Opinión

La IA y los estudiantes

Uso IA para la mayor parte de mi trabajo de programación porque antes pasé décadas aprendiendo los fundamentos. Los estudiantes y principiantes necesitan otro enfoque: usar la IA como apoyo sin cederle el razonamiento.

Fabio Akita · ailair.akitaonrails.com/es/students/

> *Yo, senior, con años de experiencia, voy a hacer 100 % vibe coding. Tú, estudiante, junior, DEBES escribir lo máximo posible a mano, dejar que la IA haga las tareas más rutinarias en paralelo, y DEBES revisarlo todo manualmente. Necesitas decenas de miles de horas de práctica deliberada y estudio para convertirte en alguien que después sepa qué pedirle a la IA que haga por ti. No antes.*
>
> *Eu, sênior, anos de experiência, vou fazer 100 % vibe codar. você, estudante, júnior, DEVE escrever o máximo possível na mão, deixar a IA fazer tarefas mais mundanas, em paralelo, e você DEVE revisar tudo manualmente. Você precisa de dezenas de milhares de horas de prática deliberada, estudo, pra se tornar alguém que, depois, vai saber o que mandar a IA fazer por você, não antes.*
>
> **Yo en X, 24 de septiembre de 2026**

1. El multiplicador

La IA multiplica lo que traes

La misma herramienta le da a un senior más trabajo bueno, más rápido, y a un principiante le da más de lo que ya tiene, errores incluidos.



1. El multiplicador

La IA multiplica lo que traes

- La IA puede multiplicar los errores cuando los fundamentos son débiles. Con fundamentos sólidos, puede multiplicar el trabajo útil.
- El informe DORA 2025 de Google encontró lo mismo en equipos: "la IA no arregla un equipo; amplifica lo que ya está ahí".
- Si tienes experiencia y todavía te resistes, para. No hay ningún mérito en escribir un Dockerfile o un Makefile a mano por centésima vez. Las tareas rutinarias van para la IA; los fundamentos no.
- Quien entiende sistemas operativos, bases de datos, redes, estructuras de datos y compiladores usa los agentes como un exoesqueleto. Quien no, los usa como muleta. Un exoesqueleto amplifica la fuerza; una muleta solo esconde la debilidad.
- Y alguien tiene que revisar lo que sale. En la prueba de Veracode de 2025 con más de 100 modelos, el 45 % de las muestras de código falló los tests de seguridad e introdujo vulnerabilidades del OWASP Top 10.

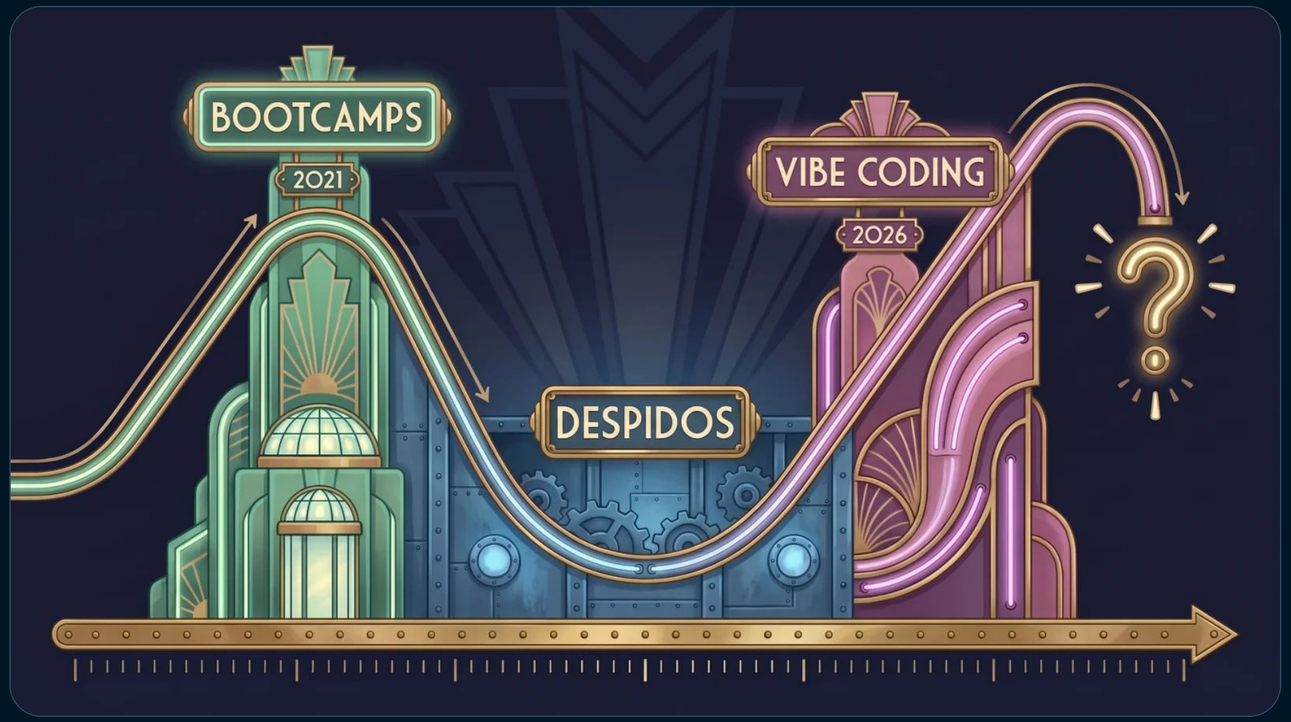
1. El multiplicador

> *Un programador bien preparado no va a ser reemplazado. Lo dije claramente: yo nunca lo voy a ser.*
>
> *Programador bem preparado não vai ser substituído. Disse claramente: "EU" nunca vou ser.*
>
> **Yo en X, 28 de septiembre de 2026**

2. La última burbuja

Este atajo ya lo vimos

Antes de la IA, la promesa era un bootcamp: dos meses y ganarías un sueldo de Google. Terminó en cierres y despidos.



2. La última burbuja

Este atajo ya lo vimos

- Los egresados de bootcamps se multiplicaron por unas ocho veces entre 2013 y 2018. Lambda School anunciaba tasas de colocación "de hasta el 86 por ciento"; la Consumer Financial Protection Bureau encontró que sus números internos estaban más cerca del 50 %, a veces tan bajos como el 30 %.
- Empecé Akitando en 2018 justamente para esto: preparar programadores de verdad para sobrevivir al estallido de la burbuja. En 2022 le decía a la gente que cancelara los cursos y dejara de creerles a los influencers.
- Ahora la misma promesa vuelve con otra cara: no necesitas aprender a programar, la IA lo hace por ti. Nadie se vuelve ingeniero en un bootcamp de un mes, y nadie se vuelve ingeniero escribiendo prompts.
- Luego se acabó el dinero barato. Cerraron bootcamps en 2023 y 2024, 2U anunció a fines de 2024 que dejaba el negocio, y la industria tecnológica recortó más de 150.000 empleos solo ese año.
- Lo que esos cursos enseñaban, formularios, CRUD, reportes y copias de la misma interfaz, es justamente lo que los LLMs replican mejor. La mayoría de esos empleos ya se recortó después de 2022.

3. La evidencia

Aprender en piloto automático no se fija

Los estudios hasta ahora apuntan en la misma dirección: cuando la IA piensa por ti, no aprendes.

17 pts

menos puntos en el quiz para ingenieros mayormente junior que aprendieron una librería nueva con IA

17 %

peores notas de examen para estudiantes que practicaron con GPT-4 sin más, una vez que se lo quitaron

19 %

más lentos: desarrolladores experimentados con IA, que igual creían ser más rápidos

3. La evidencia

Aprender en piloto automático no se fija

- El propio estudio de Anthropic, enero de 2026: ingenieros mayormente junior que aprendían una librería nueva (Trio) con IA sacaron 50 % en el quiz, contra 67 % de quienes programaron a mano, sin un ahorro de tiempo significativo. Quienes usaron la IA para que les explicara conceptos conservaron la mayor parte del aprendizaje; quienes le delegaron el código quedaron entre las notas más bajas.
- El ensayo de METR de 2025: desarrolladores open source experimentados tardaron un 19 % más con IA y, después, seguían creyendo que los había hecho un 20 % más rápidos.
- En un estudio de PNAS con unos 1.000 estudiantes de secundaria, GPT-4 sin más subió las notas de práctica un 48 % y luego bajó las de examen un 17 % cuando se lo quitaron. Un tutor hecho para obligarlos a pensar evitó en gran parte esa pérdida.
- Un preprint del MIT, aún no revisado por pares, encontró que quienes escribieron ensayos con ChatGPT tenían dificultades para citar su propio trabajo.

4. El mercado

El camino de entrada es más estrecho

Las empresas siguen contratando a gente que sabe revisar y decidir. Contratan muchos menos principiantes para teclear.

65 %

menos contratación de recién graduados en las grandes tecnológicas que en 2019

4,5 %

de las ofertas de desarrollo de software son de nivel inicial (1.er trimestre de 2026)

19 %

menos jóvenes de 22 a 25 años de lo esperado en los empleos más expuestos a la IA

El camino de entrada es más estrecho

- SignalFire, 2026: la contratación de recién graduados cayó cerca de un 65 % en las grandes tecnológicas y un 76 % en startups en etapa temprana frente a 2019, mientras que la contratación de ingenieros en las grandes tecnológicas bajó solo un 11 %.
- El estudio "Canaries" de Stanford: el empleo de jóvenes de 22 a 25 años en los trabajos más expuestos a la IA, desarrolladores de software incluidos, está un 19 % por debajo de donde estaría si hubiera seguido el ritmo de los trabajos menos expuestos, sobre todo por menos contrataciones. Los autores tienen cuidado de llamarlo descriptivo más que causal.
- Indeed: los puestos de nivel inicial eran el 4,5 % de las ofertas de software a comienzos de 2026. Las ofertas crecieron casi un 15 % después de que saliera Claude Code en febrero de 2025, y el 71 % de ese crecimiento fueron puestos senior.
- Los recién graduados en ciencias de la computación tienen una de las tasas de desempleo más altas de todas las carreras (7,0 %), pero una de las tasas de subempleo más bajas (19 %, contra 39 % en general): cuando los contratan, suele ser en un trabajo que requiere el título. El camino típico de entrada sigue siendo una licenciatura.

5. El vacío

La IA no adivina lo que necesitas

Hace lo que le pides y no hace lo que no le pediste. Saber qué pedir es el trabajo.



5. El vacío

La IA no adivina lo que necesitas

- Un agente de programación necesita que le expliques el objetivo, el contexto, las restricciones, las prioridades, lo que no se puede romper, qué caminos son aceptables y cómo se mide el éxito. Muchos programadores nunca aprendieron a decir eso.
- Incluso para una app diminuta, todo lo hecho con vibe coding que va a producción tiene que revisarlo un programador experimentado. Eso no es opcional.
- Mi forma de hablarle a un agente tiene cuatro partes: qué quiero, cómo, qué no quiero y cómo lo comprobamos. No puedes llenarlas sin conocer el dominio.

6. El mito del lenguaje

Los tests importan más que el lenguaje

En octubre de 2026 les di a 4 modelos la misma tarea difícil en 7 lenguajes. Ningún lenguaje quedó por delante.

112

intentos: 7 lenguajes, 4 modelos, 2 condiciones, 2 corridas cada una

24

resultados correctos de los dos modelos más débiles cuando les di los tests

16

resultados correctos de los mismos modelos sin los tests

Los tests importan más que el lenguaje

- Los dos modelos de frontera acertaron todos los intentos válidos, en todos los lenguajes.
- Aprende a escribir tests y convierte cada bug en uno. Esa habilidad sirve en cualquier lenguaje, con cualquier modelo.
- Cada fallo rompió una regla del propio problema. Ninguno fue un error de tipos, así que ningún compilador los habría detectado.

7. La pregunta

Si la IA puede programar todo, ¿para qué aprender?

Porque no puede programar todo. Reproduce lo que ya vio, y lo que no vio es justo donde están los próximos empleos.



7. La pregunta

Si la IA puede programar todo, ¿para qué aprender?

- Los modelos aprendieron de código público. La mayor parte del código del mundo es privado, el 81,5 % de las contribuciones solo en GitHub, y un problema que nadie resolvió antes no está en ningún conjunto de entrenamiento.
- Con el legado pasa lo mismo. No hay una gran comunidad open source de COBOL: el código que mueve bancos, aseguradoras y gobiernos es privado, y los modelos casi no lo vieron.
- Donde se publicó algo parecido, la IA lo replica rápido. Donde no, gira en bucles largos y caros, y alguien tiene que tomar el control: definir qué significa "correcto", construir el oráculo, escribir tests con objetivos claros, encontrar la referencia que nadie publicó y darse cuenta de cuándo el agente empieza a dar vueltas en círculo.
- Cada uno de esos trabajos necesita a alguien que entienda el problema desde abajo: cómo funcionan de verdad las computadoras, los sistemas operativos, los compiladores, las redes y las bases de datos. Ese es el conocimiento difícil que la mayoría se salta.

7. La pregunta

Si la IA puede programar todo, ¿para qué aprender?

- En mi opinión, ahí está la oportunidad. Cuanta menos gente esté dispuesta a aprender los fundamentos por el camino difícil, más valdrán quienes sí lo hagan. Serán la generación que cierre las brechas que los modelos no pueden cerrar solos.

> *La habilidad que la máquina más necesita de ti es la que menos puede aprender de su entrenamiento.*
>
> *A habilidade que a máquina mais precisa de você é a que ela menos consegue aprender com o treino dela.*

8. Un ejemplo real

Lo que de verdad costó "lo hizo la IA"

ai-memory, mi sistema de memoria open source para agentes de programación, está escrito mayormente por IA. Todo lo que lo mantiene correcto salió de mi revisión y mi dirección, semana tras semana.



8. Un ejemplo real

Lo que de verdad costó "lo hizo la IA"

4743
palabras de reglas para los agentes en AGENTS.md

16
invariantes que nunca se pueden romper

22
límites de seguridad aplicados, cada uno ligado a una guarda y a tests adversariales

3000+
tests unitarios

600+
tests de integración

- Unas 65 directivas explícitas, 28 notas de memoria curadas que existen por errores concretos, y más de 60 documentos con investigación, post-mortems y decisiones de arquitectura.
- Cada capa marca un lugar donde la IA sola no acertó y yo tuve que saber cómo arreglarlo.
- Nada de eso apareció el primer día. Son semanas viendo a la IA hacer el grueso del trabajo cada día y refinándolo, y el refinamiento depende de la experiencia.

8. Un ejemplo real

> *Nada de esto es automático, hecho por la IA. Todo esto es donde la IA sola no lo consigue y yo necesito saber cómo ajustar. Ningún amateur puede hacer esto.*
>
> *Nada disso é automático, feito pela IA. Tudo isso é onde a IA sozinha não consegue fazer e eu preciso saber com ajustar. Nenhum amador consegue fazer isso.*
>
> **Yo en X, 25 de septiembre de 2026**

9. Qué hacer

Si eres estudiante o junior

Usa la IA como un senior usa a un pasante: para las tareas rutinarias. El aprendizaje tiene que pasar por tus propias manos.



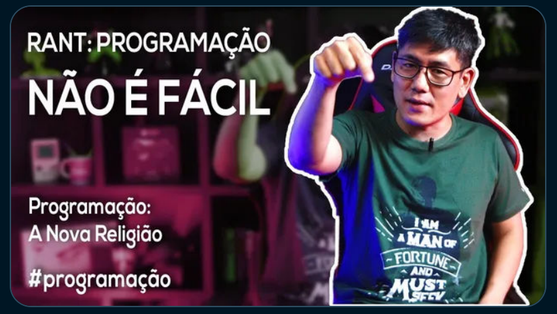
9. Qué hacer

Si eres estudiante o junior

- Escribe a mano todo lo que puedas. Rómpelo, depúralo, arréglalo. Atascarse es parte del proceso: los investigadores de Anthropic escriben que el esfuerzo cognitivo, "incluso atascarse dolorosamente", probablemente es importante para dominar algo.
- Revisa todo lo que produce, línea por línea. Si no puedes explicar un cambio, no estás listo para aceptarlo.
- Un senior es alguien que ya fue junior en cada tema que domina.
- Deja que la IA se encargue de las tareas rutinarias en paralelo, y pídele que te explique conceptos, no que escriba el código del que se supone que debes aprender.
- Cuenta las horas. De cero a un primer empleo como junior son al menos unas 1.400 horas de estudio y práctica; en mi opinión, dominar el oficio lleva decenas de miles. Los mejores violinistas de Ericsson tenían más de 10.000 a los 20 años, e investigaciones posteriores muestran que la práctica no lo explica todo, pero nada la reemplaza.

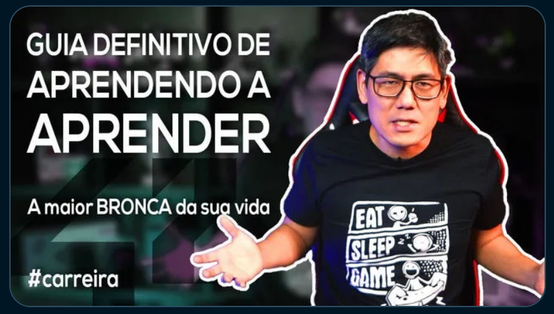
◈ Por dónde empezar

El archivo de Akitando



RANT: Programação NÃO É Fácil

feb 2020 · 32:22



Guia DEFINITIVO de Aprendendo a Aprender | A maior BRONCA d...

abr 2020 · 1:04:37



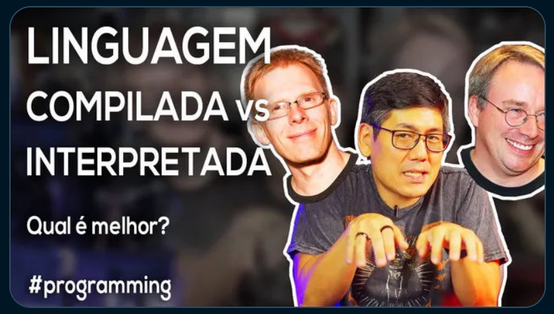
O Guia +Hardcore de Introdução à COMPUTAÇÃO

jun 2020 · 1:18:28



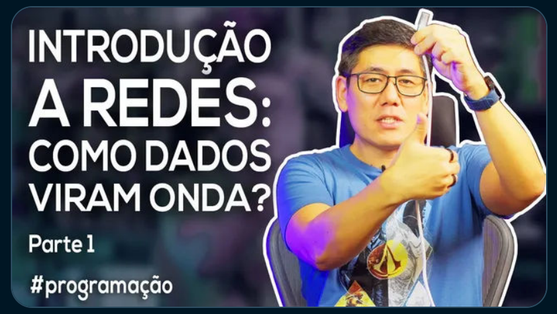
Hello World Como Você Nunca Viu! | Entendendo C

mar 2021 · 1:09:22



Linguagem Compilada vs Interpretada | Qual é melhor?

abr 2022 · 1:11:44



Introdução a Redes: Como Dados viram Ondas? | Parte 1

jul 2022 · 37:59

Las conclusiones

1

La IA multiplica lo que ya sabes. Con poco que multiplicar, obtienes muy poco.

2

Los atajos vendidos como carreras ya tuvieron un desplome. El nuevo es la misma promesa.

3

Aprender en piloto automático no se fija. Los estudios hasta ahora coinciden.

4

El trabajo que conserva su valor necesita personas capaces de juzgar el resultado, mientras se reduce la contratación de nivel inicial.

5

La IA no adivina tu objetivo, tus límites ni tus trade-offs. Conocerlos es el trabajo.

6

En mi benchmark, darle tests al modelo cambió los resultados. Elige el lenguaje que encaje con el problema.

7

La IA replica lo que es público. Los huecos que deja, en el trabajo privado y nuevo, son el trabajo, y los fundamentos difíciles son la forma de llenarlos.

8

Escribe a mano, deja las tareas rutinarias a la IA, revísalo todo y pon las horas.

Lo único que importa es la realidad objetiva.

A única coisa que importa é realidade objetiva.

Para leer más: mis posts y las fuentes

La IA multiplica lo que traes

- De cero a posproducción en una semana: usando IA en proyectos reales
- VS Code es la nueva tarjeta perforada
- Deja de poner excusas y publica más
- Google DORA 2025: la IA amplifica lo que hay
- Veracode: fallas de seguridad en código generado por IA

Este atajo ya lo vimos

- Desahogo: ¿la IA mató a los programadores?
- CFPB: las promesas de colocación de Lambda School
- Course Report: el mercado de bootcamps, de 2013 a 2018
- Inside Higher Ed: los cierres de bootcamps
- Higher Ed Dive: 2U deja los bootcamps
- TechCrunch: los despidos tech de 2024
- Akitando #132: estalló la burbuja de las startups (2022, en portugués)

Aprender en piloto automático no se fija

- Nuevo LLM Benchmark v4, parte 1: el proceso
- VS Code es la nueva tarjeta perforada
- Cómo DriveClub y shadPS4 casi nos vencen a la IA y a mí: cómo aprender
- Anthropic: cómo la asistencia de IA afecta las habilidades de programación
- Bastani et al., PNAS: la IA generativa puede perjudicar el aprendizaje
- METR: desarrolladores experimentados un 19 % más lentos
- MIT Media Lab: Your Brain on ChatGPT (preprint)

Para leer más: mis posts y las fuentes

El camino de entrada es más estrecho

- Desahogo: ¿la IA mató a los programadores?
- SignalFire: State of Tech Talent 2026
- Indeed: el mercado laboral se inclina hacia la experiencia
- Indeed: ofertas de software desde Claude Code (julio de 2026)
- Stanford Digital Economy Lab: la brecha de la IA para los trabajadores jóvenes (agosto de 2026)
- Fed de Nueva York: el mercado laboral de los recién graduados
- BLS: desarrolladores de software

La IA no adivina lo que necesitas

- Por qué los LLMs van a fracasar en tu empresa
- Por qué los LLMs no te dan el resultado que esperas
- Vibe code: hice una pequeña app 100 % con GLM 4.7 (TV Clipboard)

Los tests importan más que el lenguaje

- ¿Hay lenguajes mejores para programar con LLMs?
- El benchmark: código, datos y método

Si la IA puede programar todo, ¿para qué aprender?

- Los límites de los LLMs: buenos para reproducir lo que existe, caros para lo que no
- Ingeniería inversa con IA: ¿Adobe murió?
- VS Code es la nueva tarjeta perforada
- GitHub Octoverse 2025: 81,5 % privado
- arXiv (2026): los LLM y COBOL, un código que casi nunca es público
- ARC Prize: tareas nuevas, 0 % para LLM puros

Lo que de verdad costó "lo hizo la IA"

- ai-memory 2.0
- El software nunca está terminado: cuatro proyectos y la vida después del deploy

Para leer más: mis posts y las fuentes

Si eres estudiante o junior

- Cómo DriveClub y shadPS4 casi nos vencen a la IA y a mí: cómo aprender
- Desahogo: ¿Akita se rindió ante la IA?
- Cerrando mi maratón de IA: ¿éxito o fracaso?
- Anthropic: "getting painfully stuck"
- Ericsson et al. (1993): práctica deliberada
- Macnamara y Maitra (2019): la replicación
- Akitando #72: programar no es fácil (en portugués)

Cada punto, con todos los enlaces, en la web

ailair.akitaonrails.com/es/students/

Fabio Akita · akitaonrails.com · x.com/AkitaOnRails