

Apuntes

**Curso de
Desarrollo
con IA**

DE 0 A PRODUCCIÓN

DÍA 1

Índice

El Nuevo Paradigma del Sector Tecnológico	3
Realidad del Sector en 2026	3
SWE-Bench verificado	4
¿Cuál es el impacto real de la IA?	4
Las claves	5
¿Qué pasa con los despidos?	5
Los datos	6
La realidad	8
Los problemas	8
Fundamentos Técnicos de los Modelos de Lenguaje (LLM)	9
Conceptos clave	9
Elección de modelo	12
Ingeniería de Prompts de alto valor	13
Análisis estructural del "Prompt Perfecto"	13
El mal prompt	13
El buen prompt	14
Fine-tuning	15
Metodologías de Aprendizaje y Recuperación (RAG)	15
IA para estudiar	15
Vibe coding vs. Ingeniería de software	18
La manera de Programar está cambiando muy rápido...	19
La nueva forma de programar con IA	19
Los Agentes	19
MultiAgentes: Git Worktrees	20
La Ventana de Contexto	20
Controlar el Contexto es clave para ser eficientes	21
Tipos de metodologías con Spec-Driven Development	21
Herramientas y Flujo de Trabajo Agéntico	22
Conclusiones	23
Súmame a nuestro directo del día 2	24

El Nuevo Paradigma del Sector Tecnológico

Realidad del Sector en 2026

La industria del software ha consolidado una transformación estructural donde la Inteligencia Artificial ha dejado de ser un accesorio para convertirse en el motor central de la competitividad.

Para el Arquitecto de Software moderno, la IA no es un simple "copiloto", sino una infraestructura que redefine el Ciclo de Vida del Desarrollo (SDLC). No se trata de escribir código a mayor velocidad, sino de liderar un ecosistema donde la ejecución manual se delega a modelos generativos, mientras el humano asume la responsabilidad del diseño, la integridad y la escalabilidad del sistema.

La magnitud de este cambio está respaldada por métricas de mercado críticas recopiladas por **Google, Microsoft** y **GitHub**:



SWE-Bench verificado

Un subconjunto verificado de 500 problemas de ingeniería de software de problemas reales de GitHub, validados por anotadores humanos para evaluar la capacidad de los modelos de lenguaje para resolver problemas de codificación del mundo real mediante la generación de parches para bases de código Python.



<https://llm-stats.com/benchmarks/swe-bench-verified>

¿Cuál es el impacto real de la IA?

La IA ha reconfigurado cuatro dimensiones críticas de nuestra profesión:



Educación

Procesos de estudio más eficientes.



Desarrollo

Incremento de la velocidad en todo el ciclo de vida del producto.



Profesional

Oportunidades laborales ante el cambio de paradigma.



Futuro

Evolución tecnológica a nivel transversal.

Las claves

- Crear software no es programar.
- Especializarse ya no es suficiente.
- La IA no puede dominarte.
- Utilizar IA no es opcional.

¿Qué pasa con los despidos?

Es imperativo analizar con rigor el fenómeno de los despidos masivos. La realidad del mercado dicta que muchas empresas han utilizado la "excusa de la IA" para realizar reajustes estructurales tras la sobrecontratación del periodo COVID. Sin embargo, paradójicamente, existe una escasez de profesionales cualificados. El mercado no demanda usuarios de chat, sino arquitectos y orquestadores con fundamentos de ingeniería inquebrantables.

Did A.I. Take Your Job? Or Was Your Employer 'A.I.-Washing'?

As companies blame artificial intelligence for layoffs, skeptics wonder whether that explanation captures the full picture.

The New York Times

Companies are blaming AI for job cuts. Critics say it's a 'good excuse'

CNBC

Jack Dorsey's 4,000 Job Cuts at Block Arouse Suspicions of AI-Washing

Bloomberg

Sam Altman says the quiet part out loud, confirming some companies are 'AI washing' by blaming unrelated layoffs on the technology

FORTUNE

Como afirma Andrew Ng, líder en el ámbito de la IA en Stanford: "**Aconsejar a otros que dejen de aprender a escribir código porque la IA lo va a automatizar será recordado como uno de los peores consejos profesionales de la historia**".

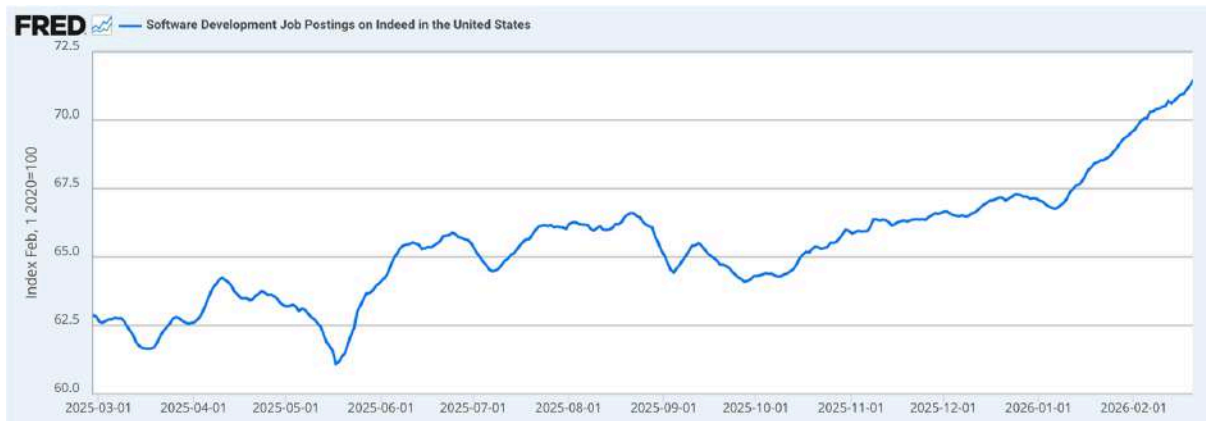
La programación sigue siendo el lenguaje del control; la IA es simplemente el motor de ejecución.

La necesidad de adaptación es inminente, pero esta solo es posible si se dominan los pilares técnicos que sostienen esta tecnología.

Los datos

La realidad de la programación en 2025: más demanda y más contratación

¿Por qué, a pesar de la oferta de empleos estos no se cubren? ¿Cuál es la razón?



Brais Moure
@MoureDev

¿En serio el CEO de Replit (plataforma de vibe coding) ha dicho "Los programadores se pierden en los detalles, no tener experiencia en programación se vuelve una ventaja"?



Guillermo Rauch  
@rauchg

Traducido del inglés al Google

No saber codificar te da una ventaja, es una completa tontería.

Cuanto más comprendas, mejores serán tus indicaciones, mejores serán los comentarios que des y mejor producto entregarás.

Lo que cambiará es que las complejidades de la sintaxis, los compiladores, los sistemas de módulos y los detalles más finos de los sistemas de tipos no importarán tanto a todos.

Pero es fundamental comprender cómo encajan las piezas, desde las llamadas al sistema hasta los píxeles. Aprenda cómo fluyen los datos para proteger sus sistemas. Aprenda sobre el rendimiento para optimizar su agente. Aprenda sobre las API para integrar sistemas. Aprenda sobre cómo fallan los sistemas para crear programas confiables.

Califica esta traducción:  

8:15 p. m. · 8 mar. 2026 · **209,1 mil** Visualizaciones

 292  579  5 mil  835 

Entonces la IA no me va a quitar el trabajo como programador... Eso depende de ti

La realidad



El sector ha cambiado para siempre.



Los fundamentos son más importantes que nunca.



Usar IA no es preguntarle a ChatGPT.



Las empresas ya no buscan programadores, buscan arquitectos y orquestadores.

Los problemas



Por donde empezar



Es muy complicado



Es solo para seniors



Esto es el futuro



Fundamentos Técnicos de los Modelos de Lenguaje (LLM)

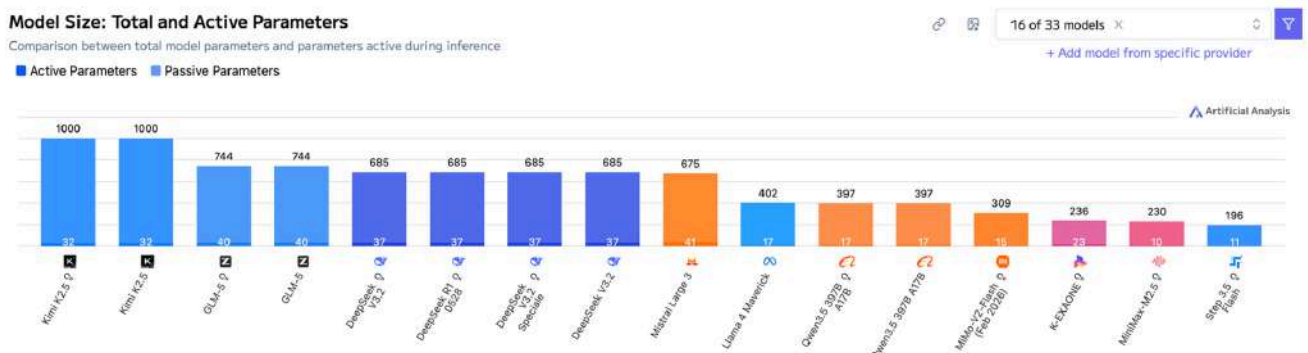
Para el arquitecto de software moderno, entender "qué hay bajo el capó" de un LLM no es opcional. Es la única forma de evitar el "vibe coding" (programar por intuición superficial) y convertirse en un piloto capaz de diagnosticar y optimizar el rendimiento de los modelos en producción.

Conceptos clave

LLM (Large Language Model): Entidad matemática diseñada para procesar y predecir texto/código. Funciona en dos fases: **Entrenamiento** (ingesta masiva de datos para aprender patrones) y **Predicción** (probabilidad estadística de la siguiente secuencia de información). No razona, predice basándose en lo aprendido.

Parámetros: Son las "conexiones neuronales" del modelo que almacenan lo que ha aprendido. A mayor número de parámetros (ej. modelos de 70B frente a 1B), mayor es la finura para representar conceptos abstractos y complejos.

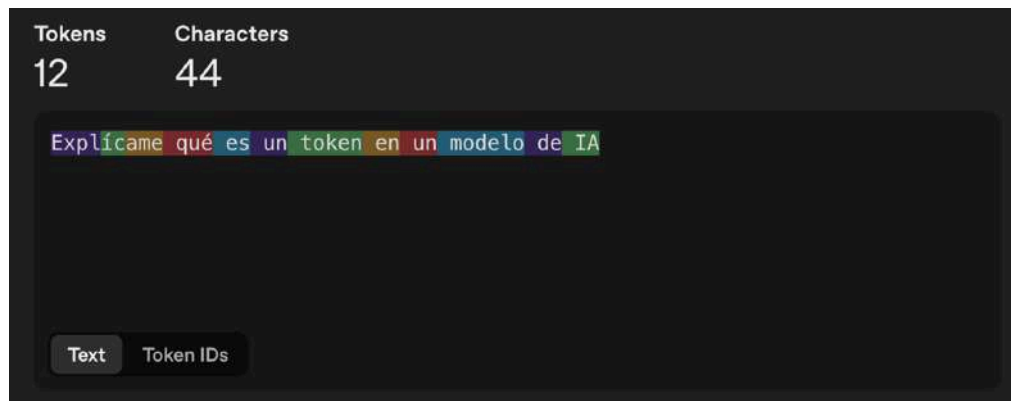
entrada → multiplicaciones por parámetros → resultado
 Más parámetros = más capacidad de representar conocimiento



<https://artificialanalysis.ai/models/open-source/large>

Tokens: Son las piezas de Lego en las que el modelo divide el texto (pueden ser palabras enteras, sílabas o letras).

Los LLM no leen palabras, leen y calculan tokens, y es la medida que se usa para cobrar su uso.



<https://platform.openai.com/tokenizer>

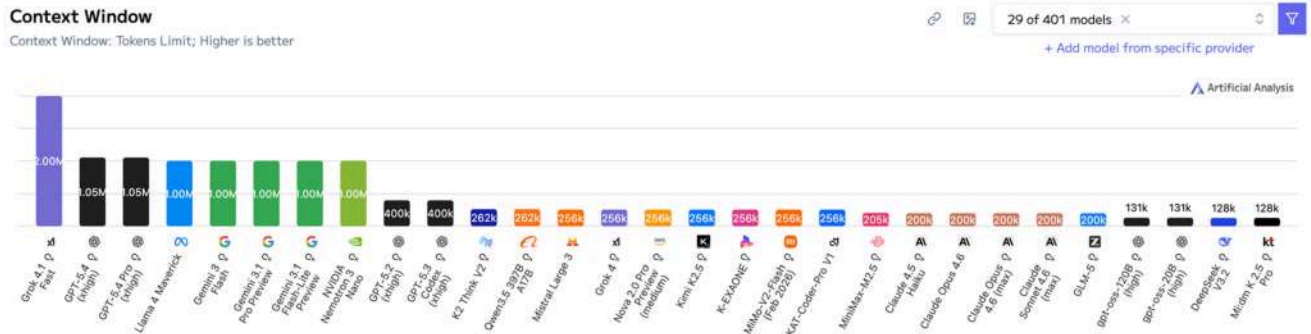
Costes: Lo que cuesta cada vez que genera una respuesta, normalmente cobrado por número de tokens.

GPT-5.4	GPT-5 mini
Nuestro modelo más competente para el trabajo profesional	Una versión más rápida y económica de GPT-5 para tareas bien definidas
Precio	Precio
Entrada: 2,50 US\$/millón de tokens	Entrada: 0,250 US\$/millón de tokens
Entrada en caché: 0,25 US\$/millón de tokens	Entrada en caché: 0,025 US\$/millón de tokens
Salida: 15,00 US\$/millón de tokens	Salida: 2,000 US\$/millón de tokens

<https://developers.openai.com/api/docs/pricing>

Ventana de Contexto: Es la "memoria a corto plazo" del modelo en una conversación. Define la cantidad máxima de texto (tokens) que el modelo puede leer, procesar y recordar a la vez sin "olvidar" el principio del documento o la charla. Si un proyecto excede este límite, la IA empezará a "olvidar" requisitos

previos o a alucinar, lo que hace indispensable una gestión estratégica de la información enviada al modelo.



<https://artificialanalysis.ai/models#context-window>

Multimodalidad: Un modelo multimodal puede "ver" imágenes, "escuchar" audios, analizar vídeos.

Latencia: Es el tiempo de espera. Se mide desde que pulsas "Enviar" hasta que el modelo responde el primer token.

Alucinaciones: Es el mayor defecto de los LLM. Ocurre cuando el modelo no sabe la respuesta y se inventa información.

Razonamiento vs. Rapidez: Diferencia entre modelos convencionales y los nuevos modelos de razonamiento que pausan, planifican y verifican internamente sus pasos lógicos antes de darte la respuesta final.

Inteligencia: Es la capacidad de un sistema para comprender información, razonar sobre ella, aprender patrones y generar respuestas o acciones útiles para resolver tareas.



<https://artificialanalysis.ai/models#intelligence>

Elección de modelo

La soberanía de los datos y la eficiencia de los recursos obligan al ingeniero a decidir entre dos ecosistemas de ejecución:

Ecosistema Local

Permite ejecutar modelos **Open-Source** directamente en la infraestructura propia, garantizando privacidad absoluta y eliminando costos por uso.

- **LM Studio**: Interfaz visual para configurar y descargar modelos como Llama o Qwen, permitiendo ajustar parámetros de ejecución localmente.
- **Ollama**: Herramienta de terminal optimizada para gestionar modelos abiertos, ideal para flujos de trabajo integrados en entornos de desarrollo Linux/macOS.

Comparativa de Privacidad y Estrategia

- **Modelos Privativos (Cloud)**: Ofrecen potencia de vanguardia y ventanas de contexto más amplias, pero implican el envío de datos a servidores de terceros, lo que puede ser un riesgo en entornos corporativos sensibles. Ej. GPT-5.4, Claude Sonnet 4.6, Gemini.
- **Modelos Abiertos (Local)**: Son la opción preferida para el manejo de lógica propietaria o datos sensibles, permitiendo una personalización total sin dependencia de APIs externas.

<https://artificialanalysis.ai/leaderboards/models>

<https://openrouter.ai/rankings>

<https://ollama.com/search>

<https://lmstudio.ai/models>

Ingeniería de Prompts de alto valor

La calidad del software que entrega una IA es un reflejo exacto de la arquitectura de la instrucción suministrada. El axioma "Garbage in, Garbage out" nunca ha sido tan vigente como en la era de los LLM.

Análisis estructural del "Prompt Perfecto"

Una instrucción profesional debe articularse bajo cinco ejes:

1. **Rol (¿Quién soy?):** Define el nivel de experiencia y la especialidad que quieres que asuma la IA.
2. **Contexto (¿Dónde estamos?):** Explica de qué trata el proyecto, tecnologías y cuál es el problema general.
3. **Tarea exacta (¿Qué necesitas?):** Sé específico. En lugar de pedir un "sistema" algo concreto.
4. **Restricciones o Reglas (¿Qué límites hay?):** Indica convenciones y estándares.
5. **Formato de salida (¿Cómo lo quieres?):** Cómo quieres recibir la información.

El mal prompt

"Haz un código para un login en Python."

- No hay contexto: No sé si es para una web, una app de consola o un microservicio.

- No hay tecnologías específicas: Podría usar Flask, Django, FastAPI o Python puro.
- No hay reglas de seguridad: Podría darte un código que guarde las contraseñas en texto plano (una pésima práctica).
- El modelo tendrá que "adivinar" lo que quieres, y probablemente fallará en integrarse con tu proyecto.

El buen prompt

[1. Rol]

Actúa como un desarrollador backend Senior especializado en ciberseguridad.

[2. Contexto]

Estoy construyendo una API REST para un e-commerce usando Python 3.10.

[3. Tarea]

Necesito que escribas un endpoint para el login de usuarios.

[4. Restricciones]

El endpoint debe estar hecho con el framework FastAPI. Debe recibir un email y una contraseña, validar las credenciales simulando una consulta a una base de datos PostgreSQL y devolver un token JWT. Asegúrate de incluir el manejo de errores (por ejemplo, devolver un error 401 si las credenciales fallan) y usa contraseñas hasheadas con bcrypt.

[5. Formato]

Devuélveme únicamente el bloque de código bien comentado, sin explicaciones previas ni introducciones.

<https://platform.openai.com/chat>

<https://aistudio.google.com>

Fine-tuning

Es elegir un modelo general (que sabe de todo un poco) y darle un entrenamiento extra con ejemplos muy específicos para que se vuelva un experto total en una sola tarea, como redactar contratos legales o escribir código en un lenguaje concreto.

Metodologías de Aprendizaje y Recuperación (RAG)

La Generación Aumentada por Recuperación (RAG) es el "Guardrail of Truth" (Guardarraíl de la Verdad) de la IA. Permite anclar el conocimiento del modelo a documentos específicos, eliminando alucinaciones y forzando a la IA a operar dentro de una base de conocimiento verificada.

IA para estudiar

Optimización del Estudio con NotebookLM

[NotebookLM](#) representa la cúspide del uso de RAG para el aprendizaje activo.

Con NotebookLM, tú decides cuáles son las fuentes de información (vídeos de YouTube, documentación técnica, PDFs, sitios web), acotando el conocimiento de la IA para obtener respuestas precisas y confiables.

Guía Paso a Paso para Dominar NotebookLM

Paso 1: Crea tu Cuaderno de Conocimiento

El primer paso es el más sencillo: **crear un nuevo cuaderno en NotebookLM**. Al hacerlo, te encontrarás con un espacio completamente vacío. Este cuaderno es tu "lienzo en blanco", un entorno de IA donde tú, y solo tú, tienes el control total.

Paso 2: Alimenta tu IA con Fuentes Confiables

Ahora es el momento de añadir las "**fuentes**" que conformarán la base de conocimiento de tu IA. NotebookLM es increíblemente versátil y te permite añadir información de diversos formatos.

- Videos de YouTube: Pega un enlace y la IA procesará la transcripción completa.
- Sitios Web: Proporciona una URL para que analice el contenido de una página.
- Documentos de Google Drive: Conecta tu cuenta y añade textos, presentaciones, etc.
- Archivos PDF y de Texto: Sube documentos directamente desde tu ordenador.

Una vez que tus fuentes están cargadas, comienza la verdadera magia: el diálogo inteligente con tus datos.

Paso 3: Dialoga con tus datos a través del Chat

Con tus fuentes listas, puedes empezar a hacer preguntas en el chat. La IA responderá basándose exclusivamente en la información que le proporcionaste.

La característica más importante de estas respuestas son las **citas numeradas** que aparecen junto a cada afirmación. Estos números **[1], [2]**, etc., son enlaces que te llevan al punto exacto de la fuente (ya sea el minuto de un video o el párrafo de una web) de donde se extrajo la información. Esto elimina por completo las alucinaciones y te da total confianza en la respuesta.

Paso 4: Otras herramientas de NotebookLM

NotebookLM va mucho más allá de un simple chat. Ofrece un conjunto de herramientas diseñadas para transformar el consumo pasivo de datos en una experiencia de aprendizaje activa. Estas funciones se adaptan a diferentes estilos y situaciones, permitiéndote sintetizar la información de formas innovadoras.

Herramienta	Descripción	Beneficio para el Aprendizaje
Mapa Mental	Organiza visualmente los conceptos clave, características y recursos de cada lenguaje.	Te permite comparar de un solo vistazo las estructuras de ambos lenguajes, facilitando la identificación de similitudes y diferencias.
Resumen de Video/Audio	Genera un podcast o un video corto (ej. de 5 minutos) que resume toda la información de las fuentes.	Perfecto para repasar conceptos clave mientras viajas o haces otras actividades. Condensa horas de material en un formato fácil de consumir.
Tarjetas Didácticas (Flashcards)	Crea automáticamente una baraja de tarjetas con preguntas y respuestas sobre los conceptos fundamentales (ej. "68 tarjetas para conceptos clave").	Ideal para reforzar el aprendizaje activo y autoevaluar tu conocimiento sobre los temas más importantes de cada tecnología.
Cuestionario	Genera una prueba formal (ej. de 15 preguntas) para evaluar tu comprensión sobre el material cargado.	Te ayuda a medir de manera objetiva cuánto has aprendido y a identificar las áreas que necesitas reforzar antes de tomar una decisión.

Vibe coding vs. Ingeniería de software

- Vibe coding: programar guiándose por intuición y ayuda de IA para generar código rápido.
- Ingeniería de software: diseñar y construir software de forma sistemática con arquitectura, control, pruebas y mantenimiento a largo plazo.



La manera de Programar está cambiando muy rápido...

En menos de 3 años...

- Utilizamos la IA con el autocompletado.
- Fueron apareciendo herramientas que cada vez generaban más código, la revolución de las CLI (Claude, OpenCode).
- La época multiagente: Múltiples IA trabajando en paralelo

La nueva forma de programar con IA

- Cada vez las especificaciones claras cobran más importancia.
- Son la forma en la que nos comunicamos con la IA.
- Empezamos con técnicas de prompting, para acabar llegando al **Spec-Driven Development**.

Los Agentes

- Un agente de IA es un sistema que puede percibir, decidir y actuar para cumplir un objetivo.
- Utiliza modelos de IA (LLMs) para razonar sobre qué hacer.
- Puede usar herramientas externas (APIs, bases de datos, código).
- Opera de forma semi-autónoma o autónoma.

Ejemplos de Agentes:

Generador de Tests

Qué hace:



Analizar funciones



Genera tests unitarios



Ejecuta los tests

Código:

```
def sum(a, b):  
    return a + b
```

El agente genera:

```
def test_sum():  
    assert sum(2,3) == 5
```

MultiAgentes: Git Worktrees

En entornos multi-agente, surge el problema de la concurrencia: múltiples IAs intentando mutar los mismos archivos simultáneamente.

La solución arquitectónica son los **Git Worktrees**.

Permiten que cada agente trabaje en una carpeta física independiente vinculada a diferentes ramas, eliminando las colisiones de agentes y permitiendo una orquestación asíncrona y segura.



La Ventana de Contexto

Al igual que los humanos, las IA pueden perder el foco, e incluso olvidarse de ciertas cosas.

La memoria de las IA, aunque grande, es limitada.

La IA a veces "simplifica" el contexto, lo que se conoce como Compresión de Contexto. Algunos ejemplos: Summarization, Sliding Windows, Compresión Jerárquica e incluso Guardados en Memoria.

Controlar el Contexto es clave para ser eficientes

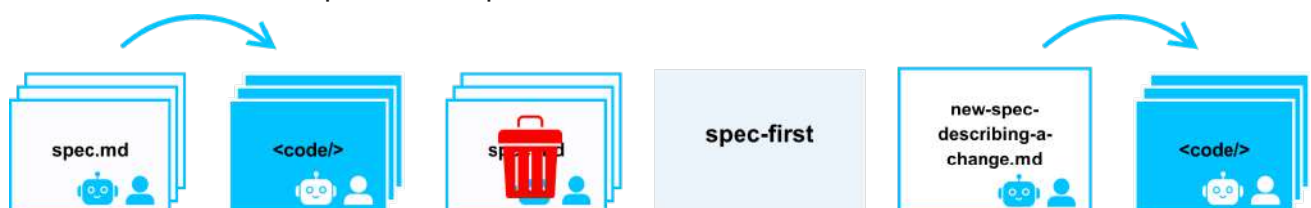
Para utilizar la IA en sus máximas capacidades, podemos realizar distintas técnicas para gestionar el contexto en proyectos tanto grandes como pequeños.

Una de las metodologías que está apareciendo es el **Spec-Driven Development (SDD)**.

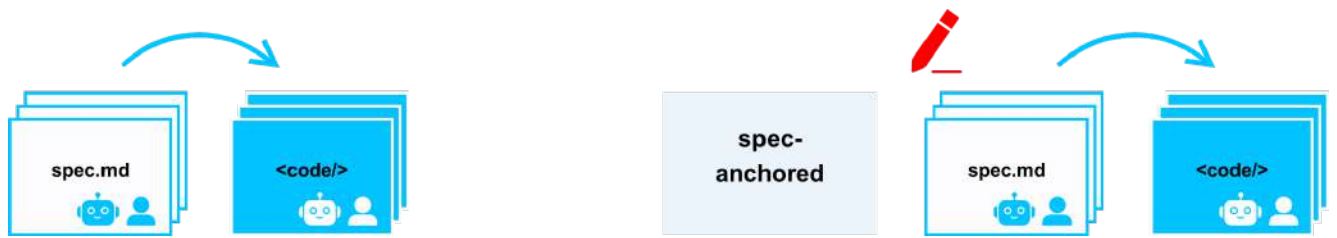
El **SDD** es la metodología definitiva que devuelve el control al ingeniero. Desplaza el foco de la "escritura de código" hacia la "validación de la intención". En este flujo, el humano actúa como el Validador de Intentos, mientras que el código se convierte en un artefacto transitorio y derivado.

Tipos de metodologías con Spec-Driven Development

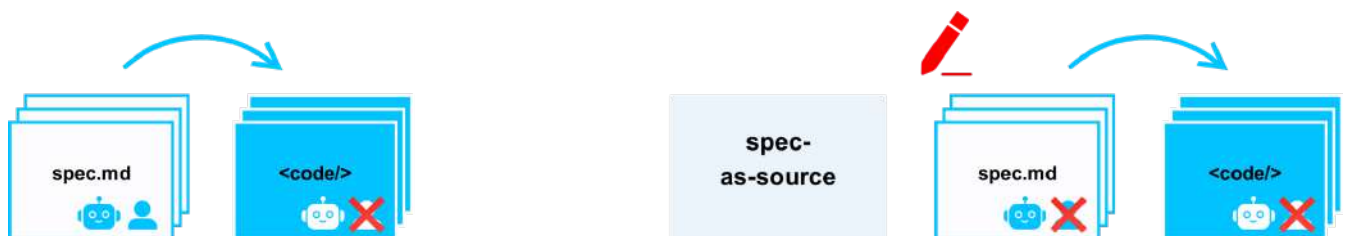
Spec First: Nosotros definimos los requerimientos y la IA genera el código. Para nuevos desarrollos, repetimos el proceso.



Spec Anchored: Los requerimientos y los documentos de diseño evolucionan con el código, la propia IA los actualiza. Son documentos vivos.



Spec As Source: Nosotros "desarrollamos" en el requerimiento, todo el código es autogenerado y no se espera que un humano lo edite directamente.



Herramientas y Flujo de Trabajo Agéntico

Las herramientas "AI-First" están liderando la transición hacia un desarrollo donde el código desaparece de la vista principal del ingeniero.

Evaluación de Ecosistemas



Cursor: Editor pionero que integra un "modo agente" capaz de leer el contexto completo del repositorio y proponer cambios estructurales, no solo líneas sueltas.



Claude Code (CLI): La revolución de la terminal agéntica. Permite al desarrollador delegar tareas complejas (ej. "ejecuta los tests y arregla lo que falle") directamente desde el prompt de la terminal.



Kiro: Especializado en SDD, divide el flujo en Requerimientos, Diseño y Tareas, forzando al desarrollador a pensar antes de ejecutar.



Codex y Antigraivty: Herramientas que apuntan a la desaparición visual del código. El desarrollador interactúa con especificaciones de alto nivel mientras la IA gestiona los archivos de implementación en segundo plano.

Conclusiones

Las herramientas son efímeras; los fundamentos son permanentes.

El éxito en 2026 no depende de qué IA uses, sino de tu capacidad para mantener la arquitectura, las pruebas y el mantenimiento por encima del ruido tecnológico.

El control final siempre pertenece al profesional.

BIG school



**Súmate a nuestro
directo del día 2**

ACCEDE AQUÍ