más avanzadas como el prompt engineering.

**Matias Mansilla:** Perfecto. Y si lo analizás desde los distintos roles, desarrolladores, QA, diseñadores, ¿en cuáles estás viendo un mayor impacto?

**José Gazzano:** En los desarrolladores junior noto un cambio importante. Antes, muchos se trababan navegando por foros o sentían cierta timidez para preguntar. Hoy, con herramientas como los chats de IA, se sienten más cómodos para consultar y experimentar. Eso les da más velocidad. Sin embargo, también ocurre que algunos confían exclusivamente en el chat y pierden la costumbre de buscar en otras fuentes, lo cual puede afectar su autonomía. En QA, el impacto también es fuerte. Ya no es necesario codificar todos los test cases a mano, lo que agiliza mucho esa etapa del proceso.

**Matias Mansilla:** Hoy hay una discusión sobre quién se beneficia más con estas herramientas: ¿los juniors o los seniors? ¿Tenés una postura?

**José Gazzano:** Creo que hay dos formas de verlo. Si hablamos de tiempos de entrega y velocidad de desarrollo, sin dudas el junior se beneficia más. Antes tenía que buscar mucho hasta encontrar algo útil. Hoy pregunta en el chat y suele obtener una respuesta precisa de inmediato. En cambio, el senior,

que ya tiene claro lo que quiere hacer, también se ve beneficiado, pero en menor medida, porque la herramienta muchas veces le entrega exactamente lo que necesita. Ahora, desde la perspectiva del aprendizaje, es otra historia. El junior que depende solo del chat se pierde parte del proceso de razonamiento que antes era inevitable. Aquella experiencia de adaptar una solución que encontrabas en un foro, entenderla, modificarla... eso hoy se pierde. Aprender a pensar el problema sigue siendo esencial, y ahora recae mucho más en la motivación personal del junior profundizar y no quedarse solo con lo que le devuelve la IA.

**Matias Mansilla:** ¿Eso los llevó a modificar cómo contratan o capacitan a los juniors?

**José Gazzano:** Sí, definitivamente. Nos ha pasado ver juniors trabados en algo y decirnos: “El chat me dijo esto”, y no haber mirado la documentación oficial. Les insistimos mucho en que exploren otras fuentes, porque sabemos que el chat puede equivocarse o estar desactualizado. A veces les cuesta salir de esa zona de confort.

También ajustamos un poco el foco en nuestras contrataciones y capacitaciones. Siempre valoramos el pensamiento crítico y la capacidad de razonar y resolver problemas, pero ahora somos más enfáticos. Antes, un junior con capacidad de seguir una especificación y escribir código alcanzaba. Hoy, con estas herramientas, también tiene que hacerse preguntas, analizar, y buscar distintas soluciones. Esas habilidades blandas, de análisis y autonomía, se vuelven cada vez más necesarias, incluso desde los primeros niveles.

**Matias Mansilla:** ¿Y creés que es realista esperar que un junior adquiera esas competencias tan rápidamente?

**José Gazzano:** Depende mucho de la persona. En mi experiencia, influye cómo manejan la frustración, qué herramientas personales traen desde antes, incluso desde su formación o historia de vida. Las habilidades blandas suelen tomar más tiempo, pero si hay predisposición y apertura, se pueden trabajar. Obviamente, no es algo inmediato, pero en nuestro equipo vemos que muchos juniors van adoptando ese enfoque.

**Matias Mansilla:** Cambiando un poco el foco, ¿cómo imaginás el rol del desarrollador de acá a 1, 3 o 5 años? Es decir, en el corto, mediano y largo plazo.

**José Gazzano:** Creo que la evolución va a ser gradual, pero clara. Cada vez nos vamos a apoyar más en herramientas de inteligencia artificial, y eso va a llevarnos a enfocarnos menos en lo técnico-operativo del código y más en la resolución de problemas. A medida que estos agentes se vuelvan más robustos, probablemente los adoptemos como una nueva capa de abstracción, como sucedió antes cuando se pasó de Assembler a C, y luego a lenguajes más de alto nivel como Java. Si bien aún hay limitaciones y es necesario entender los lenguajes de programación para garantizar que todo funcione correctamente, veo un camino donde el desarrollador y la IA estén cada vez más integrados. No creo que vayamos a una sustitución completa, pero sí a estructuras con equipos más pequeños, porque se necesitará menos gente para lograr lo mismo. En ese sentido, el rol del desarrollador no desaparece, pero cambia su naturaleza.

**Matias Mansilla:** Excelente. ¿Y de las tareas que realiza hoy un desarrollador, cuáles pensás que van a ser las más relevantes en ese nuevo contexto?

**José Gazzano:** Probablemente ganen relevancia las tareas que requieren mayor comprensión del contexto general del proyecto, como la definición de arquitecturas, el diseño de soluciones y la evaluación de impacto. Todo lo que implique una visión más estratégica y holística del producto.

Las tareas más repetitivas, como codificar formularios o funciones simples, son las que más fácilmente se delegan a la IA. En cambio, analizar los objetivos del negocio, entender a todos los actores involucrados y tomar decisiones informadas van a ser habilidades cada vez más necesarias. Ahí es donde el desarrollador va a tener que enfocarse y aportar valor, más allá del código en sí.

**Matias Mansilla:** Perfecto. En relación con los IDEs con inteligencia artificial como Cursor o Windsor, o asistentes de código como Copilot, ¿qué impacto estás viendo hoy en la productividad de los desarrolladores? No necesariamente con métricas, más desde tu percepción.

**José Gazzano:** El mayor impacto lo veo en las etapas de prueba de concepto y prototipado rápido. Ahí estas herramientas permiten avanzar muy ágilmente. Sin embargo, en proyectos grandes, con una base de código extensa, como uno que tenemos desde 2019, los resultados no son tan buenos. Nuestros desarrolladores, por ejemplo, comentan que las sugerencias de Copilot no les sirven porque no tienen el contexto suficiente del sistema. De hecho, a veces prefieren desactivarlo porque más que ayudar, entorpece. Creo que estas herramientas funcionan mejor para tareas más aisladas o acotadas, como implementar una feature puntual. Pero cuando hablamos de refactorizaciones importantes o resolver bugs complejos en aplicaciones grandes, el valor vuelve a estar en la experiencia del desarrollador humano.

**Matias Mansilla:** Claro. Y ahora, pasando raya sobre algunas cosas que ya fuiste mencionando, ¿cuáles dirías que son hoy las competencias más valiosas para un desarrollador, tanto técnicas como blandas?

**José Gazzano:** Primero y principal, el pensamiento crítico. Siempre defendí la idea de que un ingeniero de software no debe definirse por el lenguaje que usa. En nuestra empresa tratamos de inculcar que no somos “desarrolladores React” o “Rails” o “Android”, sino ingenieros que usan herramientas distintas según el problema. Lo importante es saber tomar decisiones, cuestionar requerimientos, dar un paso atrás cuando algo no funciona, ver el panorama completo y buscar soluciones. Ese tipo de razonamiento es cada vez más valioso, especialmente en un contexto donde lo más automatizable ya lo hace la IA. También creo que habilidades como el diseño de arquitectura, la visión global del sistema y la capacidad de adaptación a distintas herramientas o lenguajes son diferenciales importantes. Son competencias que combinan la experiencia personal con la colectiva, y que marcan la diferencia. Y finalmente, algo que va a ser cada vez más relevante es la comunicación: saber expresar ideas claramente, colaborar en equipo, ser capaz de explicar y discutir soluciones. Todo lo que la IA aún no puede hacer de forma auténtica.

**Matias Mansilla:** Genial, gracias. Cambiando nuevamente el foco: pensando en el mercado del software, ¿cómo creés que va a evolucionar su tamaño en los próximos 3 a 5 años? ¿Va a crecer, mantenerse o achicarse?

**José Gazzano:** Esa es una de las grandes preguntas que todos nos estamos haciendo. Más allá del impacto de la inteligencia artificial, venimos de años con muchas incertidumbres en la industria. Personalmente, no me animo todavía a hacer una predicción clara. Lo que sí veo es que hay personas que hoy están logrando programar y construir productos sin tener formación técnica previa, algo impensado hace un par de años. Herramientas de IA les permiten crear soluciones básicas por su cuenta, y eso, lógicamente, puede reducir la demanda de desarrolladores para ciertas tareas. Pero por otro lado, cuando esos productos necesitan escalar o volverse más sólidos, esas mismas personas o empresas terminan buscando apoyo externo. Por eso creo que la demanda de software en general va a continuar, aunque sí van a cambiar las expectativas: tanto en términos de costos como de tiempos.

Antes, desde Sudamérica, muchas veces nos enfrentábamos al típico “yo tengo un equipo en otro país que lo hace por diez veces menos”. Y nuestro diferencial era ofrecer mejor calidad, mayor solidez técnica, mejor código. Hoy, esos equipos más baratos también acceden a las mismas herramientas de inteligencia artificial, lo que los nivela parcialmente con nosotros. Eso nos obliga a encontrar nuevas formas de mantener una ventaja competitiva.

**Matias Mansilla:** Entiendo. Esa era una de las preguntas que venía después, pero aprovecho: ¿ya estás viendo ese cambio en clientes actuales, respecto a expectativas de gasto, tiempo o tamaño del equipo?

**José Gazzano:** Todavía no me ha pasado enfrentarme a un caso directo donde se compare nuestro equipo con uno potenciado por IA. Pero sinceramente, me parece algo inminente.

**Matias Mansilla:** Si enfocamos ahora en el rol del desarrollador, dejando de lado otros perfiles de la industria, ¿cómo ves la evolución de la demanda específicamente para desarrolladores?

**José Gazzano:** Creo que ya no vamos a ver el mismo nivel de contratación que antes. Hoy, con equipos más chicos, se logra mucho más. Las herramientas permiten avanzar más rápido, automatizar tareas, y eso naturalmente reduce la necesidad de sumar tantas personas.

**Matias Mansilla:** ¿Eso implica menos demanda de desarrolladores o puede haber otros factores que compensen?

**José Gazzano:** Es difícil decirlo. Por un lado, es cierto que los tiempos de ejecución se acortan y que los equipos son más pequeños. Eso puede reducir la necesidad de contratar. Pero al mismo tiempo, si los proyectos duran menos, tenés más tiempo para tomar otros, y eso puede compensar la reducción. Desde el punto de vista de mi empresa, por ejemplo, sí veo que la necesidad de sumar desarrolladores al equipo ha bajado. Pero del lado del cliente, la persona que busca una empresa de desarrollo, puede que la demanda siga estable o incluso baje un poco, justamente porque hoy cualquiera puede hacer una POC o un prototipo simple. Honestamente, no lo tengo del todo claro aún.

**Matias Mansilla:** Claro. Y para ir cerrando: como líder de una consultora de servicios de tecnología, ¿qué recomendaciones estratégicas le harías a otras empresas del rubro, en este contexto actual de transformación con inteligencia artificial?

**José Gazzano:** Creo que lo más importante es encontrar un diferencial que vaya más allá del resultado técnico o del costo. Hoy más que nunca, el mayor valor que puede ofrecer una empresa es posicionarse como un partner tecnológico real para sus clientes, generando una relación de acompañamiento humano. Eso es algo muy difícil de reemplazar con IA. También es fundamental no cerrarse a incorporar herramientas de inteligencia artificial. Es cierto que hay mucho ruido y muchas soluciones que parecen más una moda que algo útil. Pero algo de todo esto va a perdurar, y tenemos que estar preparados. Estoy convencido de que se trata de un cambio de paradigma, y como industria, debemos estar listos.

**Matias Mansilla:** ¿Conocés alguna empresa del rubro que consideres referente en este aspecto, que esté navegando bien este escenario?

**José Gazzano:** Es difícil dar un ejemplo claro. Han aparecido noticias de empresas grandes que afirman usar herramientas de IA en casi todo su desarrollo. Recuerdo que Shopify, por ejemplo, comentó que el uso de asistentes de código era obligatorio para sus empleados. Pero la verdad, no sé si ese es el camino ideal, imponer su uso a todos por igual. Creo que todavía

estamos en una etapa de descubrimiento, intentando entender cuál es el balance correcto, y por eso no tengo una empresa puntual para señalar como modelo.

**Matias Mansilla:** Bien. Y para cerrar, ¿hay algún otro aspecto sobre esta temática que te gustaría compartir, que no hayamos abordado?

**José Gazzano:** Sí, una reflexión final. Me parece importante tener cuidado con el exceso de marketing en torno a la inteligencia artificial. En el último tiempo, vimos cómo todo producto parecía tener “IA” solo para captar atención. Recuerdo un stand en TechCrunch el año pasado donde, al preguntar qué hacían con IA, la respuesta fue: “En realidad no hacemos nada, solo usamos Copilot para programar, pero decimos que está hecho con IA”. Ese tipo de cosas pueden terminar dañando la credibilidad del sector.

Ya vimos algo similar con el hype del blockchain, que terminó generando desconfianza. Por eso, creo que hay que avanzar con responsabilidad, no dejarse llevar por promesas exageradas, y no caer en ideas simplistas como “ya no voy a necesitar juniors porque tengo un generador de código”. El desafío es, justamente, encontrarle el verdadero valor a esta tecnología.

**Matias Mansilla:** Bien, perfecto. Muchas gracias, José.

## Trascripción entrevista a Cecilia Mariani

**Matias Mansilla:** Bienvenida, Cecilia. Para comenzar, ¿podrías contarme brevemente tu trayectoria en tecnología y qué haces hoy?

**Cecilia Mariani:** Actualmente soy CEO de Darwoft y dirijo la empresa a nivel general. Inicié mi carrera en el sector de telecomunicaciones en Argentina, trabajando en Telecom Personal y Globalstar. Más tarde pasé al desarrollo de software: me uní a Motorola en 2000, donde estuve diez años, ocupé casi todos los roles y descubrí mi vocación por la innovación al desempeñarme como Engineering Manager. Luego ingresé a Intel, donde consolidé esa formación y me acerqué a proyectos de hardware; allí logré lanzar dos productos al mercado, uno de ellos durante la fusión con McAfee. Esa experiencia me preparó para emprender: fundé Fucuro, luego Merable y finalmente Darwoft, la compañía que lidero hoy.

**Matias Mansilla:** Perfecto. Pasemos a la temática de hoy. En términos generales, ¿qué cambios está produciendo la inteligencia artificial en el desarrollo de productos digitales?

**Cecilia Mariani:** Lo primero que notamos es que los perfiles más jóvenes, recién salidos de la universidad o pasantes, ya llegan usando IA para programar. A veces no está claro si dominan los fundamentos, pero encuentran y generan código con mucha rapidez. En cambio, a los desarrolladores senior les cuesta un poco más adoptarla: ya saben programar y ven la IA casi como un “atajo” que les genera cierto recelo. Sin embargo, incluso los clientes que al principio prohibían estas herramientas terminaron pidiéndonos que todos los desarrolladores las usen para acelerar el delivery.

**Matias Mansilla:** Comentabas que los juniors adoptan la IA muy rápido. ¿Ves algún desafío particular con ese segmento?

**Cecilia Mariani:** Aún no medimos su impacto en productividad. Empezamos con un enfoque prudente: primero habilitamos las herramientas a los líderes para que las usaran y guiaran al equipo. Ahora las tiene todo el mundo. Los juniors las utilizan con soltura, pero todavía no sabemos si realmente elevan la productividad o sólo cambian la forma de trabajar. Nuestro objetivo es acortar el ciclo de un MVP: hoy lo hacemos en tres meses; queremos ver si, combinando IA y componentes reutilizables, podemos reducirlo.

**Matias Mansilla:** ¿En qué etapas del proceso ves más impacto, o potencial impacto, de la IA?

**Cecilia Mariani:** En Product discovery y UX: nos interesa mucho cómo la IA puede acelerar la investigación de usuario y la generación de experiencias conversacionales. El cambio en la forma de buscar, de Google al chat, va a modificar la interacción y, seguramente, el SEO de los productos web. En prototipado y desarrollo: las herramientas ya permiten generar código más rápido. Esto abre la puerta a que personas sin un perfil técnico profundo puedan construir funcionalidades básicas. En testing: con automatización apoyada en IA, se reduce la necesidad de grandes equipos manuales.

**Matias Mansilla:** ¿Creés que esto llevará a una industria con menos ingenieros “puros” y más perfiles apoyados en herramientas?

**Cecilia Mariani:** No del todo. Siempre va a haber una distinción entre quienes diseñan la solución y quienes implementan. Necesitaremos menos manos para tareas repetitivas, pero los productos críticos seguirán requiriendo ingenieros capaces de pensar la lógica, la arquitectura y la experiencia global. Los roles senior, arquitectos, tech leads, no van a desaparecer; la reducción vendrá del lado

operativo, donde la IA ayuda o reemplaza parte del esfuerzo manual. Así como el testing manual se redujo con la automatización, veremos optimización en desarrollo puro, pero la capa de diseño y estrategia seguirá necesitando talento especializado.

**Matias Mansilla:** Entonces, ¿la barrera de entrada bajará para perfiles menos especializados, mientras se mantiene la necesidad de roles senior?

**Cecilia Mariani:** Exacto. Habrá más juniors apoyados por IA y menos proporción de seniors, pero seguirá siendo clave contar con arquitectos y líderes técnicos que entiendan el negocio y definan la solución. En QA ya es evidente: con automatización no necesitás veinte testers manuales. En desarrollo ocurrirá algo similar, aunque todavía no reducimos el número de desarrolladores; más bien estamos observando dónde la IA nos permitirá optimizar sin sacrificar calidad.

**Matias Mansilla:** Justamente esa era mi siguiente pregunta. En el proceso de desarrollo de productos digitales, ¿qué roles ves más afectados, para bien o para mal, por la inteligencia artificial?

**Cecilia Mariani:** El impacto más claro lo observo en QA. Hace años, la proporción podía ser de hasta dos testers por cada desarrollador. Con la automatización, y ahora con IA, esa relación cambió drásticamente. Todo lo repetitivo se puede automatizar, reduciendo la necesidad de grandes equipos manuales.

Otro punto es la reutilización de componentes. Durante mi etapa en Motorola ya intentábamos acelerar proyectos con librerías internas: “¿cuántas veces vas a crear un login desde cero?”. El problema era lograr que los equipos realmente reutilizaran ese código; todos querían rehacerlo. Hoy la IA facilita encontrar y adaptar componentes reutilizables, o incluso fragmentos de código completos, lo que vuelve más eficiente el trabajo de desarrollo.

**Matias Mansilla:** Centrémonos ahora en el rol del desarrollador. ¿Cómo imaginás que podría cambiar de aquí a 1, 3 y 5 años?

**Cecilia Mariani:** En el corto plazo el cambio principal será dónde se pone el foco. Las tareas mecánicas se irán automatizando y el desarrollador dedicará más tiempo a pensar: definir arquitecturas, diseñar flujos lógicos y resolver migraciones complejas. Esa faceta de análisis todavía recae en la persona; quizás a largo plazo la IA también sugiera arquitecturas completas, pero hoy seguimos necesitando criterio humano. Por eso creo que las carreras aceleradas deberían reforzar la formación en lógica y pensamiento crítico, para que los nuevos profesionales no se queden solo en “copiar código”.

**Matias Mansilla:** Cuando hablás de “pensar”, ¿incluye también comprender el negocio y el producto, además de la arquitectura técnica?

**Cecilia Mariani:** Totalmente. El verdadero valor aparece cuando el desarrollador entiende el producto y su contexto: por qué se pide una funcionalidad, cómo impacta en el usuario y cómo puede mejorarse. No se trata de ejecutar al pie de la letra un requisito, sino de proponer la mejor implementación.

Los clásicos memes de “lo que me pidieron vs. lo que entregué” reflejan un problema de entendimiento. Ese desafío sigue vigente: hay que capturar bien la necesidad y aportar soluciones que la eleven, no limitarse a replicarla. Ahí es donde el rol del desarrollador mantiene, y refuerza, su relevancia.

**Matias Mansilla:** Siguiendo esa línea, ¿qué tareas creés que serán más relevantes para un desarrollador y cuáles perderán importancia?

**Cecilia Mariani:** Ganarán peso las tareas que exigen pensamiento lógico y diseño: entender arquitecturas, captar requisitos, definir flujos y tomar decisiones técnicas bien fundamentadas. También será clave profundizar en experiencia de usuario: saber cómo la gente interactúa con el producto y cómo mejorar esa relación.

En contraste, las tareas puramente repetitivas, como escribir código boilerplate, irán perdiendo importancia, porque la IA las automatiza con facilidad. El rol ya no será “solo programar”, sino pensar la solución completa.

**Matias Mansilla:** Además de las tareas, ¿cuáles dirías que son las tres competencias clave para el desarrollador en este contexto?

**Cecilia Mariani:** Lógica y pensamiento crítico: Capacidad para analizar problemas, evaluar alternativas y diseñar soluciones sólidas, independientemente del lenguaje o la tecnología. Comunicación y liderazgo: Explicar ideas, colaborar con distintos perfiles, resolver conflictos y, en general, entender el negocio detrás del software. En Darwoft, por ejemplo, ofrecemos cursos de oratoria, liderazgo y resolución de conflictos porque lo técnico ya no alcanza. Adaptabilidad tecnológica: La habilidad de cambiar de stack sin drama. Con IA, la barrera técnica baja; lo importante es aprender rápido, moverse entre tecnologías y aplicar la herramienta adecuada según el contexto. En síntesis, reducimos el peso del “saber X lenguaje” y reforzamos habilidades lógicas, comunicacionales y de negocio. Ahí es donde el desarrollador seguirá aportando mayor valor.

**Matias Mansilla:** Hablemos de demanda. Primero, la demanda de software en general: ¿cómo creés que evolucionará en los próximos 3-5 años?

**Cecilia Mariani:** En nuestro entorno vemos un crecimiento claro en ingeniería de datos y proyectos data-centric: desde la captura del dato y la visualización, hasta algoritmos predictivos. Hay mucho por hacer, y todavía poca automatización, así que ese segmento debería seguir expandiéndose.

**Matias Mansilla:** Si acotamos la pregunta al rol del desarrollador, ¿cómo imaginás la demanda de talento técnico?

**Cecilia Mariani:** A corto plazo no cambiará demasiado, pero en unos cinco años podría reducirse la contratación de perfiles junior. Con IA y componentes reutilizables, equipos más chicos logran la misma entrega: donde antes hablábamos de un scrum de 7-9 personas, podríamos ver equipos de 3-5 desarrolladores capaces de producir lo mismo.

**Matias Mansilla:** Con equipos más pequeños haciendo lo mismo, ¿cómo afecta eso a la “carrera” entre mercado de software y productividad?

**Cecilia Mariani:** Es paradójico. Los jóvenes valoran más su tiempo libre y tienden a trabajar por objetivos. Si la IA les permite completar tareas antes, tal vez no dediquen esas horas extra a producir más, sino a vivir mejor. Eso pone en duda que la productividad neta de la empresa crezca tanto como se espera.

**Matias Mansilla:** ¿Creés que ese cambio de mentalidad generacional impactará la forma en que medimos eficiencia?

**Cecilia Mariani:** Seguramente. Antes la norma era “ocho horas en la oficina”. Hoy, si alguien entrega a tiempo, no está mal visto que use el resto del día para otra cosa. Nos falta ajustar cómo estimamos los tiempos y cómo comunicamos a los clientes que ciertos proyectos podrían hacerse más rápido. Estamos probándolo en un producto piloto antes de generalizarlo.

**Matias Mansilla:** En síntesis, veremos más demanda en data, equipos de desarrollo más pequeños y un cambio cultural hacia trabajo por objetivos y calidad de vida.

**Cecilia Mariani:** Exacto. El reto será equilibrar productividad y bienestar mientras la IA redefine nuestros procesos.

**Matias Mansilla:** Últimas dos preguntas. Primero: ¿ya notás cambios en las expectativas de tus clientes respecto a costos, tiempos o velocidad, a raíz del uso de inteligencia artificial?

**Cecilia Mariani:** Sí. Cada vez más clientes quieren acelerar el lanzamiento de sus MVP aprovechando IA y componentes reutilizables. Nuestro objetivo interno para este año es reducir el ciclo típico de un MVP, que hoy es de tres meses, acortando la fase de product discovery y UX, y usando IA para ensamblar más rápido los “assets” existentes.

**Matias Mansilla:** Y, como CEO de una consultora tecnológica, ¿qué recomendaciones estratégicas le harías a otras empresas del sector ante estos cambios?

**Cecilia Mariani:** Lo mismo que estamos haciendo en Darwoft: revisar cada proceso interno con mirada de eficiencia: pedimos a cada área que identifique cómo puede trabajar mejor usando IA o automatización, antes de solicitar nuevas contrataciones. Transformación digital puertas adentro: así como ayudamos a nuestros clientes, aplicamos esas mejoras a finanzas, administración, recruiting, etc. Prepararse para vender velocidad: sabemos que pronto llegarán competidores que prometan desarrollos más rápidos; queremos estar listos con procesos optimizados y herramientas en producción.

En resumen: reducir burocracia, automatizar lo repetitivo y reinvertir ese tiempo en valor diferencial.

**Matias Mansilla:** ¿Algún punto final que quieras destacar?

**Cecilia Mariani:** Solo subrayar la urgencia de adaptarse. He visto equipos universitarios sacar soluciones funcionales en tres días gracias a IA y APIs públicas. Si las consultoras no agilizamos nuestros procesos, corremos el riesgo de volvernos pesadas y lentas frente a esa nueva generación.

**Matias Mansilla:** Perfecto, Cecilia. ¡Muchas gracias por tu tiempo!

## Trascripción entrevista a Anthony Figueroa

**Matias Mansilla:** Bienvenido, Anthony. Para empezar, ¿podrías presentarte brevemente y contar tu trayectoria profesional en desarrollo de software?

**Anthony Figueroa:** Soy Anthony Figueroa, ingeniero en computación con más de 17 años en la industria. Empecé en una empresa que desarrollaba sistemas a medida para organismos estatales y, en 2011, cofundé **Rootstrap**. Desde entonces lidero equipos que conectan la parte técnica con las necesidades de negocio para clientes de diversas industrias en Estados Unidos.

**Matias Mansilla:** Genial, gracias. Empecemos por lo más general: ¿cómo creés que la inteligencia artificial está impactando hoy el desarrollo de productos digitales?

**Anthony Figueroa:** El primer gran cambio proviene de la popularidad de las herramientas generativas, ChatGPT, generadores de imágenes, etc. Eso aceleró la **conciencia de lo que es posible** con IA entre clientes, emprendedores y empresas. Como consecuencia, las expectativas sobre el desarrollo de software **augmentaron**: ahora se asume que ciertas soluciones deberían ser factibles, rápidas y accesibles.

Esta nueva percepción está generando y seguirá generando cambios: surgen ideas novedosas y se replantean problemas antiguos con enfoques más eficientes. Además, el usuario promedio llega mucho mejor informado sobre qué es fácil y qué es difícil. Antes, en Machine Learning, explicar esas diferencias requería una larga educación; hoy hay abundante información y ejemplos públicos que lo simplifican.

**Matias Mansilla:** ¿En qué áreas específicas del desarrollo de proyectos digitales creés que la inteligencia artificial está impactando, o impactará, más?

**Anthony Figueroa:** El impacto es **transversal**. La IA permite automatizar tareas tediosas a un costo mucho menor que antes, y eso se aplica prácticamente a todas las áreas: creatividad, seguridad, análisis, documentación, educación, onboarding... En definitiva, hablamos de una herramienta cognitiva que puede insertarse en cualquier parte de la organización.

**Matias Mansilla:** Hablando de los roles habituales en un proyecto digital, ¿cuáles están sufriendo mayores cambios? ¿Hay alguno que pueda desaparecer o fusionarse?

**Anthony Figueroa:** Todos los roles están cambiando y deben evolucionar con estas nuevas herramientas. No creo que un rol desaparezca por completo; más bien, el **conocimiento que generaba pasa a producirse de otra forma**. Ejemplo típico: el desarrollo. Hoy existen asistentes que generan código mucho más rápido que la escritura manual. Ya tuvimos abstracciones antes, librerías, frameworks; esto es otro paso. El desarrollador sigue siendo necesario para **analizar, depurar y decidir** el siguiente movimiento.

Lo mismo ocurre con otros perfiles: **Project Managers** pueden obtener resúmenes de estado y métricas en segundos, **Testers** encuentran más fácil la automatización, apoyándose en IA. **Product Managers** usan herramientas cognitivas para idear funciones y documentar de forma acelerada. Quien se niegue a adoptar estas capacidades perderá competitividad y, probablemente, quedará fuera del mercado. En 2025 sería **poco profesional no usar** algo que eleva tu efectividad.

**Matias Mansilla:** ¿Cómo imaginás el rol del desarrollador dentro de 1, 3 y 5 años?

**Anthony Figueroa:** Me gusta explicarlo diferenciando inteligencia y agencia. La inteligencia, resolver un problema de manera técnica, es justo lo que la automatización está absorbiendo. La agencia, en cambio, es la voluntad de actuar: decidir qué hacer, en qué orden y con qué prioridades. A medida que las herramientas cubran más la parte “inteligente”, el desarrollador deberá enfocarse en la parte de agencia: pensar en un nivel más alto, entender el producto, tomar decisiones estratégicas y distinguir una buena solución de una mala. Dejará que lo repetitivo lo haga la IA y ganará tiempo para aportar valor reflexionando y orientando el proyecto.

**Matias Mansilla:** Entonces, ¿qué tareas y competencias se volverán más relevantes para el desarrollador?

**Anthony Figueroa:** Las tareas centrales girarán en torno a definir la dirección técnica y de producto, evaluar alternativas y coordinar con otros roles para alinear objetivos. Las competencias clave serán la capacidad de ver el panorama completo y tomar decisiones informadas, la flexibilidad para moverse entre dominios en lugar de una especialización extrema y, por último, la habilidad de comunicarse y colaborar con los distintos actores del proyecto. En esencia, mientras los modelos de lenguaje dominan lo específico, el desarrollador deberá aportar criterio, contextualización y liderazgo.

**Matias Mansilla:** Perfecto. Yendo un poco más a lo que son IA IDEs o asistentes de código, ¿qué tanto creés que están impactando hoy en día en la productividad del desarrollador? ¿Y qué tanto creés que van a seguir impactando?

**Anthony Figueroa:** Hay dos cosas. Por un lado, está el *state of the art*, la situación actual. Creo que tienen un impacto significativo, principalmente en cosas sencillas. En proyectos que recién empiezan, podés llegar a un resultado razonablemente bueno mucho más rápido de lo que lograrías sin estas herramientas.

A medida que aumenta la complejidad del proyecto, y cuanto más largo sea su recorrido histórico, el impacto tiende a disminuir. Por ejemplo, en un proyecto que recién empieza, el impacto puede ser del mil por ciento: literalmente podés hacer las cosas diez veces más rápido. En cambio, en un proyecto que tiene cinco años de historia y muchísimo conocimiento acumulado, el impacto será quizás de un 10, 15 o 20 por ciento como mucho. No es lo mismo que crear un prototipo desde cero.

Lo segundo es que no podemos asumir que esta tecnología va a mantenerse estática. Está avanzando muy rápido, es difícil de predecir, y hay que mantener la mente abierta. Decir que no sirve para proyectos grandes porque hoy su impacto es menor, para mí es subestimar lo que pueden lograr las empresas que están empujando estos desarrollos.

**Matias Mansilla:** Claro, perfecto. Y en cuanto a desafíos, enfocándonos en el rol del desarrollador, ¿qué desafíos ves en la captación y formación de talento dentro de empresas?

**Anthony Figueroa:** Uno de los principales desafíos es que la formación tiene un componente estático. Diseñás una formación para que cualquier persona que entre a la empresa pueda atravesarla, medir su efectividad, mejorarla con el tiempo. Pero en un entorno donde todo cambia tan rápido, esa formación se vuelve obsoleta demasiado rápido.

Por ejemplo, si vas a hacer una prueba técnica, hay dos formas muy distintas de hacerla, y ambas dan información valiosa. Una es sin ningún tipo de ayuda, y otra es dejando que la persona use todas las herramientas que quiera. Para mí, es más valioso que alguien logre un objetivo usando todo lo que tiene a su alcance, que alguien que saca un mejor resultado por memoria o conocimiento acumulado sin apoyo.

Es como lo que pasaba con Google hace años: ¿tenía sentido prohibirlo en una prueba? Hoy hay herramientas que incluso abren un *language model* en segundo plano de forma automática. Me parece que lo importante es evaluar si la persona conoce herramientas, las prueba, descarta, busca alternativas. Esa capacidad de explorar y armar su propio proceso de trabajo con lo que le funciona es uno de los *skills* más importantes hoy.

**Matias Mansilla:** Genial. Cambio un poco el foco. ¿Cómo creés que va a evolucionar el tamaño total del mercado de software? A nivel mundial estoy hablando. La demanda global. No digo solo de servicios de software o consultoras, sino el mercado como un todo. En términos de tamaño, más que nada, es la pregunta.

**Anthony Figueroa:** Sí, sí. Basado en los fundamentos, creo que el software ha ganado una capacidad de hacer cosas y resolver problemas más complejos. También la percepción sobre los problemas que puede resolver el software ha aumentado.

Creo que con cualquier salto tecnológico hay una pérdida de competitividad si no usás la tecnología en tu industria. Por ejemplo, si estás en la industria de la construcción y tus competidores están haciendo cosas mucho más rápido, mucho más eficientemente, o si usan herramientas de IA para hacer presupuestos, estimaciones, etc., vas a desaparecer. Si estás en la industria de supply chain, lo mismo. Si producís energía, lo mismo. Entonces, no veo razones para que no haya un aumento significativo en la necesidad de software en todas las industrias.

**Anthony Figueroa:** Lo que sí puede pasar, como con todos los cambios demasiado innovadores, es que haya un desbalance temporario en el corto plazo. Empresas que no puedan adaptarse a tiempo van a desaparecer. Incluso empresas de tecnología que no logren adaptarse también pueden desaparecer.

Todo cambio grande implica un poco de dolor a corto plazo. Yo creo que a largo plazo todo va a crecer.

**Matias Mansilla:** Y si nos acotamos a lo que puede ser el futuro de la demanda de desarrollo de software, desarrollo tal cual como está ahora, imaginando que también se va adaptando a los diferentes cambios, ¿qué creés que va a pasar con ella en los próximos años?

**Anthony Figueroa:** De nuevo, yo creo que la demanda de software va a seguir aumentando. La cantidad de personas necesarias para transformar una idea en software quizás baje. Y ahí puede haber un balance: aunque aumente la demanda de software, puede disminuir la cantidad de personas necesarias para producir esa misma cantidad de software.

Ese es uno de los grandes signos de interrogación a futuro. En definitiva, lo que termina pasando es que cambian los roles. Si una persona era desarrollador y no se pudo reciclar a tiempo, quizás todos los *skills* que aprendió puedan aplicarse en otra parte del proceso tecnológico. Es difícil saberlo. Creo que es probable que haya algo de dolor a corto plazo, sí. Pero el hecho de que todas las industrias tienen que adaptarse, que todas tienen que sumarse a esta nueva iteración tecnológica, hace que el panorama a mediano y largo plazo sea optimista para nuestra industria.

**Matias Mansilla:** Perfecto. Nos quedan solo dos preguntas. Una es: al principio de la entrevista dijiste algo, ¿notás cambios grandes de expectativas por parte de los clientes, ya desde tu rol de CTO de una consultora de tecnología, en cuanto a velocidad, precio, tiempos, etc.?

**Anthony Figueroa:** Sí, sí. Hay clientes que están planteando hacer cosas que hace cinco años nadie hubiese planteado, porque sabían que estaba fuera de alcance o que sería extremadamente caro. Lo que se logró con una tecnología que es muy barata y bastante generalista, es que eso genera muchas

ideas nuevas. Cuando los clientes cruzan lo que pueden hacer con una app en su celular con lo que quieren resolver en su industria, surgen un montón de ideas.

Dicen: “Bueno, esto mismo puedo aplicarlo acá”. Y eso es fundamental. Ya saben que es un problema que de cierta forma está resuelto. Y como saben que está resuelto, esperan lograrlo por un costo muy bajo, que en muchos casos es cierto. Hoy se pueden resolver problemas muy complejos con costos bajos porque el “cerebro” necesario para resolverlos ya está entrenado, ya existe. Y está en manos de las tres o cuatro empresas que están manejando esto.

**Matias Mansilla:** Perfecto. Eso es lo principal. Y una última pregunta. Desde tu rol de CTO, ¿qué estrategias recomendarías a empresas de servicios de tecnología en este momento?

**Anthony Figueroa:** Yo creo que lo principal es estar actualizado con todas las herramientas y tecnologías que van surgiendo, porque los propios clientes vienen con ideas, conceptos, y referencias que ya conocen bien. Hay que ser conscientes de que esto es un cambio bastante permanente. La forma correcta de aplicar IA en la empresa todavía no está clara, pero creo que hay que exponer herramientas, dar libertad para que las personas elijan sus propios procesos, medir el impacto que están teniendo, y mantenerse abiertos a nuevas soluciones.

Esta tecnología, en particular, no es algo que funcione bien con un enfoque completamente top-down. No es como cuando se decide implementar un CRM y se obliga a todos a trabajar de determinada forma. En este caso, las personas ya están usando IA en su día a día y naturalmente la están incorporando en sus tareas: para programar, para escribir mails, como ayuda para razonar o entender documentos.

Entonces, creo que lo más importante es no bloquear eso. No poner barreras, ni conscientes ni inconscientes, al uso de estas herramientas. Eso es lo que va a marcar la diferencia.

**Matias Mansilla:** Bien, perfecto, gracias Anthony.

## Trascripción entrevista a Luis Staudenmann

**Matias Mansilla:** Bienvenido, Luis. Gracias por tu tiempo. Para empezar, ¿podés presentarte y contarnos brevemente tu experiencia y qué hacés hoy?

**Luis Staudenmann:** Mi nombre es Luis Staudenmann, soy ingeniero en informática del ITBA, en Buenos Aires. Toda mi carrera ha estado ligada al desarrollo de software.

Desde 1998 trabajé con muchos lenguajes, casi siempre en telecomunicaciones; al principio toqué algo de sistemas industriales, pero el foco fue telecom. Siempre me moví en backend y bajo nivel. Mis primeros seis años fueron en una pyme: programaba, diseñaba, documentaba y daba soporte técnico para sistemas de líneas analógicas (E1, SS7 y similares).

Empecé con Visual Basic, luego Visual C++, después C#. En 2004, con ex-compañeros de la facultad, fundamos una empresa orientada a voz sobre IP: al principio todo en C++ con algo de Java para la parte web. Con el tiempo sumamos Python y otras herramientas.

Hoy desarrollamos tanto plataformas que usan las operadoras para servicios de voz como sistemas administrativos, de provisión y control de redes de fibra óptica. Desde el inicio soy gerente de desarrollo y actualmente CTO: superviso ingeniería, desarrollo, diseño y producto, y también el área de soporte y servicios profesionales, que hace despliegues, soporte a clientes y capacitación.

**Matias Mansilla:** Excelente. Gracias, Luis. Voy a una primera pregunta, es bastante amplia. Desde tu perspectiva, ¿cómo está impactando la inteligencia artificial hoy en día en el proceso o en el desarrollo de productos digitales o desarrollo de software en general?

**Luis Staudenmann:** Sí. Yo veo que es muy dispar. Básicamente, el ChatGPT o los LLM. Hasta ese momento, hablar de inteligencia artificial significaba simplemente tener ciertas redes para extrapolar datos, para interpolar datos y ese tipo de cosas, pero se circunscribía únicamente ahí. A partir de la irrupción de ChatGPT principalmente, y después todas las otras herramientas alrededor de los LLM, hubo un boom. Pero la adopción es muy dispar y todavía está muy en pañales. Realmente se está empezando recién a ver el uso y las ventajas. Depende mucho de la empresa.

**Matias Mansilla:** ¿Dónde ves esa disparidad? ¿Incluso en la empresa donde trabajás ves esa diferencia en el uso?

**Luis Staudenmann:** Sí, sí, la verdad es que sí. Nosotros hoy en día la usamos, pero sin una estrategia muy definida. Lo que vengo notando desde este último año, desde el año pasado a esta parte, son los avances muy significativos que están teniendo estos modelos. A veces uno pedía algo y parecía un refrito de lo que podías ir a buscar. De hecho, cuando uno lo iba a buscar a una página, se sentía más seguro, porque sabías que lo había generado una persona. Especialmente en cosas técnicas, nadie escribe algo técnico en un blog o artículo sin tener datos concretos. Entonces, con esto uno siempre estaba con la duda: “¿Será verdad? ¿Lo estará inventando?”. El tema de las alucinaciones, ¿no? Siempre estaba ese miedo.

Últimamente, y con el uso que le estamos dando este año, realmente se ve mucho más sólido. Se pueden pedir cosas y obtener respuestas útiles. Igual, nuestra política interna es muy clara: vos podés usar inteligencia artificial, pero el responsable sos vos. Podés usarlo como una forma de ganar tiempo, que realmente se gana mucho tiempo.

No tenemos mediciones, lamentablemente, pero vemos que especialmente en casos concretos, en problemas puntuales, armar cáscaras, templates, unit tests, o testings con Postman para APIs, todo eso lo estamos usando y realmente se gana tiempo. Incluso en la generación de los casos de prueba, en su especificación.

**Matias Mansilla:** Pero siempre hay que ir, probarlo, verificarlo, entender qué está haciendo, ¿no?

**Luis Staudenmann:** Exactamente.

**Matias Mansilla:** Ahí me respondiste en parte ya a la próxima pregunta, que es: ¿en qué partes lo ves más? ¿En qué partes del proceso de desarrollo lo ves más?

**Luis Staudenmann:** Sí, ahora lo veo en esas. La verdad es que sí. A largo plazo, no digo largo, dos o tres años, vos también tenías una pregunta por el estilo, veo que va a empezar a impactar en todo. Es lo que se dice: uno lee artículos y todos dicen “sí, va a servir para todo”. El cambio que se ha visto en este último año ha sido muy grande, y se espera que eso siga evolucionando con todos los recursos e inversión que se están poniendo.

**Luis Staudenmann:** El tema es... bueno, justamente leyendo las preguntas que me pasaste, todo esto me hace reflexionar por ese lado. ¿A dónde va a llevar todo esto?

**Matias Mansilla:** Claro.

**Luis Staudenmann:** Y creo que todos nos estamos haciendo esa pregunta. Yo, además, doy dos materias en la facultad, de hecho ahora me tengo que ir a tomar un examen. Una de las materias es Estructura de Datos y Algoritmos, y la otra es Redes. No solo protocolos, sino organización de redes. Ambas están muy centradas en tener un conocimiento profundo, dentro de lo que se puede, obviamente.

En Algoritmos vemos estructuras de datos tradicionales: arreglos, listas, pilas, colas, árboles, grafos, y algoritmos sobre todo eso: búsquedas, recorridos, etcétera. Es la base de todo.

**Matias Mansilla:** La base de todo, exacto.

**Luis Staudenmann:** Entonces, uno se imagina que a futuro... de hecho ya es un problema tomar los exámenes gracias a todo esto. Uno se imagina que todo eso va a estar muy resuelto. Y ya incluso en los lenguajes tradicionales eso casi no se ve. El desarrollador dice “uso un HashMap” y listo, me olvido de cómo está implementado. Cada vez más. Y con la inteligencia artificial debería pasar a ser un detalle de implementación. Pero el tema es que eventualmente puede ser necesario entender qué está pasando más abajo, para poder diagnosticar desde arriba, para poder guiar desde arriba.

**Matias Mansilla:** Claro. Va a ser una capa más abstracta. Pero te voy a improvisar una pregunta, quizás aprovechando también tu parte académica y al mismo tiempo tu experiencia en la industria. ¿Cómo creés que se tienen que ir adaptando, o si creés que tienen que adaptarse de forma más rápida o diferente, los programas académicos para formar a los ingenieros del futuro?

**Luis Staudenmann:** Yo creo que sí. Definitivamente creo que se tienen que adaptar, y mucho. El tema es que hay que resolver una pregunta que yo, sinceramente, no tengo clara: ¿qué tanta gente, o qué perfil de desarrollador, tiene que tener realmente un conocimiento profundo?

Hoy en día ya se da algo de esto. Si estás programando una parte más de frontend, web, ese tipo de cosas, estás más del lado del diseño gráfico y de ciertas reglas de negocio, que tendrán su complejidad, no lo discuto, y también requieren estructuras de datos. Pero normalmente están muy abstraídas.

Yo, además, tengo un sesgo porque nuestros sistemas de telecomunicaciones requieren mucha performance. A veces manejamos miles de eventos por segundo. Entonces necesitamos un nivel de optimización importante, y muchas veces trabajar a bajo nivel, con protocolos binarios. No estamos hablando de REST o cosas modernas, sino protocolos del '95, por decir algo. Tenés que entender qué hacen esos 10 bits que vienen de un protocolo antiguo, cosas como TCP, UDP, NATs, a nivel muy bajo.

Entonces sí, yo necesito conocer ese bajo nivel porque, por ejemplo, si tengo un equipo en Telecom, donde procesan miles de llamadas por segundo, y empieza a degradarse la performance, necesito entender de dónde viene el problema. No puedo decir simplemente "confío en las abstracciones" y listo. No tener idea de dónde está el problema no sirve. Creo que ese perfil va a seguir existiendo por bastante tiempo. Pero claramente no es el único perfil. Ya hoy hay lenguajes y abstracciones muy altas, y eso también va a seguir creciendo.

**Luis Staudenmann:** Lo otro, que también lo he leído y lo charlamos con colegas, es que hoy lo más importante en la aplicación de la IA al desarrollo es saber especificar. Siempre hay que poder especificar bien. Y después, saber armar arquitecturas que cumplan con los requerimientos no funcionales: robustez, escalabilidad, flexibilidad.

Muchos de estos requerimientos hoy son más abstractos. Yo, haciendo pruebas pequeñas con IA, veo que, por ejemplo, si le pedís un script que procese imágenes, te puede leer diez veces el mismo archivo, en lugar de hacerlo una vez, guardarlo en memoria, y después operar sobre eso. Entonces tenés que entender el código, más allá de que la IA te lo explique, para saber qué hace. Y después, poder ajustar ese prompt para que lo haga mejor, para que respete los no funcionales.

**Matias Mansilla:** Claro, y para cerrar ese concepto que traés ahora, ¿cuáles son entonces, para vos, las competencias y responsabilidades que van a ser más importantes a futuro para un desarrollador o ingeniero?

**Luis Staudenmann:** Creo que tienen que ver con esto: tener cierta idea del bajo nivel, pero también mucha claridad sobre el alto nivel, sobre la arquitectura, sobre los compromisos que tiene esa arquitectura. Al menos, si le pedís una arquitectura a una IA, que la puedas validar o evaluar.

Lo importante es poder ejercer control. Porque en definitiva, como decíamos antes, el responsable sigue siendo uno. Entonces, hay que poder entender y supervisar lo que se está haciendo. Y además, algo que no te mencioné pero me parece súper importante, lo hablaba con un colega hace poco: tenemos un superpoder. El superpoder de haber hecho todo esto sin IA. De habernos golpeado contra todo esto. Y eso, te da una experiencia que es muy difícil de reemplazar.

**Luis Staudenmann:** Estamos en la generación perfecta. Estuvimos justo en el medio. Vos también pasaste por eso. Una de las cosas más grandes que tenemos nosotros es la experiencia de *debugging*. Más allá de la arquitectura, de entender los compromisos, de pensar en los requerimientos funcionales y no funcionales, el *debugging* es fundamental.

Además está el tema de los requerimientos ocultos, que también es crítico. Y hoy, con la IA, muchas veces todavía no estamos ahí. Son cosas que claramente van a ir mejorando, pero por ahora aún hay mucho por resolver.

El *debugging* es eso: tomar un código y detectar dónde está el problema. ¿Qué proceso seguís para encontrar que el error estaba en una línea que arrancaba en 1 en vez de 0? Cosas así. Ese tipo de razonamiento es muy difícil de reemplazar. Y lo que pasa ahora con algunos juniors que arrancan con inteligencia artificial es que, por ejemplo, ni siquiera leen la documentación. Escuchás testimonios de gente que dice “el ChatGPT me dijo esto” y listo, lo toman como verdad.

La otra analogía que hacíamos, y esto lo vengo viendo desde antes de los LLMs, era con Stack Overflow. Empezó a haber toda una tendencia de “copié esto de Stack Overflow y no sé lo que estoy copiando”. Para mí, hoy ChatGPT y otros modelos son bastante parecidos a Stack Overflow. La diferencia es que en Stack Overflow lo había escrito alguien, entonces, por ahí, uno le daba un poco más de entidad.

De todos modos, siempre hay que hacer hincapié en lo mismo: la responsabilidad es del desarrollador. Para eso estás desarrollando. Todo lo demás son herramientas.

**Matias Mansilla:** Es una buena analogía.

**Luis Staudenmann:** Claro, tenía muchos puntos en Stack Overflow, no podía fallar. Así y todo, había que validar. Si estabas llamando a un método, había que ir a ver la documentación de ese método. Si usabas una función del lenguaje, tenías que chequear si realmente aplicaba o no en tu caso. Porque no siempre las cosas se aplican directamente. Era un “esto es parecido”, y después había que ajustarlo, buscarlo un poco.

**Matias Mansilla:** Bien. Voy a cambiar un poco el foco, también por cuestión de tiempo, y pasar a la parte de demanda. ¿Qué expectativas tenés vos o visión en cuanto, por un lado, a la demanda del tamaño del mercado de software en general, en todo su esplendor, todo tipo de aplicaciones, servicios, y por otro lado la demanda específica de desarrolladores de software?

**Luis Staudenmann:** Yo creo que debería... no sé. Por un lado pienso que podría llegar a caer en algunos casos. Especialmente, y no es por minimizar nada, pero creo que lo más simple es lo que está siendo atacado primero. Es como el "low hanging fruit", ¿no? Cosas como alguien que te hacía queries SQL medio a medida, por ejemplo. Todas esas tareas se han ido productizando en herramientas de consulta, o de filtros.

Después, en la parte web, hay cosas que antes eran simples y que probablemente también vayan desapareciendo de a poco. Lo mismo con otras disciplinas cercanas, como diseño gráfico o UX, que si bien no son desarrollo puro, están bastante relacionadas con las interfaces de usuario. Todo eso también puede verse afectado.

Con el aumento de la productividad por desarrollador, deberías necesitar menos gente para hacer lo mismo. Pero, al mismo tiempo, lo que suele pasar con la aparición de herramientas nuevas es que aparecen también nuevas cosas por hacer. Entonces, es ese juego donde la herramienta no te quita el trabajo, te lo cambia. Dejás de hacer una cosa, pero tenés que hacer otra.

La verdad, no sé. Es ambiguo. Es muy difícil predecir. Ojalá que no caiga la demanda de desarrolladores, pero creo que está atado a cómo evoluciona también la formación. La educación va a tener que cambiar fuerte en poco tiempo. Aun así, no le quito valor al conocimiento de base. Los fundamentos siguen siendo importantes, al menos hasta que no esté todo completamente resuelto y funcione todo perfecto de forma automática.

Yo siempre hago una analogía: hoy no tengo ni idea de cómo funciona realmente un procesador. Tengo una idea mínima, pero seguro está recontra simplificada respecto de la realidad. ¿Y por qué no

sé cómo funciona? Porque está completamente encapsulado. Es una capa de alto nivel que uno simplemente usa.

**Matias Mansilla:** En la tesis exploro justamente la paradoja de Jevons, esa idea de que una mayor eficiencia en el uso de un recurso hace que, en lugar de reducir su uso, aumente la demanda. Y también el concepto de democratización del software.

**Luis Staudenmann:** Exacto. Eso se ve. Mirá, te doy un ejemplo con mis hijos. Más allá de que son mis hijos, viste que cuando uno quiere enseñarles, no quieren, entonces van y exploran por su cuenta. Hace unos meses encontraron una herramienta, ahora no me acuerdo el nombre, que les genera código en JavaScript y HTML para crear jueguitos. Le dicen cosas como “quiero un juego donde los jugadores se tiren tiros y haya obstáculos”, y la herramienta les arma todo. Ven cómo se escribe el código y todo eso.

Y ellos no saben programar. Nunca se interesaron mucho. El más chico tiene nueve años. Pero es el que más se enganchó con eso. Antes, para hacer algo así, necesitabas por lo menos uno o dos años de haber programado. Ahora, él ya tiene claro que esas cosas existen, tiene una necesidad y sabe que hay herramientas que le permiten hacerlo.

Y listo, ya está. Las usa, arma sus juegos, y medio que juega a crearlos.

**Matias Mansilla:** Exacto. Bien. Capaz, como para ir cerrando, te hago dos preguntitas más breves, desde tu rol de CTO, ¿qué recomendaciones hacés a otras personas en tu misma posición para enfrentar estos cambios que está produciendo la inteligencia artificial?

**Luis Staudenmann:** La verdad es que ser muy flexible. Realmente estar muy abierto a todo esto. Yo creo que es una enorme oportunidad. Enorme. Incluso en los productos, ¿no? Ni hablar en los productos. Como rol de CTO, yo y nosotros como empresa lo estamos tomando muy fuertemente como parte de los productos.

Lo que tratamos de no hacer, y es difícil, ¿no?, es meter AI por meter AI. O sea, ponerlo por una cuestión de marketing. Todavía no sucumbimos en eso. Realmente estamos tratando de usarlo con criterio y explorarlo. Todavía no lo tenemos en un producto concreto, pero ya estamos investigando y poniéndolo en práctica a nivel de desarrollo y pruebas. Queremos que realmente nos dé valor, que nos aporte, que nos reduzca mucho los tiempos y que nos agregue flexibilidad. Eso es lo que buscamos.

La idea es que cosas que antes tenían que estar, de alguna manera, harcodeadas, y cuando digo harcodeadas no me refiero a constantes o cosas configurables, sino al código en sí, ahora puedan ser generadas por la AI. Y de esa manera, flexibilizar mucho más el producto.

Pero bueno, ese tipo de cosas. Hay que estar muy atentos. Y también en todo lo que son los procesos. Ahí también puede haber ventajas temporales de valor. Pero, y como vengo diciendo, hay que tener mucho cuidado. Especialmente en los no funcionales. Es ahí donde más impacto puede haber si uno confía ciegamente.

**Matias Mansilla:** Claro, bien. Bien, perfecto. Bien, como te dije, íbamos a ir tomando algo de las entrevistas...

**Luis Staudenmann:** Sí, sí, sí. Está bien, está bien. Perdón si por ahí me extendí.

**Matias Mansilla:** No, no, estuvo