BIG school {m}

CURSO DE DESARROLLO CON IA

Programa con agentes

Índice

Algunos datos en 2026…

Nos encontramos ante el cambio estratégico más profundo de la última década en el sector tecnológico: el tránsito definitivo del acto de "escribir código" al de "dirigir inteligencia artificial". No estamos ante la obsolescencia del programador, sino ante su evolución hacia un rol de supervisor crítico, donde la IA genera el volumen y el humano garantiza la veracidad y la seguridad.

Developers

Utilizan IA para programar en su día a día

Código

Generado está escrito por IA

Confianza

En los resultados producidos por IA sin supervisión

Datos extraídos de diferentes informes de Google, Microsoft, GitHub y Stack Overflow.

¿Cuál es el impacto real de la IA?

Educación

La barrera de entrada nunca fue tan baja: hoy puedes construir cosas reales desde el primer día.

Desarrollo

El rol cambió de teclear código a dirigirlo: el valor está en decidir y revisar, no sólo en la sintaxis.

Empleo

Las ofertas ya no piden "saber programar", piden saber programar con fundamentos + IA.

Futuro

No te va a sustituir una IA, te va a sustituir alguien que sabe usarla mejor que tú.

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

¿Qué pasa con las noticias?

Jensen Huang lo tiene claro: a estas alturas nadie debería aprender a programar, ya lo hará la IA por nosotros

Xataka

Según Mark Zuckerberg, Meta está desarrollando una IA capaz de escribir código como un ingeniero de nivel medio.

Entrepreneur

Dario Amodei, CEO de Anthropic, lanza la advertencia definitiva sobre la IA: "Estará lista para programar por completo en 2025 o 2026"

Computer Hoy

TECNOLOGÍA

El director ejecutivo de Klarna afirma que la IA ayudó a la empresa a reducir su plantilla en un 40%.

CNBC

Esto es lo que verdaderamente pasa…

Mark Zuckerberg admite que Meta "cometió errores" mientras la IA transforma el 20% de su plantilla.

Sam Altman advirtió que la IA podría provocar un "apocalipsis laboral". Ahora dice estar "encantado de haberse equivocado"

Entrepreneur

FORTUNE

Sam Altman y Dario Amodei están retractándose de sus profecías sobre el apocalipsis laboral provocado por la IA, mientras buscan realizar salidas a bolsa de gran envergadura.

Jensen Huang acusa a CEOs tech de tener un complejo de dios con la IA

"Ridículos": Jensen Huang, CEO de Nvidia, explotó contra quienes dicen que la IA destruirá empleos

TECNO NEWSROOM

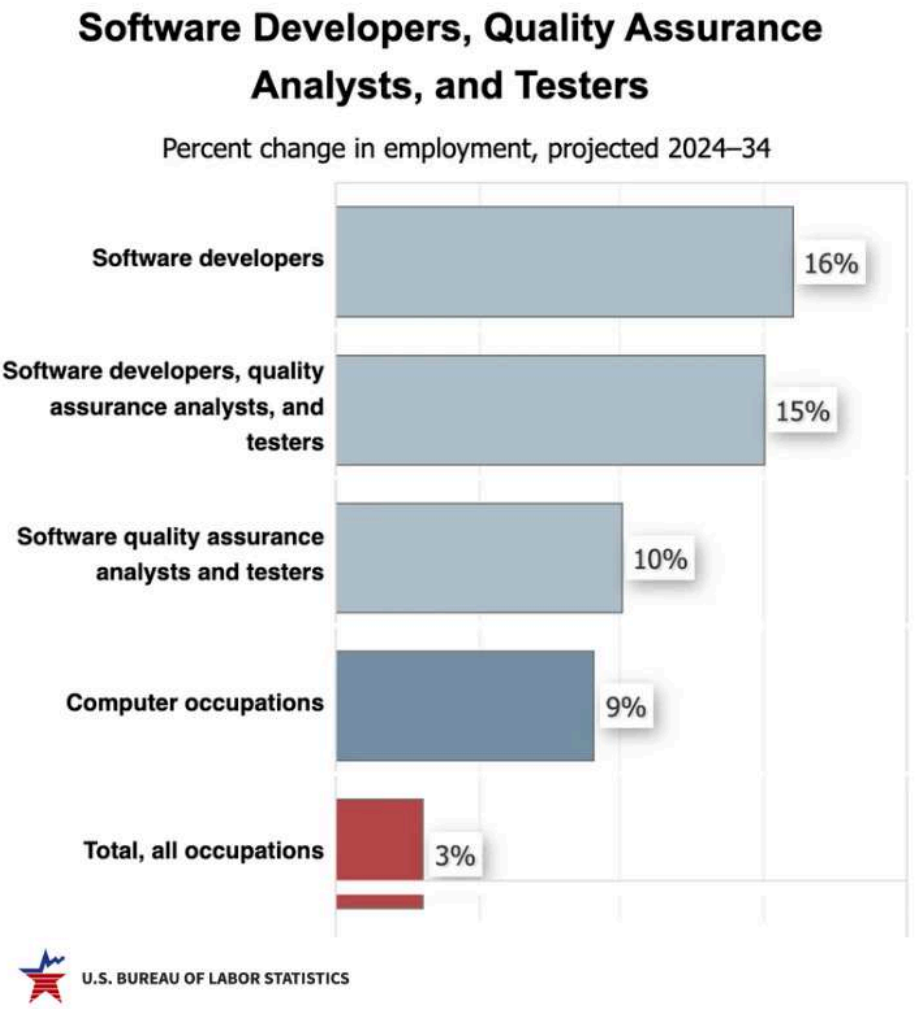
mechatronic NOTICIAS

Klarna recontrata humanos tras sustituirlos con IA, y no es la única empresa que pasa por lo mismo

WIRED

'Díganle que es un imbécil': La nueva unidad de IA de Meta es un desastre total.

Los datos


Software Developers, Quality Assurance Analysts, and Testers
Percent change in employment, projected 2024–34
Software developers
16%
Software developers, quality assurance analysts, and testers
15%
Software quality assurance analysts and testers
10%
Computer occupations
9%
Total, all occupations
3%
U.S. BUREAU OF LABOR STATISTICS

europapress / portaltic / empresas

El 92% de las empresas tecnológicas españolas no encuentra los perfiles que necesita

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

Las claves

- Crear software no es programar.
- Los fundamentos importan más que nunca.
- Tú diriges, la IA ejecuta.
- Utilizar IA no es opcional.

La realidad

El sector ha cambiado para siempre.

Los fundamentos importan más que nunca.

Usar IA no es chatear. Es dirigir.

El mercado ya no paga por teclear, sino por orquestar.

Los problemas

Por donde empezar

Es muy complicado

Es solo para seniors

Esto es el futuro

Fundamentos de la Inteligencia Artificial

¿Qué es un LLM y cómo funciona?

Large Language Model (LLM): Tipo de IA diseñada para entender y escribir texto de manera parecida a la nuestra.

Base matemática: No piensa o razona como el ser humano. Funciona en dos grandes pasos.

Entrenamiento: Lee una cantidad inmensa de información para aprender cómo se relacionan las palabras entre sí.

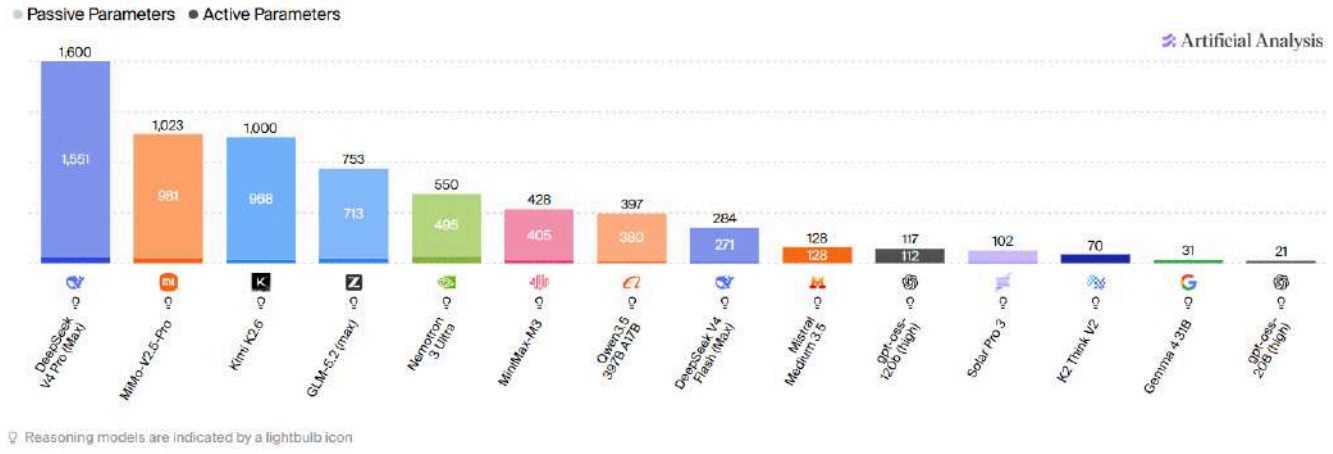
Predicción: No busca en una base de datos ni reflexiona. Calcula matemáticamente la probabilidad de la siguiente.

Predice el siguiente token. No "piensa", calcula probabilidades.

Conceptos clave

Número de parámetros: Son las "conexiones neuronales" del modelo que almacenan lo que ha aprendido.

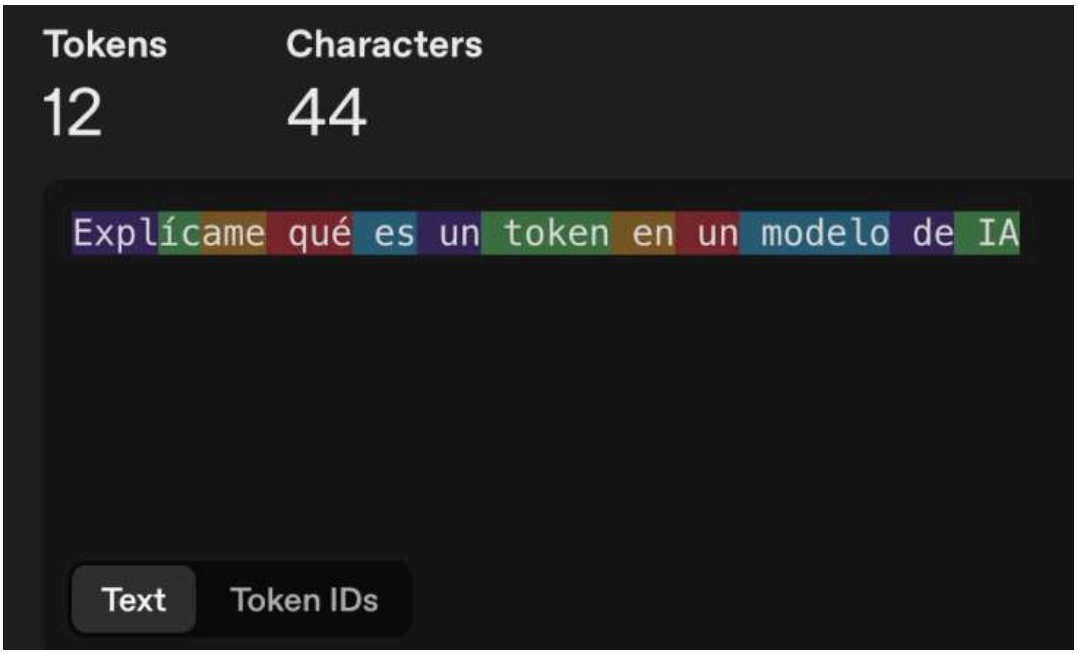
Más parámetros = más capacidad de representar conocimiento

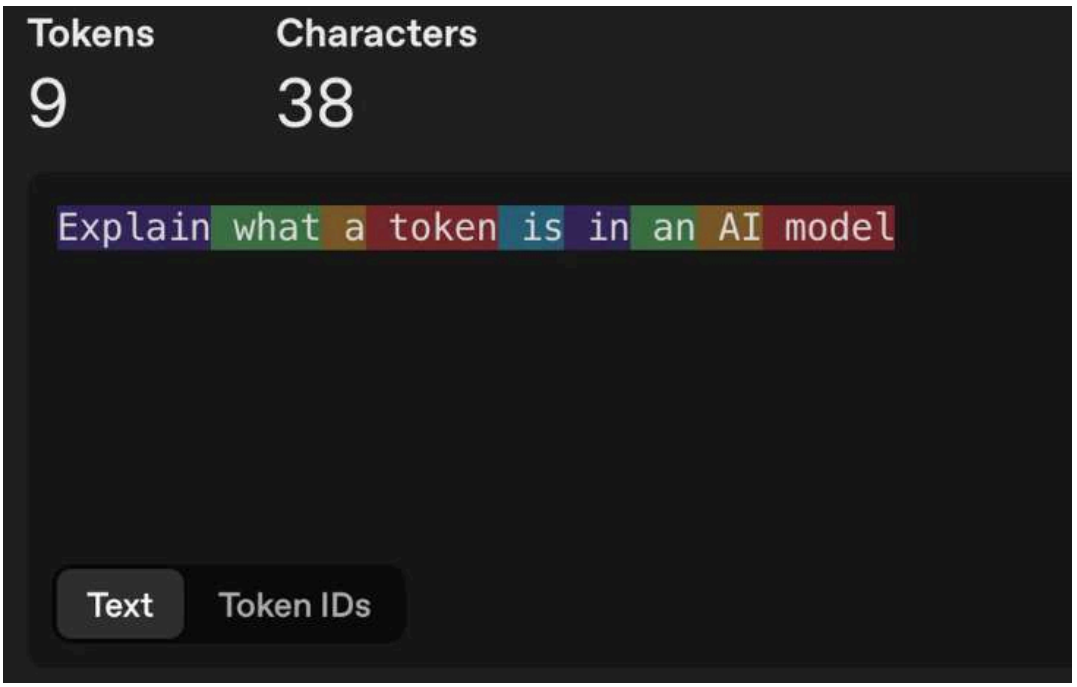


https://artificialanalysis.ai/models#model-size-tabs

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.



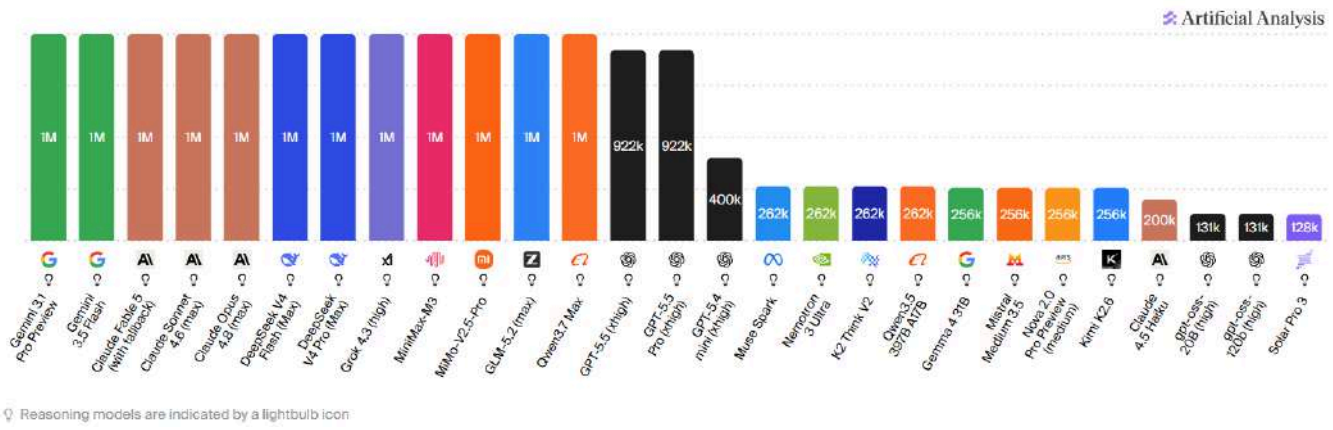


https://platform.openai.com/tokenizer

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

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

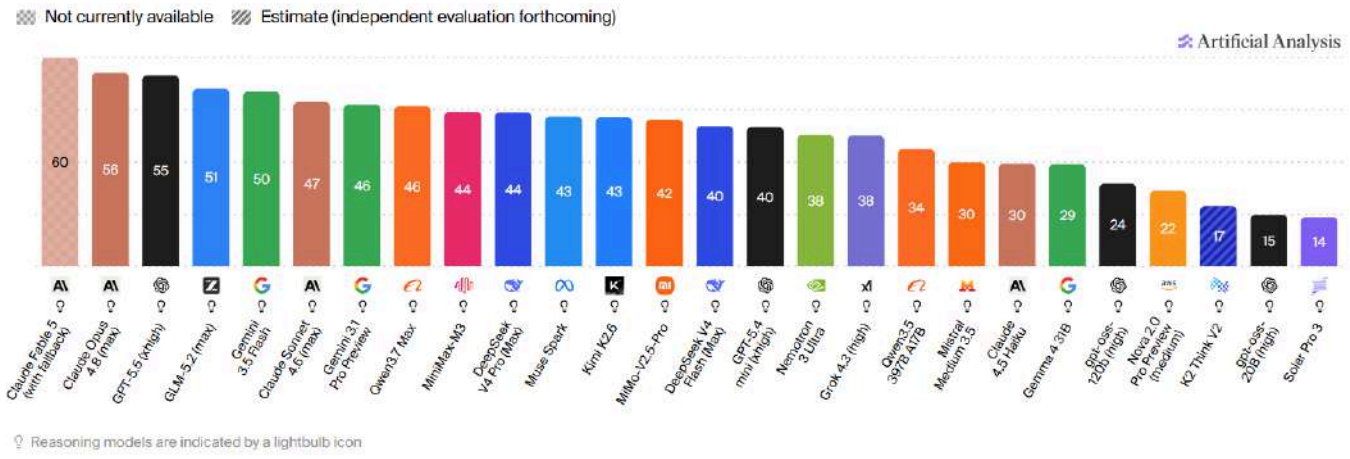
Multimodalidad: Un modelo multimodal puede "ver" imágenes, "escuchar" audios, analizar vídeos.

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

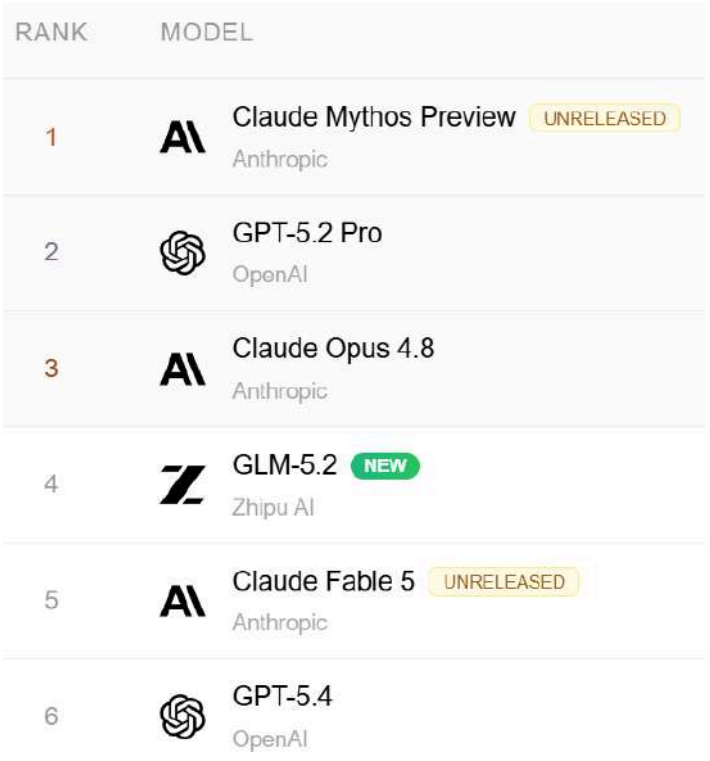
Elección de modelo

- **Fundamentos**: Parámetros, tokens, coste, ventana de contexto, latencia, inteligencia…
- **Ejecución**: Local (LM Studio / Ollama) vs. Cloud
- **Privacidad**: Open-source vs. privativo

https://artificialanalysis.ai/leaderboards/models

https://openrouter.ai/rankings

Modelos… ¿Cuál para qué?



https://llm-stats.com/

Potencia ↔ Velocidad ↔ Coste: Elegir según tarea.

- Modelo potente para pensar y arquitectura.
- Modelo medio para el día a día.
- Modelo rápido y barato para tareas mecánicas.

Anatomía de un prompt (Prompt engineering)

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 3.14.

[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)

Es una técnica para evitar las alucinaciones. Consiste en conectar el LLM a un buscador de internet o a los documentos privados de una empresa para que lea esa información real antes de responderte.

Vibe coding vs. Ingeniería de software

- **Vibe coding**: programar guiándose por intuición y ayuda de IA para generar código rápido. Describir y aceptar sin revisar. Rápido, divertido y peligroso.

- **Ingeniería de software**: diseñar y construir software de forma sistemática con arquitectura, control, pruebas y mantenimiento a largo plazo.

Editores de código AI-first

Desde la irrupción masiva de la inteligencia artificial a finales de 2022, el entorno de desarrollo integrado (IDE) ha dejado de ser una simple superficie de escritura para transformarse en un socio cognitivo. En este nuevo paradigma, el editor no solo asiste en la sintaxis, sino que participa en la lógica y la toma de decisiones arquitectónicas. Como arquitectos, debemos entender que no estamos ante una mejora incremental, sino ante un cambio en la naturaleza misma de la interacción hombre-máquina.

Evolución de los editores de código en la era de la IA:

- **2022** - IDEs clásicos (autocompletado)
- **2023** - Chat-based coding
- **2024** - Copilots
- **2025** - AI agents
- **2026** - AI-first development environments

Familias de editores

- **IDE-first**: VSCode (Copilot), Cursor, Antigravity, TRAE, ZED…
- **Terminal-first**: Claude Code, OpenCode, Neovim, Warp…
- **ADE (Agentic Development Environment)**: Claude app, Codex...
- **Cloud/Async**: Cursor, Codex, Claude Code, Copilot…


VS Code


Cursor


Claude Code


OpenCode


Antigravity


Codex


Warp

Otros: Zed - Trae - Kiro - Replit - Lovable - IntelliJ - Neovim

Agentes de código

Para un profesional de la ingeniería de software, seguir utilizando un chat convencional para programar es una ineficiencia estratégica. El salto cualitativo hacia los flujos agénticos permite que el desarrollador opere en un nivel de abstracción superior, validando intenciones en lugar de corregir líneas de código aisladas.

Chatbot

- Pregunta → Respuesta
- Sin acceso al sistema
- Un solo turno
- Requiere copy-paste

Agente

- Bucle autónomo
- Lee/escribe archivos
- Ejecuta y verifica
- Itera

El bucle del Agente

Primeros pasos con el editor

Checklist código seguro con IA

- **Da contexto y pide seguridad por adelantado**: di qué quieres y exige validación antes de que escriba nada.
- **Lee el código y entiéndelo entero**: revisa lo que cambió de verdad. Lo que no entiendas, pídelo explicado. No apruebes a ciegas.
- **Verifica antes de subir nada**: pasa tests y confirma que no se cuela ningún dato sensible.
- **Cuida lo que compartes**: nunca pegues claves ni datos reales de clientes en un prompt.

Prompt utilizado en el ejemplo del video:

“Crea una web que muestre una línea del tiempo desde la aparición de ChatGPT a la actualidad, donde aparezcan predicciones realizadas por CEO y figuras relevantes de la tecnología que hablen sobre cómo los programadores iban a desaparecer y cómo se equivocaron ya que sólo buscaban favorecer la expectativa por sus empresas.”

Súmate a nuestro directo del día 2

IR AL DIRECTO