

The background of the image is a dark, blurred collage of various code snippets. These snippets include JSON-like structures with keys such as "place", "mag", "properties", "felt", and "properties". There are also some text-based snippets like "at only have a magnitude greater than 4.0" and "where at least 1 person reported feeling". The text is in various colors (green, orange, white) and is out of focus, creating a sense of depth and a 'hacker' or 'developer' aesthetic.

<it's all'bout> vibecoding

A.PACHECO

<índice>

<carta del autor>	3
<panorama actual>	5
<IA generativa y cambios en el desarrollo>	5
<¿qué es vibe coding?>	7
<el cerebro como herramienta central>	9
<el "kit de supervivencia" del vibe coder>	11
<escalabilidad, riesgos y lecciones>	13
<el futuro del trabajo es ahora>	15
<el nuevo paradigma laboral>	17
<la IA como catalizador de la productividad>	17
<impacto en España y Latinoamérica>	17
<roles humanos aumentados por la tecnología>	17
<la paradoja de la automatización>	18
<requisitos de los roles>	19
<ingeniero de prompt: el traductor entre negocio y máquina>	19
<arquitecto de soluciones de IA: el estratega que evita el caos>	20
<ingeniero de MLOps: el que convierte ideas en productos>	21
<auditor de seguridad y ética en IA: el guardián de tu reputación>	22
<roles complementarios>	23
<corrector de prompts>	23
<analista de calidad de modelos>	23
<especialista en interacción humano-IA>	24
<especialista en ética y gobernanza de IA>	24
<optimizador de modelos o AI trainer>	24
<el talento no siempre tiene el currículum perfecto>	25
<el futuro es el de un arquitecto de intención>	26

<carta del autor>

Este documento parte de una observación sencilla: con la IA generativa es posible pasar de idea a prototipo en horas, pero la factura llega después. El ahorro inicial se traslada a depuración, refactorización, seguridad y mantenimiento. Si no se gestiona, se convierte en deuda técnica acelerada. El objetivo de este whitepaper es dejar claro **cómo trabajar con esa realidad sin autoengañarse**.

Cuando hablo de *vibe coding* no me refiero a “pedirle cosas a una IA”. Hablo de dirigir un sistema que escribe código a la velocidad suficiente como para desordenar un proyecto en días. La herramienta sirve si hay una intención clara, un contexto bien curado y una arquitectura que soporte cambios. Sin eso, solo acelera los errores.

Este texto no se centra solo en metodología: analiza también **el impacto organizativo y los nuevos roles** que surgen de este enfoque. Vibe coding no es una habilidad aislada; es una forma distinta de estructurar equipos. Aparecen perfiles como arquitectos de intención, curadores de contexto y auditores especializados en código generado por IA, que antes no tenían un lugar claro en los organigramas.

Lo que encontrarás aquí se organiza en cuatro bloques:

- **Rol del desarrollador:** de implementador a **arquitecto de intención**. Definir el resultado deseado, los límites del sistema y las interfaces importa más que escribir cada línea.
- **Rendimiento cognitivo:** diferenciar **flow** de **trabajo profundo**. El flow ayuda a avanzar; el trabajo profundo endurece criterio y evita que la IA te lleve por atajos frágiles.
- **Operativa real:** entorno, tácticas de prompt y gestión del contexto. No es “prompt engineering” como truco, es **higiene de proyecto**: dividir el trabajo en encargos precisos, mantener artefactos de arquitectura y hacer que la máquina tenga a mano la información correcta.
- **Ecosistema profesional:** perfiles emergentes, nuevas dinámicas de equipo y formas de medir impacto cuando la generación de código es asistida.

Aquí también encontrarás límites claros y ejemplos reales: patrones inseguros que se cuelean en el código, pérdida de contexto en proyectos grandes, sobredependencia de herramientas externas y la ilusión de que “ya está hecho” porque compila.

Mi conclusión es simple: la IA no quita trabajo, lo redistribuye. La parte mecánica baja; la parte de **criterio, diseño y verificación** sube. *Vibe coding* funciona cuando esa redistribución está pensada desde el principio. Este documento explica cómo hacerlo sin perder el control del sistema ni del equipo.

<panorama actual>

<IA generativa y cambios en el desarrollo>

El desarrollo de software vive su mayor cambio de paradigma desde la adopción de los sistemas de control de versiones y las metodologías ágiles. La aparición de modelos de lenguaje y asistentes de codificación no solo ha añadido una herramienta más a la mesa: ha modificado de raíz el ciclo de trabajo del desarrollador. Hoy, una persona con experiencia puede pasar de una idea a un prototipo funcional en horas. Esa capacidad ha transformado las expectativas de clientes, startups e inversores, y ha empujado a la industria hacia un modelo donde la velocidad inicial ya no es una ventaja competitiva, sino el punto de partida esperado.

La adopción de la IA generativa ha sido explosiva. GitHub informó en su **2024 Copilot Report** que cerca del **30% del código nuevo en lenguajes como Python y JavaScript** ya es generado con asistencia de IA, y este porcentaje es mucho mayor en equipos que adoptan la herramienta de forma plena. Herramientas como GitHub Copilot, Cursor, Replit Ghostwriter y los copilotos integrados en IDEs han normalizado el uso de la IA como parte del proceso de desarrollo. El resultado es que tareas que antes requerían horas de investigación y pruebas se resuelven en minutos, y funciones repetitivas que antes bloqueaban equipos completos ahora se delegan.

Sin embargo, la aceleración tiene un coste. Equipos que adoptan la IA sin un marco claro reportan un aumento significativo de deuda técnica: código generado rápido, pero sin contexto global, con errores difíciles de reproducir y patrones de seguridad débiles. Investigaciones publicadas por **Veracode en 2025** demuestran que el código generado automáticamente introduce **vulnerabilidades críticas en el 45% de las tareas** evaluadas, y que **Java es el lenguaje más afectado, con fallos graves en más del 70% de los casos**, mientras que Python, C# y JavaScript muestran tasas de entre el **38% y el 45%** ([Veracode GenAI Security Report, 2025](#)).

A nivel organizativo, la IA también está cambiando los roles. Las empresas que dependen de software complejo ya no pueden permitirse equipos que funcionen como líneas de montaje. La demanda de perfiles híbridos ha crecido: ingenieros con criterio de producto, arquitectos que entienden prompting avanzado, testers que también son auditores de seguridad, y diseñadores de flujo humano-IA. Lo que antes se consideraba una especialización (como “prompt engineering”) se está integrando en las competencias básicas de desarrollo.

El resultado es un ecosistema donde la velocidad de entrega y la capacidad de iterar se dan por sentadas, pero el verdadero reto es la gestión del contexto y la sostenibilidad. Los proyectos que sobreviven no son los que generan más líneas de código, sino los que mantienen una arquitectura clara y un ciclo de validación constante. La IA ha convertido al

desarrollador en arquitecto de intención: su trabajo ya no es solo escribir código, sino guiar sistemas que lo escriben.

Este cambio es irreversible. Ignorar la IA generativa es tan inviable como ignorar en su momento el control de versiones o los frameworks. La diferencia es que esta transición es mucho más rápida y con mayor impacto: en dos años, la industria ha pasado de “probar asistentes” a integrar pipelines completos basados en modelos de lenguaje. El panorama actual exige que los equipos redefinan sus procesos antes de que la deuda técnica y el caos organizativo se acumulen.

<¿qué es vibe coding?>

El "vibe coding" se ha consolidado como una práctica de desarrollo de software que aprovecha la inteligencia artificial (IA) para generar código funcional a partir de instrucciones en lenguaje natural. Este enfoque, acuñado a principios de 2025 por el investigador de IA Andrej Karpathy, redefine el rol del desarrollador: en lugar de escribir código línea por línea, el profesional se convierte en un supervisor o guía, dirigiendo a un asistente de IA a través de un proceso más conversacional. La esencia del término reside en un flujo de trabajo que prioriza la intuición, la creatividad y el "sentimiento" por el proyecto, permitiendo al desarrollador enfocarse en la visión y la arquitectura en lugar de en los detalles técnicos repetitivos.

Este concepto no representa una simple herramienta, sino un cambio de paradigma que transforma el proceso de desarrollo. Al liberar al programador de la labor mecánica, el vibe coding le permite concentrarse en la resolución de problemas complejos y en la innovación. A diferencia de los modelos de "no-code" o "low-code", que limitan la personalización a plataformas predefinidas, el vibe coding sigue exigiendo un profundo conocimiento técnico para refinar, validar y escalar el código generado por la IA.

La promesa más atractiva del vibe coding es su velocidad: pasar de una idea a un prototipo funcional en cuestión de minutos o incluso horas. Sin embargo, esta aparente eficiencia inicial no está exenta de costos. El tiempo que se ahorra en la generación del código inicial a menudo se pierde más tarde en la depuración, la corrección de errores y la refactorización. Esto revela una transferencia de esfuerzo, no una eliminación. El costo real se manifiesta en la acumulación acelerada de "deuda técnica", que incluye código carente de estructura, difícil de mantener y que introduce vulnerabilidades latentes. El éxito, por tanto, no se mide por la velocidad de la generación inicial, sino por la habilidad del desarrollador para gestionar la fase posterior. Esto exige una disciplina de ingeniería de software incluso mayor para evitar que el desastre a gran escala se manifieste en proyectos complejos.

Para contextualizar este cambio, la siguiente tabla detalla las diferencias fundamentales entre el vibe coding y la programación tradicional, destacando cómo el rol del desarrollador evoluciona de artesano de la implementación a arquitecto de la intención.

	Vibe Coding	Programación tradicional
Entrada principal	Lenguaje natural y prompts conversacionales	Código preciso y sintaxis estricta
Velocidad de desarrollo	Potencialmente más rápido, ideal para prototipos y tareas simples	Generalmente más lento y metódico
Manejo de errores	Depuración y refinamiento conversacional con la IA	Comprensión del código
Curva de aprendizaje	Más accesible para principiantes y no programadores	Requiere años de estudio
Rol del desarrollador	Supervisor, arquitecto de la intención, director del proceso	Artesano, implementador de lógica,

<el cerebro como herramienta central>

El verdadero éxito en la programación, más allá del dominio de lenguajes o frameworks, **depende del cerebro del desarrollador como instrumento de alto rendimiento**. No importa cuán sofisticada sea la IA: el factor decisivo sigue siendo **la capacidad de concentración sostenida, claridad mental y toma de decisiones estratégicas**.

El **vibe coding** potencia esta capacidad al permitir que el desarrollador entre en **estado de flujo (flow state)**, concepto popularizado por el psicólogo **Mihaly Csikszentmihalyi**. El flujo es una condición mental en la que la persona está **totalmente inmersa y concentrada en una tarea**, con sensación de control y pérdida de la noción del tiempo. Este estado se asocia con **máxima creatividad y productividad**; es el equivalente mental de un atleta en su mejor momento.

Al delegar tareas mecánicas y repetitivas a la IA, el vibe coding elimina muchas de las **interrupciones cognitivas** que antes eran inevitables. En este escenario, el desarrollador mantiene su energía en la **arquitectura, el diseño lógico y la resolución creativa de problemas**. La IA no solo escribe código: **actúa como un filtro de ruido**, liberando la mente del trabajo monótono y dejando espacio para el pensamiento abstracto.

Sin embargo, este entorno tiene una **paradoja peligrosa**: la facilidad que ofrece la IA puede inducir una **falsa sensación de maestría**. El autor **Cal Newport**, en su libro *Deep Work*, diferencia entre el “flow” y el **trabajo profundo**. Mientras que el flujo ocurre cuando las habilidades coinciden perfectamente con el desafío, el trabajo profundo implica **adentrarse en tareas incómodas y cognitivamente exigentes** que fuerzan el crecimiento real. En otras palabras, **sentirse bien programando no significa necesariamente estar mejorando**.

El riesgo es evidente: depender exclusivamente de la IA puede generar un flujo superficial, pero **no desarrolla habilidades avanzadas** ni refuerza la capacidad de resolver problemas complejos sin asistencia. Para **alcanzar excelencia profesional**, el programador debe combinar ambos estados:

- **Flow state**: para ejecutar tareas creativas con energía y concentración.
- **Trabajo profundo**: para enfrentarse a límites reales, mejorar razonamiento y consolidar conocimiento técnico.

En esta dinámica, la IA debe ser vista como una **herramienta de aceleración**, no como sustituto del aprendizaje duro. Sirve para liberar tiempo y foco, pero no para evitar la práctica incómoda que construye maestría.

Lograr este equilibrio requiere desarrollar una **mentalidad específica**:

- **Shoshin (mente de principiante):** afrontar la tecnología sin ego, con curiosidad constante y disposición a aprender incluso lo básico.
- **Mentalidad de crecimiento:** asumir que las habilidades avanzadas se construyen con dedicación y esfuerzo sostenido.
- **Disciplina cognitiva:** integrar prácticas como **meditación, respiración consciente y mindfulness** para entrenar la atención y reducir ruido mental.

Estas técnicas son más que consejos de bienestar: son **herramientas neurocientíficas probadas** para aumentar la densidad de materia gris en áreas asociadas con concentración y memoria de trabajo. Su aplicación diaria permite al desarrollador entrar en estados de concentración profunda de forma más predecible y recuperarse más rápido de contextos de multitarea.

En resumen, **el vibe coding redefine el rol del cerebro humano en el desarrollo**: el código ya no es escrito línea a línea como un artesano, sino que se diseña y supervisa desde un nivel de abstracción superior. La mente del desarrollador se convierte en **sistema operativo de alto nivel**, mientras que la IA funciona como procesador que ejecuta instrucciones. Pero este cambio exige un **entrenamiento deliberado**; quien no fortalezca su capacidad cognitiva acabará siendo un operador de modelos, no un arquitecto de soluciones.

<el "kit de supervivencia" del vibe coder>

Dominar el **vibe coding** exige mucho más que saber usar una IA. Implica construir un **ecosistema personal y técnico** diseñado para sostener sesiones de trabajo profundas, estructuradas y de alto rendimiento. Este ecosistema empieza en el **entorno físico**, que debe estar optimizado para **minimizar distracciones** y favorecer la **concentración sostenida**. Elementos como **auriculares con cancelación de ruido**, **sillas ergonómicas** y una **organización intencional del espacio** no son simples comodidades, sino requisitos básicos para que el cerebro funcione como **extensión directa del editor de código**.

En el plano digital, el **entorno de desarrollo integrado (IDE)** pasa de ser una herramienta funcional a convertirse en **la interfaz entre el pensamiento del programador y las capacidades de la IA**. Editores como **VS Code**, **Cursor** o **Replit Ghostwriter**, combinados con asistentes como **GitHub Copilot**, representan el punto de partida de cada tarea. Su **personalización mediante extensiones y plugins** es crítica, ya que un entorno afinado actúa como **multiplicador de productividad**, mientras que una configuración pobre ralentiza todo el flujo.

El dominio del **prompting avanzado** es el verdadero núcleo del kit de supervivencia. Un vibe coder no se limita a dar órdenes simples, sino que diseña interacciones estratégicas con la IA:

- Con **“show me the pattern”**, puede obtener la **estructura esencial de una librería o concepto** en segundos, lo que evita lecturas extensas de documentación.
- Con **“fix this quickly”**, transforma un error detectado en una solución inmediata al describir el comportamiento esperado frente al actual, agilizando la depuración.
- Con **“build on this”**, extiende una funcionalidad existente sin reescribir todo el código, lo que garantiza **mejoras incrementales seguras**.
- Con **“stub this out”**, define la **arquitectura de nuevos componentes** mediante esqueletos de métodos vacíos, lo que facilita el diseño sin perder el ritmo creativo.

Estas tácticas son ejemplos de cómo convertir a un modelo de lenguaje en un **colaborador especializado**, no en una simple calculadora de texto.

Sin embargo, la **gestión de contexto** es el verdadero reto. Los modelos de IA tienden a perder información en proyectos grandes y esta limitación convierte al desarrollador en **curador activo de conocimiento**. Para sostener el flujo, el vibe coder mantiene **documentación clara, manuales de despliegue, diagramas de arquitectura e índices de proyecto**, asegurando que cada consulta a la IA tenga precisión. Los proyectos se fragmentan en **tareas pequeñas y bien delimitadas**, similares a historias de usuario o tickets de Jira, cada una acompañada de prompts específicos. Esta práctica de **microinstrucciones bien diseñadas** es la base de un proceso profesional y repetible.

El ecosistema también incluye una capa invisible pero crítica: **rituales de alto rendimiento**. Automatizar tareas repetitivas mediante **scripts, pipelines y bots internos** libera espacio mental y permite que el desarrollador reserve su atención para decisiones de diseño y arquitectura. Además, la **salud física y mental** se integra como herramienta de trabajo: prácticas de **meditación, ejercicio regular y pausas intencionales** no son simples consejos, sino sistemas que permiten **mantener claridad cognitiva, prevenir agotamiento y sostener rendimiento prolongado**.

Este kit no es una lista de accesorios o consejos dispersos, sino un **sistema integral** donde cada decisión, desde la elección del editor hasta el orden de la mesa, está alineada con la capacidad de trabajar con **máquinas generativas de forma efectiva y sostenible**.

<escalabilidad, riesgos y lecciones>

El **vibe coding** promete velocidad y creatividad, pero este potencial viene acompañado de **riesgos significativos** que no se pueden ignorar. Lo que para algunos es una revolución democratizadora del desarrollo, para otros expertos es una señal de alarma. **Dave Farley**, coautor de *Continuous Delivery*, ha criticado públicamente este enfoque, calificándolo como una “**pantalla**” que **esconde la necesidad de ingeniería sólida**. Su argumento es contundente: aunque la IA puede generar código rápidamente, aún no existe modelo capaz de **entender los requisitos de negocio, gestionar la complejidad de sistemas reales o anticipar decisiones arquitectónicas con impacto a largo plazo**.

Los casos reales refuerzan esta visión. El ejemplo más citado es el de **EnrichLead**, una aplicación construida íntegramente con asistentes de IA. Lanzada con entusiasmo, rápidamente ganó notoriedad por su capacidad para entregar funcionalidades a gran velocidad. Sin embargo, **falló estrepitosamente** cuando las vulnerabilidades de seguridad, nunca auditadas, permitieron ataques que obligaron a cerrar el servicio. Este caso demuestra que la **velocidad inicial sin control técnico es una ventaja efímera**: sin procesos de validación, el código generado se convierte en un riesgo operativo y reputacional.

La investigación académica confirma el problema. Un estudio de la **Universidad de Stanford** en 2024 analizó fragmentos de código generados por GitHub Copilot y encontró que **el 40 % de los snippets contenía vulnerabilidades** en Python y hasta **el 70 % en Java**. Estos porcentajes reflejan que la IA, aunque poderosa para generar código funcional, **no prioriza seguridad ni buenas prácticas a menos que se le indique explícitamente**. Además, los modelos de lenguaje tienden a reproducir patrones inseguros encontrados en datos de entrenamiento públicos, lo que puede amplificar problemas históricos de seguridad.

A estos riesgos se suma la **deuda técnica acelerada**. La IA tiende a generar soluciones que resuelven el problema inmediato, pero carecen de **contexto global del proyecto**. Sin una supervisión rigurosa, esto conduce a **código difícil de depurar, inconsistencias estructurales y una base de software prácticamente imposible de escalar**. La aparente “facilidad” del vibe coding también puede erosionar habilidades: delegar sistemáticamente tareas básicas en la IA reduce el **razonamiento lógico, la destreza en depuración manual y la capacidad de enfrentar problemas desde cero**, competencias críticas para cualquier ingeniero.

El reto, por tanto, no es decidir si se usa IA, sino **cómo** se integra. El enfoque que está ganando fuerza entre equipos técnicos de alto nivel es el del **innovador pragmático**:

- Usar IA para **generación rápida de prototipos y boilerplate** (código repetitivo y estandarizado).

- Mantener siempre **un humano en el bucle** para revisar y refactorizar cada entrega antes de desplegar.
- Establecer **políticas de gobernanza estrictas**, auditorías continuas de seguridad y estándares de calidad.
- Incorporar herramientas de análisis automático como **Snyk, SonarQube y Dependabot** para reforzar seguridad y evitar que errores se propaguen.

Este modelo híbrido ya es visible en empresas aceleradas como **Y Combinator**. Las startups que adoptan IA no buscan reemplazar ingenieros, sino **acelerar ciclos iniciales** mientras concentran el talento humano en **decisiones estratégicas y arquitectura compleja**. En este contexto, la IA se convierte en un **multiplicador de fuerza** para equipos experimentados, no en un sustituto de su criterio.

El mensaje es claro: el vibe coding no es una bala de plata. Su éxito depende de **disciplina, procesos y visión técnica**. Adoptarlo sin una base sólida equivale a construir rascacielos sobre cimientos de arena. Integrarlo con criterio, en cambio, permite liberar el tiempo de los desarrolladores para lo que realmente importa: diseñar sistemas robustos, escalables y seguros.

<el futuro del trabajo es ahora>

La adopción acelerada de la inteligencia artificial, especialmente de la IA generativa, está reconfigurando el mercado laboral global a una velocidad sin precedentes. Lo que hace apenas cinco años parecía ciencia ficción, con modelos capaces de generar texto, código y estrategias de negocio, hoy es una herramienta cotidiana en empresas de todos los tamaños. Este cambio no es únicamente una amenaza para los empleos. Es una transformación profunda y sostenida que está creando más oportunidades de las que destruye.

Los datos lo confirman. El Foro Económico Mundial estima que, aunque la IA podría desplazar 92 millones de empleos a nivel mundial para 2030, también generará una ganancia neta de 78 millones de nuevos roles especializados. La IA no solo automatiza tareas repetitivas, sino que eleva la productividad humana y habilita funciones profesionales que requieren pensamiento crítico, creatividad y criterio humano. El éxito en este entorno depende de habilidades híbridas que combinan tecnología, estrategia y visión de negocio.

Entre los perfiles que concentran mayor demanda destacan cuatro que representan la transformación en curso.

El **ingeniero de prompt** traduce necesidades complejas de negocio en instrucciones precisas para modelos generativos. Su especialidad está en el diseño de prompts que maximizan el rendimiento de los sistemas de IA. Para lograrlo necesita experiencia en procesamiento de lenguaje natural, dominio de Python y una capacidad de abstracción que le permita alinear los resultados con la estrategia de una organización. En sectores como salud, finanzas y logística este rol ya es crítico, con salarios que alcanzan en España los 60.000 euros anuales y superan los 130.000 dólares en Estados Unidos.

El **arquitecto de soluciones de IA** diseña la infraestructura y gobernanza que permite que estos modelos funcionen de forma segura y escalable. Sus responsabilidades incluyen la integración de sistemas, la definición de políticas de uso, la gestión de riesgos y el cumplimiento normativo. Requiere experiencia en arquitecturas cloud y certificaciones avanzadas en plataformas como AWS, Azure y Google Cloud. En Europa, sus salarios oscilan entre 52.000 y más de 100.000 euros anuales, reflejando su papel estratégico.

El **ingeniero de MLOps** es el vínculo entre la ciencia de datos y las operaciones. Se encarga de que los modelos pasen de prototipos a entornos de producción sin perder calidad ni estabilidad. Su trabajo implica el diseño de pipelines de despliegue, la monitorización de modelos y la integración de herramientas como Docker, Kubernetes y MLflow. Su perfil es uno de los más valorados en empresas tecnológicas y startups de alto crecimiento. En Estados Unidos su salario supera los 140.000 dólares anuales y puede alcanzar los 280.000 en compañías líderes.

El **auditor de seguridad y ética en IA** evalúa la integridad de los sistemas en todo su ciclo de vida. Su objetivo es identificar vulnerabilidades, sesgos y riesgos de cumplimiento en

modelos que procesan datos sensibles. Requiere experiencia en ciberseguridad, machine learning y normativa como el GDPR y la Ley de IA de la Unión Europea. Este rol se ha convertido en esencial en sectores regulados como banca y salud. En España su salario supera los 60.000 euros, mientras que en Estados Unidos ronda los 125.000 dólares anuales.

Estos perfiles ilustran cómo la IA no solo está cambiando las herramientas de trabajo, sino también la estructura de los equipos y el valor que las organizaciones asignan a ciertas competencias. El conocimiento técnico especializado ya no es suficiente por sí solo. Los profesionales más demandados son aquellos que entienden el contexto de negocio, saben tomar decisiones estratégicas y dominan las tecnologías que están marcando el ritmo del cambio.

<el nuevo paradigma laboral>

La conversación sobre el impacto de la inteligencia artificial en el mercado laboral ha pasado de centrarse en el miedo al reemplazo a reconocer una colaboración creciente entre humanos y máquinas. La IA, especialmente en su vertiente generativa, no es únicamente una herramienta de automatización: es un motor de productividad y un catalizador de nuevas oportunidades profesionales. Si bien algunos empleos con alta repetitividad están siendo transformados, el resultado global es más complejo y equilibrado.

<la IA como catalizador de la productividad>

El Foro Económico Mundial estima que para 2030 la IA podría desplazar 92 millones de empleos a nivel mundial, pero también generará 78 millones de nuevos puestos. Esto no refleja solo la creación de roles inéditos, sino la transformación de los existentes. Profesionales de sectores como ingeniería, marketing y análisis de datos están utilizando la IA para delegar tareas repetitivas, lo que incrementa su capacidad de concentrarse en procesos estratégicos y creativos. Esta colaboración reduce el margen de error humano, optimiza flujos de trabajo y eleva la productividad general de las organizaciones.

<impacto en España y Latinoamérica>

La adopción de la IA no se limita a los centros tecnológicos tradicionales. España, por ejemplo, ha experimentado un incremento del 454 % en las ofertas laborales relacionadas con IA durante los últimos cinco años, según Hays. Este crecimiento ha generado una brecha entre la demanda de profesionales cualificados y la oferta disponible. Hoy, el 52 % de los profesionales españoles ya usa IA generativa en su trabajo diario, y el 51 % de las empresas fomenta activamente su adopción. América Latina sigue una tendencia similar, impulsada por la inversión extranjera y el crecimiento de ecosistemas tecnológicos locales, lo que ha convertido estas regiones en mercados estratégicos para talento especializado.

<roles humanos aumentados por la tecnología>

El impacto de la IA se manifiesta en la redefinición de roles tradicionales. La inteligencia artificial no está reemplazando la inteligencia humana, sino que la complementa. Un informe de LinkedIn destaca que las habilidades de los ingenieros de software se ven potenciadas en un 96% con la ayuda de la IA generativa, demostrando un claro efecto de **upskilling**. Este nuevo entorno exige una **alfabetización digital** que ha dejado de ser una habilidad deseable para convertirse en un requisito indispensable. La capacidad de colaborar eficazmente con herramientas de IA, interpretar sus resultados y aplicar el juicio humano para la toma de decisiones estratégicas se está convirtiendo en una habilidad fundamental.

<la paradoja de la automatización>

La dinámica del mercado laboral de la IA presenta una serie de complejidades. A medida que la IA acelera la automatización, especialmente en roles con un alto potencial de automatización como auxiliares de contabilidad, representantes de servicio al cliente y secretarías, se observa una clara cadena de causalidad: la IA empuja a los profesionales a adquirir habilidades de mayor valor. Esta especialización se traduce en la creación de nuevos roles de alta demanda, como los especialistas en IA, **big data** y **machine learning**. Esto transforma el panorama profesional de una amenaza a una hoja de ruta para el **reskilling** y el **upskilling**, donde la IA **quita carga** de las tareas repetitivas en lugar de eliminar el trabajo.

Paralelamente, la facilidad de uso de la IA generativa está acelerando su adopción y, por ende, la automatización. Sin embargo, la tecnología sigue necesitando el criterio humano para establecer estrategias, interpretar contextos complejos y tomar decisiones éticas. Esto ha generado una paradoja: una tecnología que, por su accesibilidad, automatiza más rápido, pero que al mismo tiempo aumenta la demanda de habilidades blandas y de juicio crítico. El profesional de éxito en este entorno no será el que sepa usar la herramienta más avanzada, sino el que sepa cómo validar y aplicar su conocimiento experto donde la máquina no puede.

<requisitos de los roles>

Si antes hablábamos de los trabajos, ahora toca que aprendas a entender, quien, como, cuando y donde puedes poner a cada uno de esos profesionales.

cuando hablamos de un entorno nuevo, un nuevo escenario, un nuevo deporte, un nuevo espectro de color, debemos asegurarnos de confundirnos y no dejarnos confundir, así que veamos.

<ingeniero de prompt: el traductor entre negocio y máquina>

Ya sabes lo que hace: traduce ideas vagas en instrucciones que los modelos entienden, limpia errores antes de que lleguen al usuario y convierte la IA en inversión, no en juguete. Pero ¿qué tienes que pedirle exactamente si pretendes contratar a uno?

<lo qué pide el puesto>

Experiencia aplicando técnicas como **few-shot**, **zero-shot** y generación basada en contexto. Capacidad para integrar APIs de LLMs y automatizar pruebas en Python. Conocimiento sólido para medir sesgo, evaluar errores y traducir objetivos de negocio en prompts precisos.

<credenciales deseables>

Un título en informática, IA o ciencias de datos suma, pero no sustituye. Lo que realmente cuenta es un portafolio: proyectos reales, repositorios públicos, pruebas cuantificables de mejora de modelos. Cursos y certificaciones en NLP son bienvenidos si acompañan resultados tangibles.

<hard skills>

Python avanzado, dominio de NLP, comprensión de **embeddings** y **context windows**, capacidad para diseñar métricas de calidad, integración vía APIs, y conocimiento básico de seguridad aplicada a modelos.

<soft skills>

Pensamiento crítico, comunicación clara, tolerancia a la ambigüedad, curiosidad, enfoque obsesivo en mejora continua y paciencia para iterar tras resultados fallidos.

<salarios y proyecciones>

En España, este rol se paga entre **53.800 y 54.000 euros** anuales de media. En ciudades como Madrid, la cifra sube a **59.000-60.000 euros** para perfiles con experiencia.

En EE. UU. y empresas globales, el rango es **90.000-140.000 USD** para roles senior, llegando a picos más altos en big tech.

<no ofrezcas menos de lo que deben cobrar>

Porque un ingeniero de prompt sólido reduce errores de modelo que cuestan dinero, acorta ciclos de desarrollo, evita refactorizaciones costosas y mantiene la calidad del producto sin multiplicar costes de QA y soporte. Si ofreces menos, estás aceptando fallos invisibles que después se convierten en deuda técnica y pérdidas de reputación.

<arquitecto de soluciones de IA: el estratega que evita el caos>

Este perfil no se limita a “montar servidores”. Es quien diseña cómo fluye la información, cómo se comunican los modelos y qué tan segura y escalable es toda la arquitectura de IA de tu empresa. Si quieres que tus proyectos no se conviertan en una torre de Babel tecnológica, necesitas uno.

<lo que pide el puesto>

Experiencia en **arquitectura de sistemas complejos**, gobierno de datos, cumplimiento normativo y despliegue en plataformas cloud (AWS, Azure, GCP). Capacidad para anticipar riesgos, diseñar infraestructuras modulares y tomar decisiones que mantengan control y escalabilidad.

<credenciales deseables>

Grado en informática, ingeniería de software o sistemas de información. Cinco años o más diseñando arquitecturas empresariales, idealmente con certificaciones TOGAF o cloud. Lo más importante: experiencia demostrable liderando proyectos de transformación digital.

<hard skills>

Cloud avanzado, Kubernetes y Docker, infraestructura como código, integración de microservicios, redes seguras, autenticación y cumplimiento GDPR, monitorización de sistemas críticos.

<soft skills>

Visión estratégica, liderazgo técnico, comunicación clara con equipos directivos, capacidad de priorizar riesgos, mentalidad de producto y resistencia a la presión.

<salarios y proyecciones>

En España, este rol oscila entre **66.000 y 85.000 euros** anuales según experiencia y tamaño de la empresa, con picos que superan los **90.000 euros** en Madrid y Barcelona.

En EE. UU., un arquitecto de soluciones de IA senior está entre **120.000 y 200.000 USD**, con casos que superan esta cifra en compañías globales.

<no ofrezcas menos de lo que deben cobrar>

Un arquitecto de soluciones deficiente no solo cuesta dinero: genera sistemas inseguros, infraestructuras inestables y deuda técnica difícil de corregir. Este perfil es quien define el esqueleto tecnológico de la empresa, y pagar menos equivale a aceptar riesgos de seguridad, incumplimientos legales y futuros gastos millonarios.

<ingeniero de MLOps: el que convierte ideas en productos>

Tener modelos que funcionan en un notebook no significa que estén listos para producción. Aquí entra el ingeniero de MLOps: la persona que convierte experimentos en sistemas estables que pueden escalar.

<lo que pide el puesto>

Experiencia construyendo **pipelines CI/CD para machine learning**, monitorización de modelos, rollback automatizado, optimización de costes cloud y procesos de versionado. Capacidad para traducir modelos académicos en entornos empresariales robustos.

<credenciales deseables>

Título en informática, datos o ingeniería. Experiencia práctica de mínimo dos años en DevOps, ML o MLOps. Certificaciones en cloud y contenedores son muy valoradas, pero la experiencia real pesa más que el papel.

<hard skills>

Python, Docker, Kubernetes, Terraform, MLflow, Kubeflow, Spark, monitoreo de deriva de datos, escalado automático, seguridad en pipelines.

<soft skills>

Mentalidad de automatización, pensamiento sistémico, atención al detalle, resiliencia y capacidad de comunicación con científicos de datos, ingenieros y negocio.

<salarios y proyecciones>

En España, los salarios van de **67.000 a 75.000 euros**, llegando a **90.000 euros** en puestos senior.

En EE. UU., este rol oscila entre **140.000 y 180.000 USD**, con picos en FAANG que superan los **200.000 USD**.

<no ofrezcas menos de lo que deben cobrar>

Casi la mitad de los proyectos de IA fallan al intentar llevar modelos a producción. Este perfil existe para que eso no pase. Si ahorras en MLOps, pagas en despliegues lentos, fallos constantes y pérdidas de confianza de tus clientes.

<auditor de seguridad y ética en IA: el guardián de tu reputación>

Este rol analiza y corrige todo lo que puede dañar tu empresa: sesgos en modelos, vulnerabilidades, incumplimientos normativos. No es un “extra” de compliance, es quien evita demandas, multas y titulares negativos.

<lo que pide el puesto>

Experiencia en **ciberseguridad aplicada a IA**, pruebas de penetración, detección de sesgos algorítmicos y conocimiento profundo de normativas como GDPR, CCPA y la Ley de IA europea.

<credenciales deseables>

Formación en seguridad, auditoría o IA. Certificaciones como **CISA**, CISSP o ISACA son diferenciales. Portafolio con informes reales de auditoría o proyectos regulatorios.

<hard skills>

Machine learning avanzado, análisis de riesgos, herramientas de seguridad ofensiva y defensiva, detección de sesgos, regulación europea y estadounidense, modelado de amenazas.

<soft skills>

Ética profesional, visión crítica, detalle extremo, comunicación clara con equipos legales y técnicos, proactividad para anticipar fallos.

<salarios y proyecciones>

En España, un auditor de seguridad especializado puede ganar entre **40.000 y 60.000 euros**, con directores de ciberseguridad superando los **80.000 euros**.

En EE. UU., el salario medio está en torno a **124.000 USD**, con roles sénior que superan los **150.000 USD**.

<no ofrezcas menos de lo que deben cobrar>

Este perfil protege tu negocio de ataques, fraudes y sanciones. Ahorrar en este rol es abrir la puerta a incidentes de seguridad que pueden costar millones. Su trabajo garantiza confianza, estabilidad y cumplimiento normativo en mercados cada vez más regulados.

<roles complementarios>

La adopción masiva de IA generativa y vibe coding ha creado un ecosistema profesional mucho más amplio de lo que se imaginaba hace apenas dos años. Más allá de los cuatro roles principales, están apareciendo perfiles especializados que cubren necesidades muy concretas: control de calidad, ética, experiencia de usuario y optimización de modelos. Aunque algunos todavía no tienen bandas salariales consolidadas, su relevancia ya es clara en empresas con productos críticos o infraestructura basada en IA.

<corrector de prompts>

Este perfil se centra en revisar y optimizar prompts existentes. Su objetivo es mejorar precisión, estilo y coherencia, eliminando ambigüedades y reduciendo sesgos. También prueba variaciones para garantizar resultados consistentes entre modelos. Es esencial en proyectos con asistentes conversacionales, documentación sensible o productos de marca, donde el tono y la claridad importan tanto como la exactitud técnica.

<analista de calidad de modelos>

El analista de calidad es responsable de evaluar la estabilidad y precisión de los modelos antes y después del despliegue. Aporta una combinación de experiencia en control de calidad tradicional y conocimiento profundo de métricas de machine learning. Su función es clave para evitar errores invisibles que, sin una revisión exhaustiva, se multiplican cuando el producto llega al mercado.

<especialista en interacción humano-IA>

A medida que los sistemas impulsados por IA se vuelven parte del día a día, el diseño de la interacción cobra un papel estratégico. Este especialista une experiencia en UX con comprensión técnica para diseñar interfaces conversacionales, paneles y flujos donde el usuario entienda, confíe y adopte la tecnología. La percepción de la IA depende en gran parte de este rol.

<especialista en ética y gobernanza de IA>

Con regulaciones como el AI Act europeo, este perfil se está convirtiendo en un requisito. Evalúa riesgos y sesgos, asegura el cumplimiento legal y define políticas internas para el uso responsable de modelos. Es un rol indispensable en sectores regulados y en empresas que trabajan con datos sensibles o de misión crítica.

<optimizador de modelos o AI trainer>

Este perfil se encarga del ajuste fino y mejora continua de los modelos. Trabaja sobre pipelines de entrenamiento, analiza métricas avanzadas y adapta modelos a nuevos contextos de negocio. Su trabajo evita que la inversión en IA pierda efectividad con el tiempo y permite mantener sistemas de alto rendimiento.

<el talento no siempre tiene el currículum perfecto>

En este sector hay algo que debemos entender desde ya: la excelencia profesional en IA y *vibe coding* todavía no tiene un molde claro. Los perfiles “perfectos” en papel son escasos y, en muchos casos, irrelevantes para los problemas reales que una empresa quiere resolver. La industria avanza demasiado rápido como para que títulos universitarios, certificaciones masivas o trayectorias corporativas sean garantía de rendimiento.

Los roles que hoy definen el futuro (ingenieros de *prompt*, arquitectos de soluciones, auditores de IA) apenas llevan unos años existiendo. Esto significa que gran parte del talento que necesita el mercado **no viene “hecho” de fábrica**: hay que detectarlo, formarlo y darle espacio para crecer. Empresas que saben reconocer potencial más allá del CV logran construir equipos competitivos antes que sus competidores, porque no esperan a que aparezca el “candidato ideal” que nunca va a llegar, sino que crean su propio ecosistema de expertos.

En lugar de fijarse solo en títulos, conviene mirar:

- Capacidad real de aprendizaje y adaptación en entornos nuevos.
- Pensamiento crítico y criterio técnico para evaluar herramientas.
- Habilidad para resolver problemas sin depender ciegamente de documentación o recetas previas.
- Curiosidad y mentalidad experimental, fundamentales en un mercado donde no hay una única forma correcta de hacer las cosas.
- Ética y responsabilidad, que pesan tanto como el conocimiento técnico cuando se trabaja con sistemas de impacto masivo.

Estamos en una etapa en la que todavía **no hay “rockstars” de la IA** con décadas de experiencia. Lo que existe son profesionales en formación constante, algunos autodidactas, otros desde entornos académicos, que tienen la oportunidad de ser los próximos referentes de este sector. Y ahí entra el rol de las empresas y equipos técnicos: identificar talento emergente, invertir en su desarrollo, y acompañarlos para que sean ellos quienes, en pocos años, evangelicen y construyan estándares de calidad en IA.

Contratar para IA en 2025 no se trata de fichar currículos espectaculares, sino de detectar a personas con visión, rigor y capacidad de aprender rápido. El verdadero valor está en **apostar por quienes aún no son expertos, pero tienen todo lo necesario para convertirse en ellos**.

<el futuro es el de un arquitecto de intención>

El vibe coding es una ruptura en la forma en que diseñamos y construimos software, un cambio de paradigma que no se limita a “escribir código más rápido”, sino que redefine el papel de quienes construyen tecnología. Por primera vez, los desarrolladores tienen en sus manos una herramienta capaz de escribir por ellos, pero eso no los hace menos relevantes: los convierte en **directores de orquesta**.

El verdadero impacto de esta revolución no está en la generación automática de líneas de código, sino en la **capacidad de pensar, guiar y estructurar** sistemas de enorme complejidad sin quedar atrapados en las tareas mecánicas que consumían horas y creatividad. El desarrollador deja de ser un ejecutor y se convierte en un **arquitecto de intención**, alguien que no mide su valor por teclas pulsadas, sino por decisiones que marcan el rumbo de un proyecto.

Este cambio no es una excusa para bajar el listón. Al contrario, **la exigencia intelectual aumenta**: si una IA puede replicar el trabajo de un programador junior, el senior debe aportar mucho más que experiencia sintáctica. Debe ser el responsable de que el software sea escalable, seguro y mantenible, incluso en un entorno donde la velocidad de desarrollo ya no es un diferenciador, sino una expectativa básica.

Para navegar esta nueva realidad, los desarrolladores y equipos técnicos necesitan:

- **Dominar la psicología del alto rendimiento**: El estado de flujo y el trabajo profundo ya no son lujos personales, sino herramientas de supervivencia en un mercado donde el ruido informativo es constante.
- **Convertirse en arquitectos de intención**: No basta con saber qué construir; hay que entender cómo guiar a un sistema que escribe por nosotros, cómo estructurar contexto, diseñar pipelines de información y definir reglas de interacción humano-IA.
- **Asumir responsabilidad profesional radical**: Aceptar que la IA no resuelve problemas de arquitectura, ni sustituye validaciones de seguridad, ni previene deuda técnica por sí sola. El profesional que lidere esta era será quien audite, corrija y mantenga el control de lo que la máquina produce.

El futuro del software no se definirá por quién tenga el mejor modelo, sino por quién sepa **convertir esa capacidad bruta en soluciones sostenibles**. Las empresas que sobrevivan no serán las que generen más líneas de código, sino las que entiendan que la velocidad es inútil sin dirección, y que la IA es una herramienta de alto impacto solo en manos de equipos que saben cómo gobernarla.

Estamos entrando en una etapa en la que el talento humano vale más que nunca, no por su capacidad de escribir código, sino por su **criterio, ética y capacidad para domar la**

complejidad. El vibe coding nos da un superpoder: transformar ideas en productos reales a una velocidad sin precedentes. Pero como toda tecnología poderosa, exige más madurez, más rigor y más visión.

Si algo queda claro tras este análisis es que el futuro del programador, y de toda persona que lidere tecnología, es el de **un estratega y diseñador de intención**. No somos reemplazados, somos elevados. Lo que distingue a los profesionales de élite será su habilidad para crear orden en un panorama de posibilidades infinitas, guiar sistemas autónomos sin perder el control, y marcar el rumbo en un sector donde la única constante es el cambio.

<sobre este whitepaper>

Este documento es el resultado de meses de trabajo real con IA, no de repetir lo que ya dicen todos. Aquí encontrarás lo que está cambiando de verdad en el desarrollo de software: cómo trabajamos, qué roles hacen falta, qué riesgos existen y qué oportunidades hay para quienes se atrevan a adaptarse..

<it's all'bout> vibecoding



A. PACHECO

FOUNDER @ RLHF TECH (STEALTH MODE)

