Facsímil 01

# Los cimientos

Capítulo 01

## Qué es y qué no es la inteligencia artificial

686f6c61 · 2026

# Capítulo 01: Qué es y qué no es la inteligencia artificial

## Entrando en el tema

Abres el navegador, escribes «explícame cómo funciona un motor de combustión» y pulsas Intro. En menos de tres segundos tienes delante un texto articulado, con ejemplos, párrafos bien estructurados y un tono didáctico impecable. Tu cerebro, entrenado durante décadas para asociar lenguaje coherente con inteligencia, llega a una conclusión instantánea: aquí dentro hay alguien que sabe de motores.

Es una reacción comprensible, humana y equivocada.

Este capítulo nace exactamente de esa reacción. Vamos a desmontarla pieza a pieza. Porque entender qué es, y sobre todo qué no es, la inteligencia artificial es la primera decisión de ingeniería que tomarás al trabajar con ella. Y posiblemente la más importante.

## Qué no es la inteligencia artificial

Empecemos por lo que no es. La lista de malentendidos es larga, pero cinco negaciones bastan para despejar el terreno.

**No es magia ni ciencia ficción.** No hay una entidad misteriosa dentro de la máquina. Es *software*, ejecutándose en servidores, con electricidad y silicio.[1] Impresionante, sí. Pero explicable, medible y auditable.

**No es una mente consciente.** No tiene deseos, emociones ni intenciones. No «quiere» nada. Procesa tokens y calcula probabilidades. Punto. Si alguna vez has sentido que el modelo «te entiende», has experimentado el efecto Eliza: nuestro cerebro antropomorfiza cualquier cosa que produzca lenguaje coherente.[2]

**No es un buscador glorificado.** Un buscador recupera páginas o documentos que ya existen. Un LLM (*Large Language Model*, modelo grande de lenguaje) genera texto a partir de patrones aprendidos durante el entrenamiento y no consulta fuentes por defecto. Son mecanismos radicalmente distintos.

**No «piensa» como un humano.** Puede producir pasos que parecen razonamiento, pero su mecanismo base es predecir el siguiente token a partir de patrones estadísticos. No hay deliberación, no hay duda, no hay «déjame pensarlo». Hay una multiplicación de matrices a escala industrial.

**No es infalible ni objetiva.** Puede alucinar (generar información falsa con total confianza), heredar sesgos de los datos de entrenamiento y carecer de introspección fiable sobre sus propios errores. No te dirá «no sé» a menos que se lo hayas enseñado explícitamente.

---

## Qué sí es la inteligencia artificial

Ahora la definición positiva. Si la IA no es conciencia ni un oráculo, ¿qué es?

**Modelos matemáticos masivos.** Redes neuronales con miles de millones o billones de parámetros (números) ajustados para reconocer patrones en texto, imágenes, código y audio.

**Reconocimiento de patrones a escala.** Lo que una persona tarda horas en analizar, el modelo lo procesa en segundos. No «entiende» en el sentido humano de saber qué está diciendo, contrastarlo con el mundo y hacerse responsable de ello: aprende regularidades estadísticas y las usa para producir una salida plausible. Esa diferencia parece filosófica, pero en ingeniería es muy concreta: una salida plausible necesita verificación.

**Predicción estadística del siguiente token.** Dada una secuencia de tokens, ¿cuál es el más probable a continuación? Esta operación, repetida miles de veces, produce párrafos coherentes, código funcional y respuestas que parecen razonadas. Es el mecanismo central de todo LLM moderno.

**Un amplificador de capacidades.** Si sabes programar, te hace más rápido. Si no sabes, te da una falsa sensación de que funciona… hasta que deja de hacerlo. La IA no sustituye el criterio humano: lo amplifica. Si le das contexto claro y restricciones precisas, el resultado es impresionante. Si le das ambigüedad, el resultado es impredecible.

Para situarnos en el mapa, estos son los hitos que nos han traído hasta aquí. La parte histórica es estable; la parte de herramientas actuales no lo es. **Fecha de corte: 10 de junio de 2026.** Ese día, las fuentes consultadas para la fila de sistemas con herramientas y agentes fueron la documentación pública de OpenAI Agents SDK, Claude Agent SDK, Google ADK y Model Context Protocol.[3]

| FECHA | HITO | POR QUÉ IMPORTA |
|---|---|---|
| **1950** | Test de Turing | Turing propone un criterio para evaluar si una máquina exhibe comportamiento inteligente. No es un test técnico, pero inaugura la pregunta filosófica que aún nos hacemos.[4] |
| **1956** | Conferencia de Dartmouth | McCarthy, Minsky, Shannon y Rochester organizan un taller de verano en Dartmouth College. Acuñan el término «inteligencia artificial».[5] Es el nacimiento del campo como disciplina. |
| **1986** | Retropropagación | Rumelhart, Hinton y Williams publican el algoritmo que permite entrenar redes de más de dos capas. Sin esto no existiría el *deep learning*.[6] Lo veremos en detalle en el capítulo 6. |
| **2012** | AlexNet gana ImageNet | Una red neuronal convolucional arrasa en el concurso de reconocimiento de imágenes. Comienza la era del *deep learning* moderno.[7] |
| **2017** | Transformer | El artículo «Attention Is All You Need» presenta una arquitectura que procesa todas las palabras simultáneamente en vez de secuencialmente. Es la base de todo lo que vino después.[8] Lo desarrollaremos con todo detalle en el facsímil 3. |
| **2020** | GPT-3 | 175 000 millones de parámetros. Demostró que escalar funciona: más datos y más parámetros producen más capacidad.[9] |
| **2022** | ChatGPT | RLHF (aprendizaje por refuerzo con *feedback* humano) más una interfaz de chat. La IA se hace accesible al público general. |
| **2023** | GPT-4 | Multimodal y con razonamiento avanzado. Salto cualitativo en capacidades.[10] |
| **2024** | Claude 3 / Gemini | Competencia real. Contextos de 200 000 o más tokens. La IA se convierte en herramienta de trabajo diaria. |
| **2025-2026** | Sistemas con herramientas y agentes de software | El chat deja de ser la única interfaz. Aparecen SDKs y protocolos que combinan modelo, herramientas, estado, trazas, aprobaciones y ejecución sobre entornos reales. Lo exploraremos a fondo en el facsímil 5. |

## Cómo funciona por dentro

Este capítulo es conceptual: no vamos a entrenar modelos ni a derivar nada. Aparecerán solo dos expresiones matemáticas, las justas para ver el mecanismo con precisión. Antes de cerrarlo necesitas entender ese mecanismo, porque está debajo de todo lo demás. Cada capítulo futuro (la neurona artificial, la retropropagación, el Transformer, los agentes) se construye sobre esta pieza.

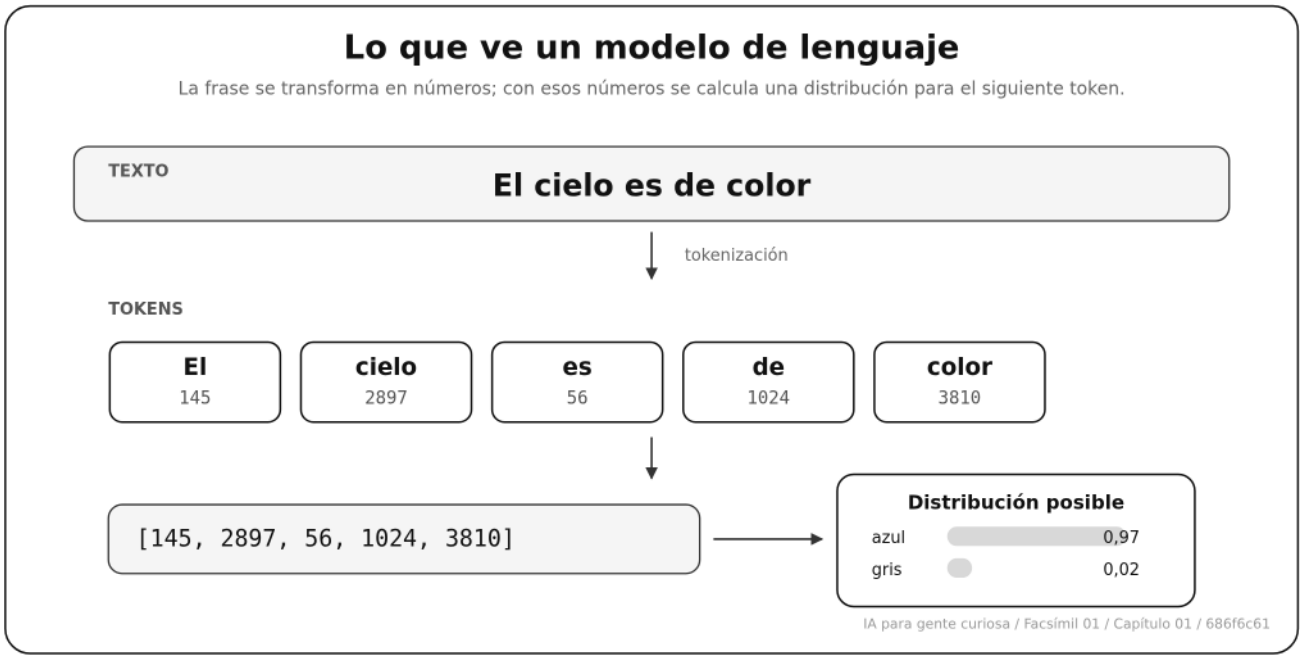
Se llama **predicción del siguiente token** y funciona así.

Imagina que escribes:

> El cielo es de color

El modelo recibe esta secuencia y ejecuta tres pasos:

**Paso 1: tokenización.** Convierte cada palabra (o fragmento de palabra) en un número que la identifica. «El» podría ser el token 145, «cielo» el 2897, «es» el 56, «de» el 1024 y «color» el 3810. El modelo no ve palabras: ve secuencias de números.




El modelo no ve palabras: ve secuencias de números. En el capítulo 9 exploraremos la tokenización en profundidad y entenderemos por qué «cielo» puede ser un token y «extraordinariamente» pueden ser tres.

**EJEMPLO, NO NOTACIÓN OFICIAL.**

$$\text{tok}(\text{El cielo es de color}) = [145,\ 2897,\ 56,\ 1024,\ 3810]$$

> No es una fórmula del campo: la tokenización es un algoritmo, y aquí ponemos un ejemplo concreto para verla como una función que recibe un texto y devuelve la lista de identificadores que ese texto tiene en el vocabulario del modelo. El algoritmo real (por ejemplo, *byte-pair encoding*) lo veremos en el capítulo 9.

**Paso 2: transformación por pesos aprendidos.** Esos números atraviesan las capas de la red neuronal. En cada capa, los miles de millones de parámetros, los pesos aprendidos durante el entrenamiento, transforman la representación. Conviene no llamarlo «memoria» sin matiz: el modelo no abre una ficha interna sobre el cielo ni recupera una frase guardada. Aplica pesos que codifican regularidades observadas durante el entrenamiento y produce una representación nueva de la secuencia.

**Paso 3: muestreo.** El modelo produce una distribución de probabilidad sobre todos los tokens posibles que podrían seguir. No elige «la correcta». Asigna probabilidades (en este ejemplo cada candidato es un token de una sola pieza):

- `azul` : 97 %
- `gris` : 2 %
- `rojo` : 0,5 %
- `verde` : 0,3 %
- `naranja` : 0,1 %
- ...y una probabilidad minúscula para todos los demás tokens del vocabulario, que son decenas de miles.

¿De dónde salen esos porcentajes? El modelo no produce probabilidades directamente, sino una puntuación sin normalizar para cada token, su *logit*. La función *softmax* convierte ese vector de puntuaciones en una distribución de probabilidad que suma 1:

$$P(t_{n+1} = v \mid t_1, \ldots, t_n) = \frac{e^{z_v}}{\sum_{u \in V} e^{z_u}}$$

| SÍMBOLO | SIGNIFICADO | EJEMPLO |
|---|---|---|
| $P(t_{n+1} = v \mid t_1, \ldots, t_n)$ | Probabilidad de que el siguiente token sea $v$, dado el contexto anterior | 0,97 para `azul` |
| $v$ | Un token candidato del vocabulario | `azul` , `gris` , `rojo` … |
| $z_v$ | Logit de $v$: la puntuación sin normalizar que el modelo asigna al token | más alta cuanto más probable es el token |
| $e^{z_v}$ | Exponencial del logit. Convierte cualquier puntuación en un número positivo | nunca negativo |
| $V$ | El vocabulario completo, todos los tokens posibles | decenas de miles |
| $\sum_{u \in V} e^{z_u}$ | Suma de las exponenciales de todos los tokens. Es lo que normaliza el resultado | el total que reparte el 100 % |

En palabras: el modelo exponencia la puntuación de cada token, así ninguna probabilidad es negativa, y divide por la suma de todas, así el conjunto suma 1. El token con el logit más alto se lleva la mayor probabilidad.[11]

Luego **muestrea** de esa distribución. Normalmente elegirá `azul`, pero no siempre. Esa pequeña probabilidad de elegir `gris` o `rojo` es lo que hace que el texto generado no sea idéntico en cada ejecución. Es lo que le da variabilidad, creatividad... y también impredecibilidad.

El proceso se repite. El token elegido, digamos `azul`, se añade a la secuencia de entrada:

```
El cielo es de color azul
```

Y el modelo predice el siguiente:

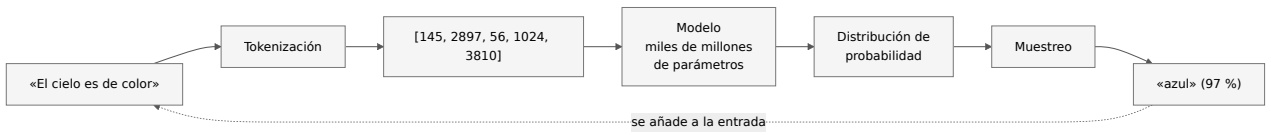
```
El cielo es de color azul ,
```

Y el siguiente:

El cielo es de color azul, **aunque**

Y así, token a token, durante cientos o miles de iteraciones, hasta que el modelo genera un token especial que significa «terminé». Lo que empezó como una pregunta de cinco palabras se convierte en una respuesta de tres párrafos.

No hay comprensión humana, intención ni conciencia. Hay una operación estadística, predecir el siguiente elemento probable de una secuencia, ejecutada a una escala que produce resultados que *parecen* inteligentes. La precisión importa: no estamos diciendo que el sistema sea inútil; estamos diciendo que su utilidad no lo convierte en sujeto, oráculo ni fuente fiable por sí misma.




## En el día a día

Entender qué es y qué no es la IA no es un ejercicio filosófico. Determina decisiones de ingeniería concretas. Veamos tres situaciones reales.

**Situación 1: ¿LLM o SQL?** Tienes una base de datos con millones de registros de ventas y necesitas saber el importe total del último trimestre. Si crees que el LLM «sabe cosas», le pasarás el *prompt* «¿cuánto vendimos el último trimestre?» y obtendrás un número. El problema es que ese número puede ser correcto, aproximado o completamente inventado, y no tendrás forma de saberlo sin verificarlo manualmente. Si entiendes que el LLM predice tokens, usarás SQL para la consulta exacta y el LLM para resumir el resultado o redactar el informe. Cada herramienta para lo que sirve.

**Situación 2: integrar IA generativa en un producto.** Tu equipo quiere añadir un asistente que responda preguntas de los usuarios sobre la documentación. Si crees que el modelo es infalible, desplegarás sin *guardrails* ni evaluaciones y el primer usuario que reciba una respuesta incorrecta, con la confianza y elocuencia características de un LLM, actuará sobre información falsa. Si entiendes que el modelo alucina, diseñarás el sistema con verificación humana para decisiones de impacto, recuperación de documentos fuente para respuestas auditables y evaluaciones automáticas que midan la tasa de alucinación antes de cada despliegue.

**Situación 3: depurar código generado.** Le pides al modelo que escriba una función de autenticación y te devuelve treinta líneas impecables. Si crees que «piensa», confiarás y desplegarás. Si entiendes el mecanismo, revisarás cada línea como harías con el código de un compañero junior muy rápido pero sin criterio: ¿valida todos los casos límite?, ¿usa una biblioteca actualizada?, ¿hay algún error sutil de lógica que solo se manifiesta con ciertos valores de entrada? La elocuencia del código no garantiza su corrección.

---

## Por qué debería importarte

La decisión técnica más importante al trabajar con IA no es qué modelo usar ni qué proveedor elegir. Es **entender qué estás usando realmente**.

De esa comprensión, o de su ausencia, se derivan todas las decisiones posteriores: cuándo desplegar un agente autónomo y cuándo un simple *prompt*, cómo diseñar las evaluaciones, qué nivel de supervisión humana exigir, cuánto presupuesto asignar a la revisión de salidas, qué arquitectura de *software* elegir para integrar el modelo.

Si entiendes que la IA no piensa, no entiende y no razona como un humano, puedes usarla mejor: le darás instrucciones más precisas, diseñarás sistemas que no dependan de su infalibilidad y confiarás menos ciegamente en sus respuestas.

El resto de este facsímil asume que has interiorizado esta idea. Cada capítulo vuelve a ella desde un ángulo distinto.

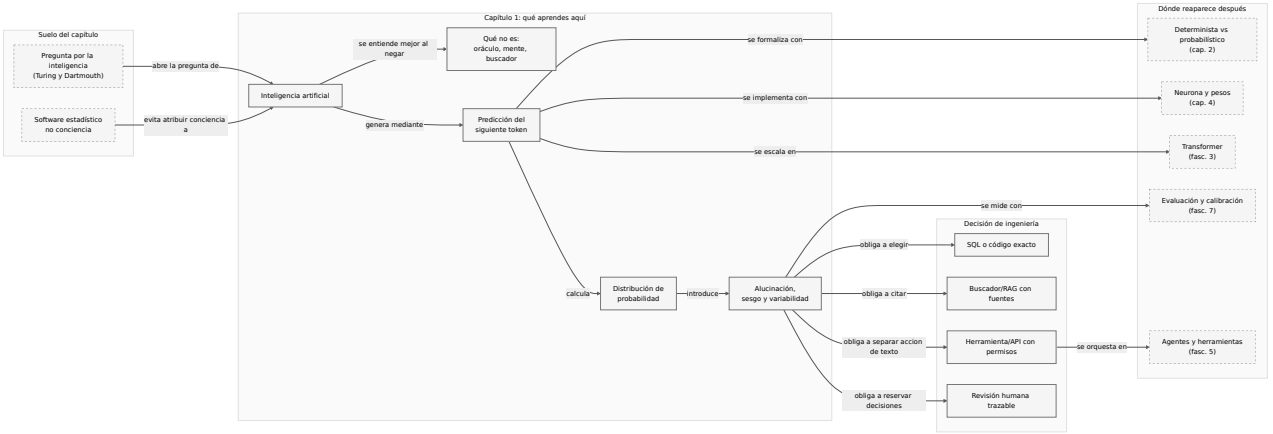
## Dónde solía tropezar yo

| ERROR | POR QUÉ ES UN ERROR | ANTÍDOTO |
|---|---|---|
| **Antropomorfizar** («la IA cree que...», «el modelo piensa que...») | Atribuir estados mentales a un sistema estadístico lleva a confusiones graves sobre sus capacidades y limitaciones. El modelo no cree nada: asigna probabilidades. | Sustituir mentalmente «la IA cree» por «el modelo asigna alta probabilidad a». Obligarse a usar el vocabulario técnico hasta que se convierta en hábito. |
| **Tratarla como un oráculo infalible** | Desplegar sin verificación humana en contextos donde el error tiene consecuencias económicas, legales o de seguridad. El modelo siempre responde con confianza, incluso cuando la respuesta es falsa. | Diseñar siempre con supervisión humana para decisiones de impacto. Si la decisión mueve dinero, afecta a personas o tiene consecuencias legales, un humano debe revisarla. |
| **Confundir elocuencia con corrección** | Un modelo puede generar texto perfectamente articulado, con estructura impecable y tono profesional, y ser completamente falso. La fluidez no es garantía de veracidad. | Verificar hechos contra fuentes externas. Las evaluaciones miden corrección, no estilo. Un texto bien escrito y equivocado es más peligroso que uno mal escrito y correcto. |
| **Subestimar el no determinismo** | Esperar la misma salida para el mismo *prompt*. El modelo muestrea de una distribución de probabilidad; dos ejecuciones pueden producir respuestas distintas. Esto rompe los tests unitarios tradicionales. | Diseñar tests que validen estructura y propiedades (¿la respuesta contiene un número?, ¿menciona los tres conceptos clave?), no texto exacto. |

## Cómo encaja todo

Este mapa se lee de izquierda a derecha. Primero aparece el suelo que hereda el capítulo: la pregunta histórica por la inteligencia y la realidad material de que hablamos de software estadístico. En el centro está la idea que debes llevarte: un modelo de lenguaje predice tokens, no consulta la verdad. A la derecha aparece la decisión de ingeniería que nace de esa idea: elegir entre consulta exacta, recuperación documental, generación, herramientas y supervisión.

Si el mapa solo dijera «IA conecta con Transformer», sería demasiado pobre. Lo importante es ver que este capítulo te prepara para distinguir mecanismos. Esa distinción reaparece en sistemas deterministas, neuronas, entrenamiento, RAG, agentes, evaluación y operación.


Suelo del capítulo
Pregunta por la inteligencia (Turing y Dartmouth)
Software estadístico no conciencia
abre la pregunta de
evita atribuir conciencia a
Inteligencia artificial
se entiende mejor al negar
Qué no es: oráculo, mente, buscador
genera mediante
Predicción del siguiente token
Capítulo 1: qué aprendes aquí
se formaliza con
se implementa con
se escala en
calcula
Distribución de probabilidad
introduce
Alucinación, sesgo y variabilidad
se mide con
Decisión de ingeniería
obliga a elegir
SQL o código exacto
obliga a citar
Buscador/RAG con fuentes
obliga a separar accion de texto
Herramienta/API con permisos
obliga a reservar decisiones
Revisión humana trazable
se orquesta en
Dónde reaparece después
Determinista vs probabilístico (cap. 2)
Neurona y pesos (cap. 4)
Transformer (fasc. 3)
Evaluación y calibración (fasc. 7)
Agentes y herramientas (fasc. 5)

# Vocabulario aprendido

| TÉRMINO | DEFINICIÓN |
|---|---|
| **Inteligencia artificial** | Rama de la informática que construye sistemas capaces de realizar tareas que normalmente requieren inteligencia humana, mediante modelos matemáticos que aprenden patrones a partir de datos. |
| **LLM** (*Large Language Model*) | Modelo de lenguaje de gran escala. Red neuronal con miles de millones de parámetros entrenada sobre cantidades masivas de texto para predecir y generar lenguaje. |
| **Token** | Unidad mínima de texto que el modelo procesa. Puede ser una palabra, parte de una palabra o un carácter. El modelo opera con tokens, no con palabras completas como las entendemos las personas. |
| **Predicción del siguiente token** | Mecanismo central de todo LLM: dada una secuencia de tokens, el modelo calcula qué token es más probable a continuación y, repitiendo la operación, genera texto coherente. |
| **Softmax** | Función que convierte las puntuaciones sin normalizar (logits) de cada token en una distribución de probabilidad que suma 1, de modo que ninguna probabilidad sea negativa. |
| **Parámetro** | Cada uno de los números que el modelo ajusta durante el entrenamiento. Un modelo de 175 000 millones de parámetros tiene 175 000 millones de números. Codifican patrones aprendidos, pero no son una base de datos consultable ni una memoria de hechos. |
| **Alucinación** | Fenómeno por el cual un LLM genera información sintácticamente correcta pero factualmente falsa, expresada con total seguridad. No es un *bug*: es una propiedad emergente de la predicción estadística. |
| **RLHF** (*Reinforcement Learning from Human Feedback*) | Técnica de post-entrenamiento donde personas evalúan respuestas del modelo y esa señal se usa para alinear su comportamiento con preferencias humanas. |
| **Transformer** | Arquitectura de red neuronal presentada en 2017 que sustituye el procesamiento secuencial por un mecanismo de atención paralela. Es la base de todos los LLMs modernos. |
| **RAG** (*Retrieval-Augmented Generation*) | Arquitectura que recupera documentos o datos externos antes de generar una respuesta, para que el modelo no dependa solo de sus patrones internos. |
| **Sistema híbrido** | Solución que combina varias piezas: consulta exacta, recuperación documental, generación, herramientas, permisos y revisión humana según el caso. |

## Antes de pasar página

- [ ] ¿Puedo explicar con mis propias palabras por qué un LLM no «piensa» ni «sabe» cosas como una persona? (Si no, vuelve a «Qué no es la inteligencia artificial».)
- [ ] ¿Entiendo el mecanismo de predicción del siguiente token? ¿Podría explicárselo a alguien sin conocimientos técnicos? (Si no, vuelve a «Cómo funciona por dentro».)
- [ ] ¿Puedo enumerar al menos tres cosas que la IA **no** es? (Si no, vuelve a «Qué no es la inteligencia artificial».)
- [ ] ¿Puedo enumerar al menos tres cosas que la IA **sí** es? (Si no, vuelve a «Qué sí es la inteligencia artificial».)
- [ ] ¿Conozco los hitos clave de la IA desde 1950 hasta hoy? (Si no, vuelve a la tabla en «Qué sí es la inteligencia artificial».)
- [ ] ¿Sé diferenciar entre elocuencia y corrección en una respuesta generada por IA? (Si no, vuelve a «Dónde solía tropezar yo», error «Confundir elocuencia con corrección».)
- [ ] ¿Podría decidir si un caso necesita SQL, RAG, LLM, herramienta/API, revisión humana o un sistema híbrido? (Si no, vuelve a «Cómo funciona por dentro».)

## En resumen

| IDEA FUERZA | DETALLE |
|---|---|
| La IA no piensa, no entiende y no es consciente. | Es un sistema de predicción estadística de tokens entrenado sobre cantidades masivas de datos. Tratarla como un oráculo es el error más caro que puedes cometer al diseñar sistemas con IA. |
| El mecanismo fundamental es la predicción del siguiente token. | Dada una secuencia, el modelo calcula qué palabra es más probable a continuación. Repetido miles de veces, produce texto coherente. Es álgebra lineal, probabilidad y muestreo. |
| La IA es un amplificador, no un sustituto del criterio humano. | Multiplica la velocidad de quien ya sabe. A quien no sabe, le da una falsa sensación de competencia. Diseña siempre con verificación humana para decisiones de impacto. |
| La elocuencia no es corrección. | Un texto bien escrito y completamente falso es la alucinación por excelencia. Verifica siempre contra fuentes. La confianza con la que el modelo afirma algo no guarda relación con la veracidad de lo que afirma. |

## Para saber más


Anthropic. (2026). *Agent SDK overview*. https://code.claude.com/docs/en/agent-sdk/overview

Bender, E. M., Gebru, T., McMillan-Major, A. y Shmitchell, S. (2021). On the dangers of stochastic parrots: can language models be too big? En *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 610-623). https://doi.org/10.1145/3442188.3445922

Brown, T. B. y otros (2020). Language models are few-shot learners. En *Advances in Neural Information Processing Systems 33* (pp. 1877-1901). https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html

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

Google. (2026). *Agent Development Kit*. https://adk.dev/

Krizhevsky, A., Sutskever, I. y Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. En *Advances in Neural Information Processing Systems 25* (pp. 1097-1105). https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

McCarthy, J., Minsky, M. L., Rochester, N. y Shannon, C. E. (1956). A proposal for the Dartmouth summer research project on artificial intelligence. http://jmc.stanford.edu/articles/dartmouth.html

Model Context Protocol. (2026). *Specification*. https://modelcontextprotocol.io/specification

OpenAI. (2026). *Agents SDK*. https://developers.openai.com/api/docs/guides/agents

OpenAI. (2023). GPT-4 technical report. https://arxiv.org/abs/2303.08774

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.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433-460. https://doi.org/10.1093/mind/LIX.236.433

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

Wolfram, S. (2023). What Is ChatGPT doing… and why does it work? *Stephen Wolfram Writings*. https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/

## Notas


1. Russell, S. y Norvig, P. (2021). *Artificial intelligence: a modern approach* (4.ª ed.). Pearson. Los autores definen la IA como el estudio de agentes que reciben percepciones del entorno y ejecutan acciones, sin atribuir conciencia o intencionalidad al sistema. ↩

2. Bender, E. M., Gebru, T., McMillan-Major, A. y Shmitchell, S. (2021). On the dangers of stochastic parrots: can language models be too big? En *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 610-623). https://doi.org/10.1145/3442188.3445922. Las autoras acuñaron el término «stochastic parrots» para describir el riesgo de atribuir comprensión a modelos que solo reproducen patrones estadísticos. ↩

3. OpenAI. (2026). *Agents SDK*. https://developers.openai.com/api/docs/guides/agents; Anthropic. (2026). *Agent SDK overview*. https://code.claude.com/docs/en/agent-sdk/overview; Google. (2026). *Agent Development Kit*. https://adk.dev/; Model Context Protocol. (2026). *Specification*. https://modelcontextprotocol.io/specification. Fuentes consultadas el 10 de junio de 2026. No se citan como verdad permanente del mercado, sino como fotografía técnica del momento. ↩

4. Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433-460. https://doi.org/10.1093/mind/LIX.236.433 ↩

5. McCarthy, J., Minsky, M. L., Rochester, N. y Shannon, C. E. (1956). A proposal for the Dartmouth summer research project on artificial intelligence. http://jmc.stanford.edu/articles/dartmouth.html ↩

6. 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 ↩

7. Krizhevsky, A., Sutskever, I. y Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. En *Advances in Neural Information Processing Systems 25* (pp. 1097-1105). https://papers.nips.cc/paper/4824 ↩

8. 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 ↩

9. Brown, T. B. y otros (2020). Language models are few-shot learners. En *Advances in Neural Information Processing Systems 33* (pp. 1877-1901). https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html ↩

10. OpenAI. (2023). GPT-4 technical report. https://arxiv.org/abs/2303.08774 ↩

1. Goodfellow, I., Bengio, Y. y Courville, A. (2016). *Deep Learning*. MIT Press. La función softmax es la forma estándar de convertir un vector de puntuaciones en una distribución de probabilidad en la capa de salida de un modelo. https://www.deeplearningbook.org ↩