Facsímil 01

Los cimientos

Capítulo 03

Principios fundamentales de la inteligencia artificial

686f6c61 · 2026

Capítulo 03: Principios fundamentales de la inteligencia artificial

Entrando en el tema

Has oído las palabras. Aprendizaje supervisado. *Transformers*. RLHF. *Scaling laws*. Suenan a conceptos avanzados que solo entiende quien tiene un doctorado. Pero no lo son. Son cinco ideas, sorprendentemente accesibles, que explican por qué la IA moderna funciona como funciona.

Este capítulo no te va a convertir en experto en ninguna de ellas. Para eso están el resto de capítulos del facsímil. Pero sí te va a dar el mapa. Cuando termines, sabrás nombrar las cinco patas de la mesa y, sobre todo, sabrás qué pata buscar cuando algo falle.

Aprendizaje supervisado: aprender con ejemplos

Imagina que quieres enseñarle a alguien a distinguir facturas falsas de facturas verdaderas. Le enseñas cien facturas. Cada una lleva una etiqueta: «verdadera» o «falsa». La persona observa, compara, detecta patrones (el logotipo borroso, el NIF que no cuadra, el formato de fecha extraño) y, tras revisar suficientes ejemplos, es capaz de clasificar facturas nuevas que no había visto antes.

Eso es el aprendizaje supervisado.

En términos técnicos: le das al modelo un conjunto de ejemplos etiquetados (cada entrada con su salida esperada) y el modelo ajusta sus parámetros para minimizar el error entre lo que predice y la etiqueta real.[1] Es la base del *fine-tuning*: coges un modelo pre-entrenado y lo ajustas con ejemplos específicos de tu dominio. El modelo ya sabe lenguaje; tú le enseñas tu jerga.

El aprendizaje supervisado es el paradigma más intuitivo porque se parece a como aprendemos muchas cosas: practicando con ejemplos cuya respuesta correcta conocemos.

Aprendizaje no supervisado: encontrar patrones sin pistas

Ahora imagina que tienes un montón de facturas sin etiquetar. No sabes cuáles son verdaderas ni falsas. Pero aun así, al observarlas, detectas que algunas se parecen mucho entre sí (mismo formato, mismos proveedores, mismos importes redondos) y otras son

completamente distintas. Sin que nadie te haya dicho qué es cada cosa, has encontrado estructura.

Eso es el aprendizaje no supervisado.

El modelo recibe datos sin etiquetar y busca patrones, agrupaciones o representaciones. No hay una «respuesta correcta» contra la que comparar. El modelo descubre la estructura por sí mismo.

Este paradigma es pariente cercano del responsable silencioso del éxito de los LLMs. Cuando un modelo como GPT se pre-entrena sobre miles de millones de documentos de internet, no hay un humano etiquetando cada frase. El modelo se entrena con una tarea **auto-supervisada** (*self-supervised*): predecir la siguiente palabra.[2] Conviene precisarlo, porque es un matiz que el campo distingue: no es exactamente aprendizaje no supervisado. En el no supervisado no hay respuesta correcta y solo se busca estructura. En el auto-supervisado sí hay etiqueta, pero sale de los propios datos: la siguiente palabra del texto es la respuesta. Esa diferencia es justo lo que permite aprovechar todo internet como ejemplos etiquetados sin que nadie etiquete nada a mano.

Post-training: enseñar preferencias, no solo hechos

Un modelo pre-entrenado sabe completar frases. Pero no sabe que es mejor decir «no tengo esa información» que inventársela. No sabe que ciertos temas requieren un tono más cuidadoso. No sabe cuándo callar.

El *post-training* resuelve esto. Tras el pre-entrenamiento masivo (aprendizaje no supervisado), se aplican técnicas que enseñan al modelo **preferencias**:

- **SFT** (*Supervised Fine-Tuning*): se entrena al modelo con ejemplos de conversaciones de alta calidad escritas por personas. El modelo aprende el formato y el tono deseados.
- **RLHF** (*Reinforcement Learning from Human Feedback*): personas comparan pares de respuestas del modelo y eligen cuál es mejor. Esa señal de preferencia entrena un modelo de recompensa que guía al modelo principal. Es aprendizaje por refuerzo, pero la recompensa la define un humano, no el entorno.[3]
- **DPO** (*Direct Preference Optimization*): una alternativa a RLHF que aprende directamente de las preferencias humanas sin necesidad de entrenar un modelo de recompensa separado.[4]

RLHF y DPO conectan con el tercer gran paradigma del aprendizaje automático, el que completa la tríada junto al supervisado y el no supervisado: el **aprendizaje por refuerzo**, donde un agente aprende por ensayo y error guiado por una señal de recompensa, no por ejemplos etiquetados. Lo distintivo en RLHF es que esa recompensa la define la preferencia humana, no el entorno. El facsímil 10 está dedicado por completo a este paradigma.

El *post-training* es la capa que convierte un modelo que «sabe cosas» en un modelo con el que puedes hablar. Lo exploraremos en detalle en el facsímil 4.

Atención: mirar todo a la vez

Piensa en cómo lees esta frase: «El gato que perseguía al ratón que robó el queso que estaba en la cocina es negro». Para saber que «negro» se refiere a «gato», tu cerebro ha tenido que conectar dos palabras separadas por una larga cláusula subordinada. Ha tenido que prestar atención a la relación entre «gato» y «negro» ignorando temporalmente el ratón, el queso y la cocina.

Los modelos anteriores a 2017 no podían hacer esto eficientemente. Procesaban el texto palabra por palabra, en orden, y para cuando llegaban a «negro» ya se habían olvidado de «gato».

El mecanismo de **atención**, presentado en el artículo «Attention Is All You Need»,[5] cambió esto radicalmente. Permite al modelo mirar **todas las palabras de la entrada simultáneamente** y decidir, para cada palabra que va a generar, cuáles de las palabras de entrada son relevantes y en qué medida.

Es el mecanismo que hizo posibles los LLMs modernos. Sin atención, los modelos no podrían manejar contextos largos ni capturar dependencias entre palabras separadas por decenas de miles de tokens. El facsímil 3 está dedicado íntegramente a abrir esta caja negra y entenderla por dentro.

Scaling laws: escala, coste y límites

En 2020, un grupo de investigadores de OpenAI se preguntó algo aparentemente simple: ¿qué pasa si entrenamos modelos cada vez más grandes con más datos y más computación? La respuesta, publicada en el artículo «Scaling Laws for Neural Language Models»,[6] fue sorprendentemente ordenada: el rendimiento mejora de forma **predecible** con la escala.

La relación simplificada es: **más datos + más parámetros + más computación = modelo más capaz**. Pero escrita así puede engañar. No significa que cualquier modelo grande sea mejor en cualquier tarea, ni que puedas ignorar la calidad de los datos, la arquitectura, el entrenamiento o la evaluación. Significa que, bajo ciertas condiciones experimentales, la pérdida baja siguiendo una relación bastante regular cuando aumentan parámetros, datos y cómputo.

Esto no es una analogía: hay una fórmula real detrás. Una forma de escribir una ley de escala, en la línea de los trabajos de Kaplan y de Chinchilla, es esta:[7]

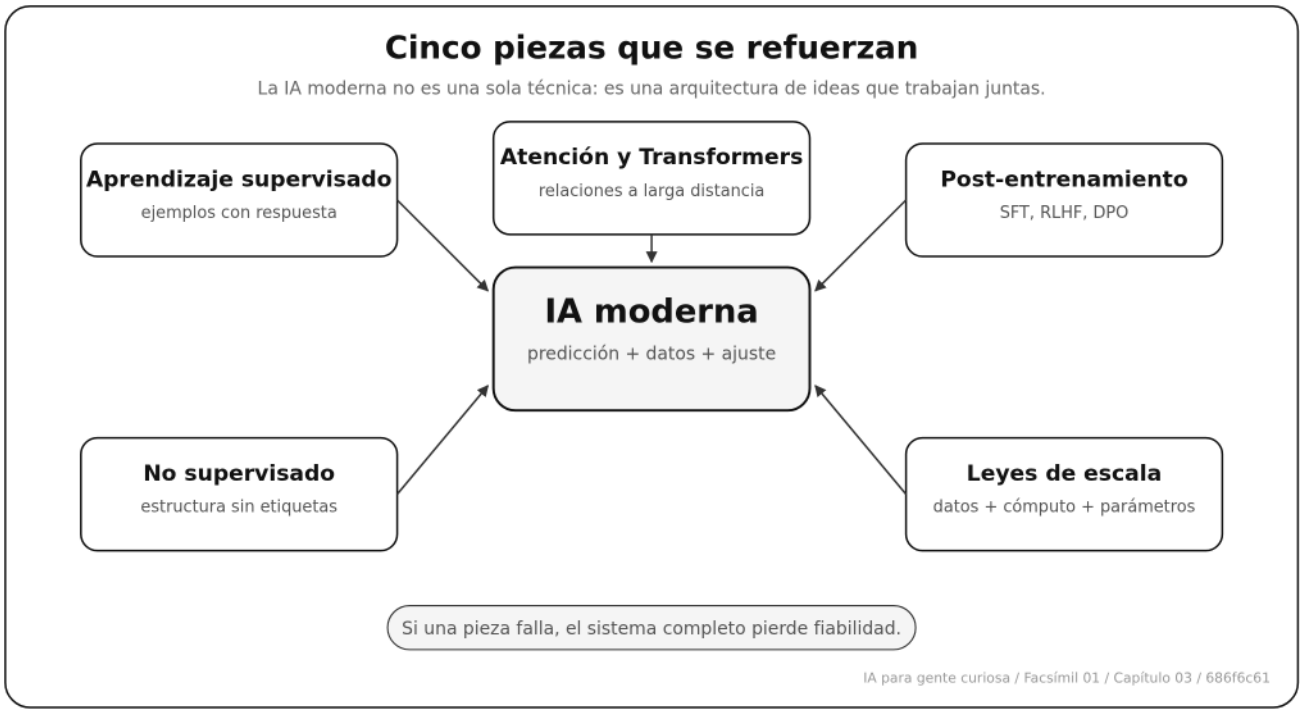
$$L(N, D) \approx L_\infty + \frac{A}{N^\alpha} + \frac{B}{D^\beta}$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
$L(N, D)$	Pérdida esperada del modelo entrenado con N parámetros y D tokens	1,72
L_∞	Límite inferior aproximado: pérdida que no desaparece aunque escales	1,55
N	Número de parámetros del modelo	7B, 70B, 405B
D	Cantidad de datos de entrenamiento, normalmente tokens	1T tokens
A, B	Constantes ajustadas empíricamente	dependen del experimento
α, β	Exponentes que dicen cuánto ayuda escalar parámetros o datos	valores pequeños, no milagrosos

Esta fórmula es una forma pedagógica de leer la idea, no una receta para presupuestar tu modelo ni una garantía de rendimiento. Las constantes y exponentes se estiman empíricamente en un experimento concreto; cambian con arquitectura, datos, régimen de entrenamiento y métrica. La intuición importante es que hay **rendimientos decrecientes**. Añadir diez veces más parámetros no suele darte diez veces más calidad. Además, Hoffmann y coautores mostraron con Chinchilla que muchos modelos grandes estaban infraentrenados: para un presupuesto de cómputo fijo, no basta con aumentar parámetros; también hay que aumentar datos de entrenamiento en proporción razonable.[8]

Esto explica por qué la industria invierte miles de millones en clústeres de GPUs, pero también explica por qué un equipo serio no compra «el modelo más grande» sin medir. Las *scaling laws* sirven para razonar sobre tendencias de entrenamiento; tu decisión de producto se toma con evals, latencia, coste por token, privacidad, facilidad de operación y calidad en tu tarea.

La lectura menos optimista es clara: mejorar el rendimiento requiere inversiones crecientes. Pasar de un modelo bueno a uno excelente cuesta mucho más que pasar de uno malo a uno aceptable. Entender esta dinámica es clave para tomar decisiones realistas sobre qué modelos usar y cuándo.



Cómo funciona por dentro

Cuando uno mira los cinco principios desde lejos, parece que cada uno exige una maquinaria distinta. No es así. Bajo el aprendizaje supervisado, el no supervisado y el post-entrenamiento late un único motor común: el descenso de gradiente (*gradient descent*), el mismo procedimiento que se desarrolla en los capítulos 6 y 7. La idea es siempre la misma. El modelo tiene millones (o miles de millones) de parámetros ajustables. Se mide cómo de mal lo hace con una función de pérdida (*loss*). Se calcula en qué dirección habría que mover cada parámetro para que esa pérdida baje un poco. Y se da un pequeño paso en esa dirección. Repetido millones de veces, ese gesto humilde es lo que convierte una red de números aleatorios en un modelo capaz.

Si la maquinaria es la misma, ¿qué distingue entonces a un paradigma de otro? La respuesta es más sencilla y más elegante de lo que suele contarse: lo único que cambia es de dónde sale la señal de error, es decir, contra qué se compara la predicción del modelo para saber si ha acertado o ha fallado. El motor no cambia. Cambia la fuente de la verdad que alimenta ese motor.

En el aprendizaje supervisado, la señal sale de una etiqueta puesta por una persona. El modelo ve una factura, predice «verdadera», y la pérdida mide la distancia entre esa predicción y la etiqueta humana («falsa»). Esa distancia es el error, y de ese error nace el gradiente que corrige los parámetros. En el aprendizaje no supervisado, y muy en particular en su variante auto-supervisada (*self-supervised*), no hay nadie etiquetando: la señal sale de la propia estructura de los datos. Cuando un modelo de lenguaje predice la siguiente palabra de un texto, la «etiqueta» es simplemente la palabra que de verdad venía a continuación. El

texto se etiqueta a sí mismo. Por eso se puede entrenar con internet entero sin pagar a nadie por anotar nada. En el post-entrenamiento (RLHF y DPO), la señal sale de las preferencias humanas: a una persona se le muestran dos respuestas del modelo y elige cuál prefiere. Esa preferencia (esta es mejor que aquella) se convierte en la fuente de error que ajusta los parámetros para que el modelo tienda a producir lo que la gente prefiere.

Conviene ser honestos aquí, porque de los cinco principios del capítulo solo tres son de verdad paradigmas de aprendizaje. La atención no lo es: no es una forma de extraer señal de error, sino un mecanismo arquitectónico, una pieza del diseño interno del modelo que decide cómo este mira la entrada y pondera la relevancia de unas palabras respecto a otras. La atención no entrena al modelo; es parte de lo que el motor de gradiente ajusta. Y las *scaling laws* tampoco son un mecanismo: son una ley empírica, una observación regular sobre cómo mejora el rendimiento cuando aumentan parámetros, datos y cómputo. Describen el comportamiento del sistema visto desde fuera, no una pieza que puedas tocar. Tanto la atención como las leyes de escala operan sobre el mismo motor de descenso de gradiente, pero en otra capa: una en la arquitectura, la otra en la observación del conjunto.

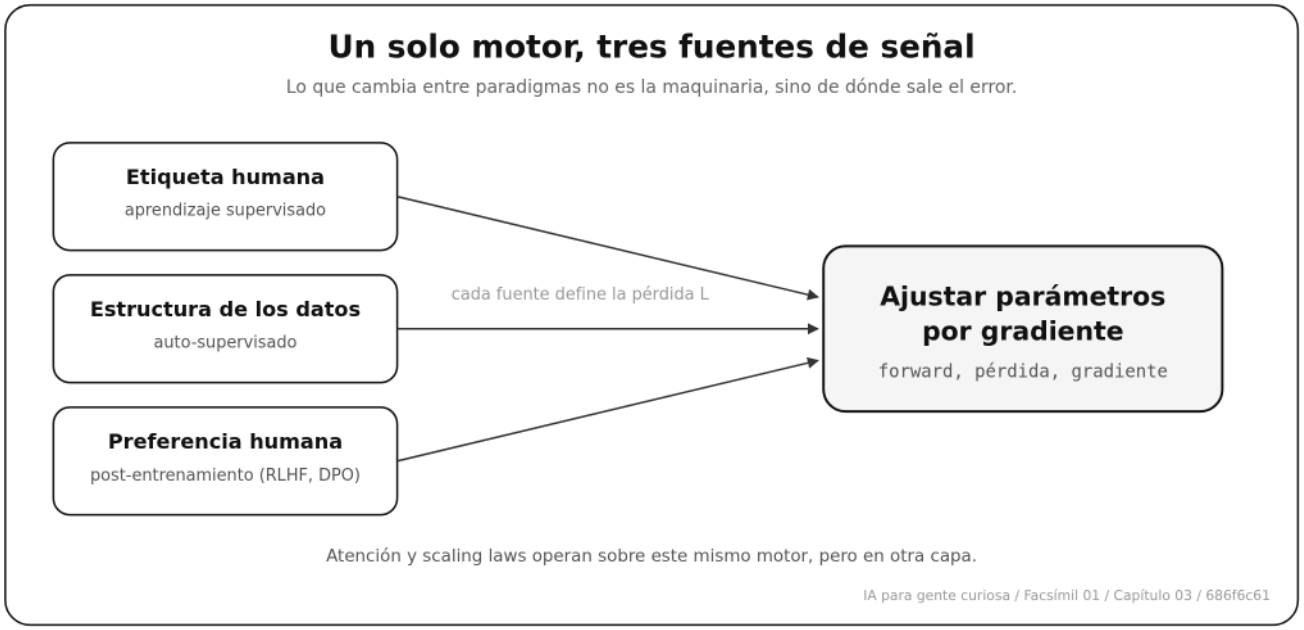
Vale la pena bajar al detalle del mecanismo común, porque entenderlo de una vez sirve para los tres paradigmas a la vez. El ciclo tiene cuatro pasos. Primero, el paso hacia delante (*forward pass*): los datos de entrada atraviesan el modelo y este produce una predicción. Segundo, el cálculo de la pérdida: se compara esa predicción con la señal de referencia, y aquí es donde entra la única diferencia entre paradigmas, porque la referencia es una etiqueta humana, o la siguiente palabra del propio texto, o una preferencia entre dos respuestas. Tercero, el gradiente: mediante retropropagación (*backpropagation*, el tema del capítulo 7) se calcula, para cada parámetro, cuánto contribuye al error y en qué dirección debería moverse para reducirlo. Cuarto, la actualización: cada parámetro se desplaza un poquito en la dirección que reduce la pérdida, regulado por un factor llamado tasa de aprendizaje (*learning rate*).

Una forma compacta de escribir ese cuarto paso, la actualización, es esta:

$$\theta_{t+1} = \theta_t - \eta \, \nabla_\theta L$$

SÍMBOLO	SIGNIFICADO	EJEMPLO
θ_t	Los parámetros del modelo en el paso actual	una red de millones de pesos
θ_{t+1}	Los parámetros ya corregidos para el paso siguiente	la red, un poquito mejor
η	Tasa de aprendizaje: tamaño del paso que se da	0,0001
$\nabla_\theta L$	Gradiente de la pérdida: dirección en la que el error sube	un vector con un valor por parámetro
L	Pérdida, calculada según la fuente de señal del paradigma	etiqueta, texto o preferencia

En palabras: coge los parámetros actuales, mira en qué dirección crece el error y muévete un paso pequeño en sentido contrario. Lo notable es que esta fórmula es idéntica en los tres paradigmas. Lo único que cambia entre ellos está escondido dentro de L: qué se usó como referencia para medir el error. Cambia la fuente de la señal, no el motor que la consume. Esa es, quizá, la idea más unificadora de todo el aprendizaje automático moderno, y la que conviene llevarse de este capítulo antes de pasar a las decisiones prácticas.



En el día a día

Cada uno de estos principios se traduce en decisiones de ingeniería.

Si tu problema tiene datos etiquetados, el aprendizaje supervisado, posiblemente mediante *fine-tuning*, es tu primer candidato. Clasificar tickets de soporte, detectar fraude, predecir abandono de clientes: todo esto es supervisado.

Si solo tienes datos sin etiquetar y necesitas encontrar estructura (agrupar clientes, detectar anomalías, reducir dimensiones), el aprendizaje no supervisado es la herramienta. Pero cuidado: que el modelo encuentre grupos no significa que esos grupos signifiquen algo útil para tu negocio. Lo veremos en el capítulo sobre *clustering*.

Si estás integrando un LLM en un producto, primero separa qué problema tienes. Si falta conocimiento que cambia (políticas internas, catálogo, normativa, documentación) normalmente necesitas recuperación de información, no tocar pesos. Si falta formato, tono, criterios de abstención o una conducta repetida y medible, entonces puede tener sentido *post-training*, *fine-tuning* o adaptadores. No basta con un buen *prompt*, pero tampoco se responde a todo con *fine-tuning*: necesitas saber qué pieza del sistema está fallando.

Si necesitas que el modelo maneje contexto largo, la atención es tu aliada y tu enemiga. Permite procesar documentos extensos, pero su coste computacional crece cuadráticamente con la longitud de la secuencia. Cada vez que duplicas el contexto, el coste se multiplica por cuatro. Diseñar estrategias de ventana de contexto (cuánto texto incluir, cuánto resumir, cuánto recuperar) es una decisión de arquitectura que depende de entender cómo funciona la atención.

Si estás decidiendo qué modelo usar, las *scaling laws* te dicen que más grande no siempre es mejor *para ti*. Un modelo de 7 000 millones de parámetros puede ser suficiente para tu caso de uso, y uno de 70 000 millones puede ser excesivamente caro y lento sin aportar una mejora proporcional. La decisión correcta depende de tus requisitos de calidad, latencia y coste.

Por qué debería importarte

Estos cinco principios no son trivia académica. Son el vocabulario con el que se toman todas las decisiones técnicas en IA.

Cuando alguien te diga «vamos a hacer *fine-tuning*», necesitas saber que está hablando de aprendizaje supervisado sobre un modelo pre-entrenado. Cuando escuches «este modelo tiene mejor atención», necesitas entender qué significa eso para tu caso de uso. Cuando un proveedor te diga «nuestro modelo es más grande, por tanto mejor», las *scaling laws* te dan el marco para evaluar si ese «mejor» justifica el coste.

Cada capítulo a partir de ahora asume que conoces este mapa. La neurona artificial es la pieza mínima del aprendizaje supervisado. La retropropagación es cómo se ajustan los parámetros. El Transformer es la atención llevada a la práctica. El *fine-tuning* es aprendizaje supervisado aplicado. Todo encaja.

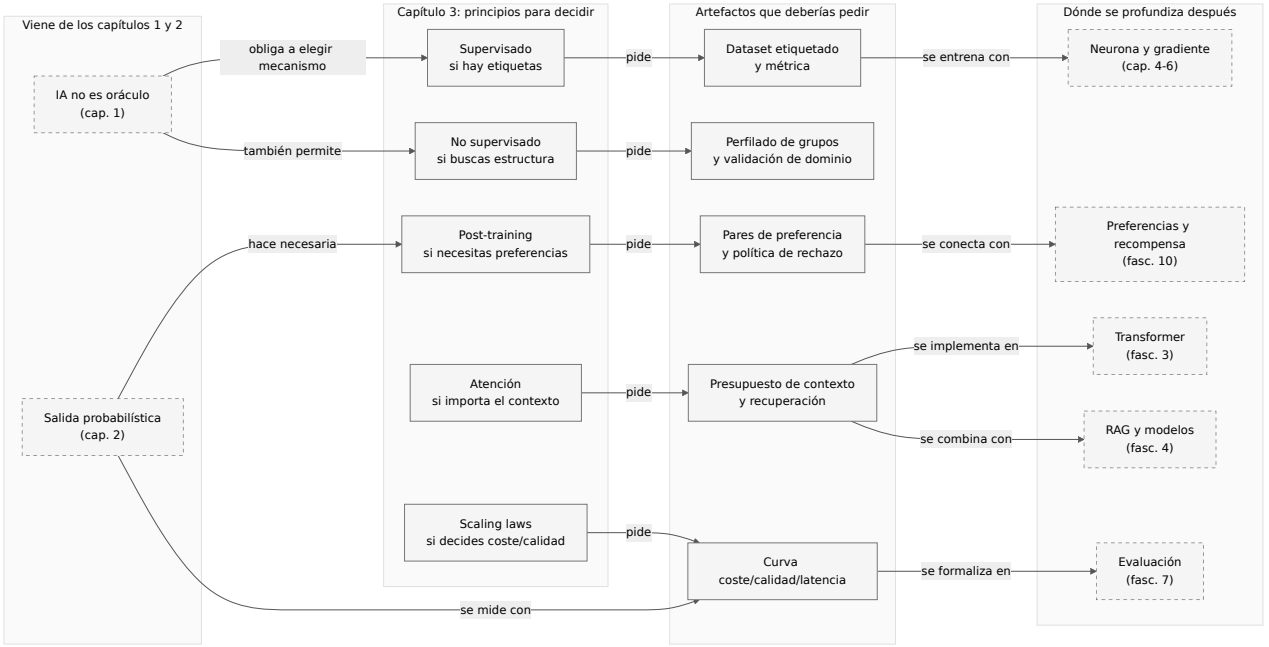
Dónde solía tropezar yo

ERROR	POR QUÉ ES UN ERROR	ANTÍDOTO
Creer que el aprendizaje no supervisado no necesita validación	Que un modelo encuentre patrones no significa que esos patrones sean útiles, reales o accionables. Un *cluster* puede ser un artefacto estadístico sin significado de negocio.	Valida siempre los resultados no supervisados contra conocimiento del dominio. Un grupo de clientes debe tener sentido para el equipo de negocio, no solo para el algoritmo.
Asumir que más parámetros siempre es mejor	Un modelo más grande es más caro, más lento y no necesariamente mejor para tu caso de uso concreto. Las *scaling laws* predicen rendimiento agregado, no rendimiento en tu tarea específica.	Evalúa el modelo en tus datos y tu tarea. No delegues la decisión en el número de parámetros.
Responder "fine-tuning" antes de diagnosticar	Un dominio especializado puede fallar por muchas razones: falta contexto, datos obsoletos, formato de salida inestable, vocabulario técnico, criterios de abstención o evaluación pobre. Ajustar pesos no arregla documentos desactualizados ni permisos mal diseñados.	Decide por causa: RAG para conocimiento vivo, reglas o herramientas para acciones verificables, *fine-tuning* o LoRA para comportamiento repetido y estable, y evaluación antes de cambiar nada.
Ignorar el coste cuadrático de la atención	Duplicar la longitud del contexto multiplica por cuatro el coste computacional. Diseñar sin tener esto en cuenta lleva a sistemas lentos y caros en producción.	Mide el coste real con tu volumen de datos y tu latencia objetivo. Considera estrategias de ventana deslizante, resumen previo o recuperación selectiva.

Cómo encaja todo

Este mapa se lee de izquierda a derecha. Hereda de los dos capítulos anteriores dos ideas: la IA no es una mente y sus salidas son probabilísticas. El capítulo 3 añade el primer vocabulario de diseño: qué tipo de aprendizaje o arquitectura estás invocando cuando dices «vamos a usar IA».

La decisión que enseña es práctica: no todos los problemas piden el mismo principio. Un caso con etiquetas pide supervisado; uno sin etiquetas pide exploración; uno con preferencias pide post-training; uno con contexto largo pide entender atención; uno limitado por coste pide mirar escala y evals. Ese mapa reaparece en casi todo lo que viene después.


Viene de los capítulos 1 y 2
IA no es oráculo
(cap. 1)
Salida probabilística
(cap. 2)
Capítulo 3: principios para decidir
obliga a elegir
mecanismo
también permite
hace necesaria
Supervisado
si hay etiquetas
No supervisado
si buscas estructura
Post-training
si necesitas preferencias
Atención
si importa el contexto
Scaling laws
si decides coste/calidad
se mide con
Artefactos que deberías pedir
pide
pide
pide
pide
pide
Dataset etiquetado
y métrica
Perfilado de grupos
y validación de dominio
Pares de preferencia
y política de rechazo
Presupuesto de contexto
y recuperación
Curva
coste/calidad/latencia
Dónde se profundiza después
se entrena con
se conecta con
se implementa en
se combina con
se formaliza en
Neurona y gradiente
(cap. 4-6)
Preferencias y
recompensa
(fasc. 10)
Transformer
(fasc. 3)
RAG y modelos
(fasc. 4)
Evaluación
(fasc. 7)

Vocabulario aprendido

TÉRMINO	DEFINICIÓN
Aprendizaje supervisado	Paradigma donde el modelo aprende a partir de ejemplos etiquetados con la respuesta correcta, ajustando sus parámetros para minimizar el error de predicción.
Aprendizaje no supervisado	Paradigma donde el modelo encuentra patrones en datos sin etiquetar, sin una respuesta correcta predefinida contra la que comparar.
Aprendizaje auto-supervisado	Variante donde la etiqueta sale de los propios datos (por ejemplo, predecir la siguiente palabra), lo que permite entrenar con texto sin etiquetar a mano.
Pre-entrenamiento	Fase inicial masiva de entrenamiento no supervisado donde el modelo aprende patrones generales del lenguaje a partir de cantidades inmensas de texto.
Post-training	Conjunto de técnicas aplicadas tras el pre-entrenamiento para enseñar al modelo preferencias de tono, formato y conducta, no solo hechos.
SFT	Ajuste supervisado del modelo con ejemplos de conversaciones de alta calidad escritas por personas para fijar formato y tono.
Aprendizaje por refuerzo	Paradigma donde un agente aprende por ensayo y error guiado por una señal de recompensa, no por ejemplos etiquetados.
RLHF	Técnica de post-entrenamiento donde la señal de aprendizaje proviene de personas que comparan y puntúan respuestas del modelo.
DPO	Técnica de optimización por preferencias que aprende de pares de respuestas elegidas y rechazadas sin entrenar un modelo de recompensa separado.
Fine-tuning	Proceso de tomar un modelo pre-entrenado y ajustarlo con ejemplos etiquetados de un dominio específico. Es aprendizaje supervisado sobre una base pre-entrenada.
Atención	Mecanismo que permite al modelo evaluar la relevancia de cada palabra de entrada para cada palabra de salida, procesando todas las relaciones simultáneamente.
Scaling law	Relación predecible entre los recursos invertidos en entrenamiento y el rendimiento del modelo.

Antes de pasar página

- [ ] ¿Puedo explicar la diferencia entre aprendizaje supervisado y no supervisado con un ejemplo concreto? (Si no, vuelve a las dos primeras secciones.)

- [ ] ¿Entiendo qué aporta el *post-training* que no aporta el pre-entrenamiento? (Si no, vuelve a «Post-training: enseñar preferencias».)
- [ ] ¿Sé qué hace el mecanismo de atención y por qué fue revolucionario? (Si no, vuelve a «Atención: mirar todo a la vez».)
- [ ] ¿Puedo explicar por qué las *scaling laws* son importantes para decidir qué modelo usar? (Si no, vuelve a «Scaling laws» y a «En el día a día».)
- [ ] ¿Entiendo por qué más parámetros no siempre es la respuesta correcta? (Si no, vuelve a «Dónde solía tropezar yo», error «Asumir que más parámetros siempre es mejor».)
- [ ] ¿Sé justificar qué principio domina un caso propio y cómo lo comprobaría? (Si no, vuelve a «Cómo funciona por dentro».)

En resumen

IDEA FUERZA	DETALLE
La IA moderna se apoya en cinco principios.	Aprendizaje supervisado, no supervisado, post-training, atención y *scaling laws*. Entenderlos es tener el mapa antes de adentrarse en el territorio.
El aprendizaje supervisado entrena con ejemplos etiquetados; el no supervisado encuentra patrones sin etiquetas.	El primero es la base del *fine-tuning*; el segundo, del pre-entrenamiento de los LLMs.
El mecanismo de atención permite procesar todas las palabras a la vez.	Es la innovación arquitectónica que hizo posibles los modelos de lenguaje modernos. Sin ella, no hay Transformer.
Las *scaling laws* demuestran que el rendimiento mejora de forma predecible con la escala.	Pero también advierten de que la mejora tiene un coste creciente. Más grande no siempre es la respuesta correcta para tu caso.

Para saber más

Goodfellow, I., Bengio, Y. y Courville, A. (2016). *Deep learning*. MIT Press. https://www.deeplearningbook.org

Hoffmann, J. y otros (2022). Training compute-optimal large language models. arXiv:2203.15556. https://doi.org/10.48550/arXiv.2203.15556

Kaplan, J. y otros (2020). Scaling laws for neural language models. arXiv:2001.08361. https://doi.org/10.48550/arXiv.2001.08361

Ouyang, L. y otros (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems 35*, 27730-27744. https://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D. y Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems 36*. https://arxiv.org/abs/2305.18290

Rumelhart, D. E., Hinton, G. E. y Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536. https://doi.org/10.1038/323533a0

Russell, S. y Norvig, P. (2021). *Artificial intelligence: a modern approach* (4.ª ed.). Pearson.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. y Polosukhin, I. (2017). Attention is all you need. En *Advances in Neural Information Processing Systems 30* (pp. 5998-6008). https://papers.nips.cc/paper/7181-attention-is-all-you-need

---

Notas

1. Russell, S. y Norvig, P. (2021). *Artificial intelligence: a modern approach* (4.ª ed.). Pearson. Los capítulos 18 a 21 cubren en profundidad los paradigmas de aprendizaje supervisado, no supervisado y por refuerzo. ↩

2. Goodfellow, I., Bengio, Y. y Courville, A. (2016). *Deep learning*. MIT Press. https://www.deeplearningbook.org. El capítulo 14 aborda los autoencoders y las representaciones aprendidas sin etiquetas humanas, fundamentos del pre-entrenamiento moderno. ↩

3. Ouyang, L. y otros (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems 35*, 27730-27744. https://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html ↩

4. Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D. y Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems 36*. https://arxiv.org/abs/2305.18290 ↩

5. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. y Polosukhin, I. (2017). Attention is all you need. En *Advances in Neural Information Processing Systems 30* (pp. 5998-6008). https://papers.nips.cc/paper/7181-attention-is-all-you-need ↩

6. Kaplan, J. y otros (2020). Scaling laws for neural language models. arXiv:2001.08361. https://doi.org/10.48550/arXiv.2001.08361 ↩

7. La idea de las leyes de escala procede de Kaplan, J. y otros (2020), arXiv:2001.08361. La forma con un término para los parámetros y otro para los datos, más una pérdida irreducible, sigue a Hoffmann, J. y otros (2022), arXiv:2203.15556, el trabajo conocido como Chinchilla. Aquí se escribe L_∞ donde el artículo original usa E. ↩

8. Hoffmann, J. y otros (2022). Training compute-optimal large language models. *Advances in Neural Information Processing Systems 35*. https://doi.org/10.48550/arXiv.2203.15556 ↩