2.4 Técnicas intermedias y avanzadas de prompting

Técnicas avanzadas de *prompting*

Cómo guiar a la IA en tareas más complejas

A estas alturas ya resulta evidente que **mejorar la forma en que escribimos los *prompts* mejora directamente la calidad de las respuestas**. Esa es la idea central de la ingeniería de *prompts*: no cambiar el modelo, sino **cambiar cómo nos comunicamos con él**.

Los ejemplos vistos hasta ahora permiten resolver tareas habituales, pero cuando el objetivo es más complejo —razonamiento, planificación, análisis profundo, uso de fuentes externas o toma de decisiones guiadas— es necesario recurrir a **técnicas de *prompting* más avanzadas**. Estas técnicas no hacen a la IA "más inteligente", pero sí la **obligan a trabajar de forma más estructurada**, más parecida a cómo razonaría una persona.

> En el contexto educativo, no es necesario dominar todas estas técnicas, pero sí **conocer que existen**, entender para qué sirven y saber en qué situaciones pueden resultar útiles.

Muchas de ellas, además, tienen un gran valor pedagógico porque fomentan procesos como el razonamiento paso a paso, la reflexión o la contrastación de información.

A continuación se presenta una **tabla resumen** con las principales técnicas avanzadas de *prompting* y su finalidad.

Principales técnicas avanzadas de *prompting*

Copyright 2026

Técnica	Idea principal	¿Para qué se utiliza?
Zero-shot prompting	Dar una instrucción sin ejemplos previos	Tareas sencillas y directas
Few-shot prompting	Incluir uno o varios ejemplos	Controlar formato y estilo de salida
Chain of Thought (CoT)	Pedir razonamiento paso a paso	Problemas lógicos y matemáticos
Auto-consistencia	Generar varias soluciones y comparar	Reducir errores en razonamiento
Prompt de conocimiento generado	Pedir primero información base	Mejorar respuestas complejas
Prompt Chaining	Encadenar varios *prompts*	Dividir tareas largas en fases
Tree of Thoughts	Explorar varias líneas de razonamiento	Toma de decisiones complejas
Retrieval Augmented Generation (RAG)	Usar información externa	Trabajar con documentos reales
Automatic Reasoning and Tool-use (ART)	Combinar razonamiento y herramientas	Resolver tareas prácticas
Automatic Prompt Engineering (APE)	Generar y optimizar *prompts*	Ajustar instrucciones automáticamente
Active prompting	Elegir ejemplos más útiles	Mejorar aprendizaje del modelo
Directional Stimulus Prompting	Introducir pistas sutiles	Guiar el tipo de respuesta
Program-Aided Language Models (PAL)	Usar código como apoyo	Matemáticas y lógica formal
ReAct	Combinar razonamiento y acción	Agentes y tareas interactivas
Reflexion	Pedir al modelo que revise errores	Aprendizaje iterativo
Multimodal CoT	Razonar con texto, imagen o audio	Problemas multimodales
Graph prompting	Representar relaciones en forma de grafo	Análisis estructural complejo

Una idea clave antes de profundizar

Aunque la lista pueda parecer extensa, **no todas las técnicas se usan siempre**, ni son necesarias en contextos educativos básicos. Lo importante es entender que:

- cada técnica responde a un tipo de problema,

Copyright 2026

- muchas fomentan procesos cognitivos valiosos,
- y varias pueden adaptarse fácilmente al aula (especialmente *Few-shot*, *Chain of Thought*, *Prompt Chaining* o *Reflexion*).

En los siguientes apartados se irán explicando **las más relevantes desde un punto de vista educativo**, con ejemplos sencillos y aplicaciones prácticas. El objetivo no es convertir al alumnado en ingenieros de *prompts*, sino **enseñar a pensar mejor usando la IA como apoyo**.

Prompt sin entrenamiento previo (*Zero-shot prompting*)

Los modelos de lenguaje actuales han sido entrenados con grandes cantidades de texto y, además, ajustados para **seguir instrucciones escritas en lenguaje natural**. Gracias a ello, son capaces de realizar muchas tareas **sin necesidad de ejemplos previos**. A este tipo de interacción se la denomina **prompt sin entrenamiento previo** o *zero-shot prompting*.

En un *prompt zero-shot* simplemente se le indica a la IA **qué tarea debe realizar**, sin mostrarle ningún ejemplo de cómo hacerlo. El modelo interpreta la instrucción basándose en los patrones generales aprendidos durante su entrenamiento.

Por ejemplo:

Prompt:

> Clasifica el texto en neutral, negativo o positivo.
> Texto: Creo que las vacaciones están bien.
> Sentimiento:

Salida:

> Neutral

En este caso, el modelo ha entendido correctamente la tarea y ha dado una respuesta adecuada **sin haber visto ningún ejemplo**. Esa es precisamente la característica principal del *zero-shot prompting*: confiar en la capacidad general del modelo para interpretar la consigna.

Copyright 2026

¿Por qué funciona el *zero-shot prompting*?

Este comportamiento es posible gracias a dos avances clave en los modelos actuales:

- El **ajuste por instrucciones**, que consiste en entrenar modelos con conjuntos de datos donde las tareas se describen explícitamente mediante instrucciones en lenguaje natural.
- El **aprendizaje por refuerzo con retroalimentación humana (RLHF)**, que alinea las respuestas del modelo con expectativas y preferencias humanas, mejorando su utilidad y coherencia.

Estos procesos permiten que modelos como los que se usan hoy en educación comprendan órdenes del tipo *explica*, *resume*, *clasifica* o *compara* sin necesidad de ejemplos adicionales.

Ejemplo educativo de *zero-shot prompting*

Supongamos una tarea habitual en Educación Secundaria relacionada con comprensión lectora.

Prompt:

> Lee el siguiente texto y di cuál es la idea principal.
> Texto:
> “La energía solar es una fuente renovable que aprovecha la radiación del Sol para generar electricidad o calor. Su uso contribuye a reducir la dependencia de combustibles fósiles y a disminuir la contaminación.”

Salida esperada (ejemplo):

> La energía solar es una fuente renovable que permite obtener energía limpia aprovechando la radiación del Sol.

Aquí no se ha dado ningún ejemplo previo. El alumno simplemente formula una instrucción clara y la IA responde aplicando su conocimiento general. Este tipo de *prompt* es muy útil para:

- tareas sencillas,

Copyright 2026

- primeras aproximaciones a un contenido,
- actividades rápidas de apoyo o repaso.

Ventajas y límites en el contexto educativo

El *zero-shot prompting* tiene varias ventajas en educación:

- es **rápido y sencillo** de usar,
- permite comprobar si una consigna está bien formulada,
- resulta adecuado para tareas básicas o bien definidas.

Sin embargo, también tiene límites:

- puede fallar en tareas complejas,
- puede devolver respuestas con un formato no deseado,
- puede ser ambiguo si la instrucción no es clara.

Cuando el *prompt zero-shot* no ofrece buenos resultados, la solución habitual no es “culpar a la IA”, sino **mejorar la instrucción** o añadir ejemplos. Eso nos lleva a la siguiente técnica: el **prompt con pocos ejemplos (*few-shot prompting*)**, donde se guía al modelo mostrando cómo debe responder.

En el próximo apartado veremos cómo **añadir ejemplos transforma la calidad y el control de las respuestas**, y por qué esta técnica tiene un gran valor pedagógico.

Prompt con pocos ejemplos (*Few-shot prompting*)

Cuando un *prompt sin entrenamiento previo (zero-shot)* no ofrece el resultado esperado —porque la tarea es más compleja, el formato es muy concreto o el contexto es ambiguo— una técnica muy eficaz consiste en **proporcionar uno o varios ejemplos dentro del propio *prompt***. A esta estrategia se la denomina **prompt con pocos ejemplos** o *few-shot prompting*.

En el *few-shot prompting*, el modelo **no aprende de forma permanente**, pero utiliza los ejemplos como guía inmediata para entender **qué tipo de respuesta se espera**, con qué

Copyright 2026

formato y con qué nivel de detalle. En educación, esta técnica resulta especialmente útil porque **funciona igual que el aprendizaje por demostración**: mostrar antes de pedir.

¿Por qué funciona el *few-shot prompting*?

Los modelos de lenguaje son muy sensibles a los patrones. Cuando se incluyen ejemplos, el modelo:

- detecta la estructura de la respuesta,
- identifica el tipo de razonamiento esperado,
- y reproduce el formato con mayor precisión.

Desde un punto de vista pedagógico, esto conecta con una idea clave:
los ejemplos bien escogidos enseñan más que instrucciones largas.

Ejemplo educativo de *few-shot prompting*

Supongamos una tarea de **Lengua en Educación Secundaria**, donde el alumnado debe identificar el tipo de texto.

Prompt (con ejemplos):

> Clasifica los textos como **descriptivo** o **argumentativo**.
>
> Texto: “El transporte público reduce la contaminación y mejora la calidad del aire.” → Argumentativo
> Texto: “La ciudad cuenta con calles amplias, parques verdes y edificios históricos.” → Descriptivo
>
> Texto: “El uso de la bicicleta es una alternativa sostenible para los desplazamientos urbanos.” →

Salida esperada:

> Argumentativo

En este caso, los ejemplos muestran claramente:

- las categorías posibles,
- el formato exacto de la respuesta,
- y el tipo de razonamiento necesario.

Gracias a ello, el modelo responde de forma mucho más ajustada que en un *prompt zero-shot*.

Comparación con *zero-shot prompting*

Si esta misma tarea se planteara sin ejemplos, el modelo podría:

- usar etiquetas diferentes,
- cambiar el formato,
- o interpretar de forma distinta el criterio de clasificación.

El *few-shot prompting* reduce esas variaciones y **aumenta el control sobre la salida**, algo especialmente importante en actividades educativas y evaluativas.

Otro ejemplo educativo: resolución guiada

El *few-shot prompting* también es muy útil para **mostrar cómo razonar**, no solo qué responder.

Prompt:

> “Resuelve el problema siguiendo el mismo razonamiento que en los ejemplos.
>
> Ejemplo:
> Pregunta: ¿Es 14 un número par?
> Respuesta: Sí, porque se puede dividir entre 2 sin resto.
>
> Pregunta: ¿Es 21 un número par?
> Respuesta: No, porque no se puede dividir entre 2 sin resto.

> Pregunta: ¿Es 35 un número par?
> Respuesta:

Salida esperada:

> “ No, porque no se puede dividir entre 2 sin resto.

Aquí, el ejemplo no solo muestra la respuesta correcta, sino **el razonamiento que la acompaña**, algo muy valioso en el aprendizaje.

Ventajas del *few-shot prompting* en educación

El *few-shot prompting* permite:

- mejorar la precisión de las respuestas,
- controlar el formato de salida,
- guiar el razonamiento,
- reducir errores en tareas complejas,
- y fomentar el aprendizaje por imitación consciente.

Por todo ello, es una técnica especialmente recomendable cuando:

- la tarea tiene criterios claros,
- el formato importa,
- o se quiere enseñar “cómo se hace” antes de pedir al alumno que lo aplique.

En el siguiente apartado se abordará una técnica clave para tareas de razonamiento más complejas: **Chain of Thought (*cadena de pensamiento*)**, donde se pide al modelo que explique paso a paso cómo llega a una respuesta.

Chain of Thought (CoT) – Razonar paso a paso

Una de las limitaciones más conocidas de los modelos de lenguaje es que, cuando se les pide directamente una respuesta, **pueden acertar por casualidad o fallar sin que sepamos por qué**. Para abordar este problema surge una de las técnicas más importantes del *prompting* avanzado: **Chain of Thought** o **cadena de pensamiento**.

La idea es muy sencilla y, al mismo tiempo, muy poderosa:
en lugar de pedir solo la respuesta final, **se le pide al modelo que muestre el razonamiento paso a paso** que le lleva a esa respuesta.

En educación, esta técnica es especialmente valiosa porque conecta directamente con una práctica pedagógica clásica: **valorar el proceso, no solo el resultado**.

¿Qué es el *Chain of Thought*?

El *Chain of Thought prompting* consiste en **indicar explícitamente al modelo que razone en pasos intermedios**, explicando cómo llega a una conclusión. No se trata de hacer a la IA "más inteligente", sino de **forzar una estructura de razonamiento más ordenada y transparente**.

Este enfoque mejora notablemente el rendimiento del modelo en tareas como:

- problemas matemáticos,
- lógica,
- comprensión lectora compleja,
- toma de decisiones,
- análisis de causas y consecuencias.

¿Por qué funciona?

Cuando se pide una respuesta directa, el modelo puede intentar "adivinar" la salida más probable. En cambio, cuando se le pide que razone paso a paso:

- reduce errores,
- organiza mejor la información,
- y produce respuestas más coherentes.

Desde un punto de vista educativo, esto tiene un valor añadido:
el razonamiento se hace visible, lo que permite analizarlo, corregirlo y aprender de él.

Copyright 2026

Ejemplo educativo sin *Chain of Thought* (puede fallar)

Prompt:

> ¿La suma de los números impares siguientes es par o impar?
> 15, 32, 5, 13, 82, 7, 1

Respuesta posible:

> Es par.

La respuesta es incorrecta, pero no sabemos por qué el modelo ha fallado.

El mismo ejemplo con *Chain of Thought*

Prompt:

> Resuelve el problema paso a paso.
>
> 1. Identifica los números impares.
> 2. Súmalos.
> 3. Indica si el resultado es par o impar.
>
> Lista: 15, 32, 5, 13, 82, 7, 1

Salida:

> Números impares: 15, 5, 13, 7, 1
> Suma: 41
> 41 es impar.

Aquí el razonamiento es claro, verificable y correcto.

Ejemplo educativo en Lengua: comprensión lectora

Prompt:

> Lee el texto y responde razonando paso a paso:
>
> “El uso del transporte público reduce la contaminación porque disminuye el número de vehículos privados en circulación.”
>
> Pregunta: ¿Cuál es la causa y cuál es la consecuencia?

Salida esperada:

> Primero identifico la causa: disminuir el número de vehículos privados.
> Luego identifico la consecuencia: reducir la contaminación.
> Por tanto, la causa es la disminución de vehículos y la consecuencia es la reducción de la contaminación.

Este tipo de *prompt* ayuda al alumnado a **estructurar el pensamiento**, no solo a señalar una respuesta.

Copyright 2026

Chain of Thought en evaluación y autoevaluación

El *Chain of Thought* también puede usarse para:

- justificar respuestas,
- explicar decisiones,
- revisar errores,
- comparar razonamientos.

Por ejemplo:

> *Explica paso a paso por qué esta respuesta es correcta o incorrecta según los criterios de evaluación.*

Así, la IA se convierte en una **herramienta de reflexión**, no solo de corrección.

Ventajas del *Chain of Thought* en educación

El uso de esta técnica permite:

- mejorar la fiabilidad de las respuestas,
- hacer visible el proceso mental,
- fomentar el pensamiento lógico y crítico,
- reducir respuestas automáticas,
- y reforzar la idea de que **razonar es más importante que acertar**.

Además, encaja perfectamente con metodologías educativas que valoran:

- el razonamiento,
- la argumentación,
- y la explicación del procedimiento.

Copyright 2026

Idea clave para el aula

> **Pedir que se explique el razonamiento es enseñar a pensar, no solo a responder.**

El *Chain of Thought prompting* no solo mejora a la IA; **mejora el aprendizaje humano** al poner el foco en el proceso.

Retrieval Augmented Generation (RAG)

Cuando la IA consulta fuentes reales para responder mejor

Hasta ahora hemos visto técnicas de *prompting* que permiten **guiar el razonamiento y la forma de responder** de los modelos de lenguaje. Sin embargo, todas ellas comparten una limitación importante: el modelo responde **solo con el conocimiento que ya tiene** incorporado desde su entrenamiento. Ese conocimiento es amplio, pero **no es infinito, no siempre está actualizado y no siempre es específico**.

Para abordar tareas que requieren información más precisa, actualizada o contextualizada, surge una de las técnicas más importantes del *prompting* avanzado y de los sistemas modernos de IA: la **Generación Aumentada por Recuperación**, conocida como **RAG (*Retrieval Augmented Generation*)**.

¿Qué es RAG?

La **Generación Aumentada por Recuperación (RAG)** es un enfoque que combina dos elementos:

1. **Un sistema de recuperación de información**, que busca documentos relevantes en fuentes externas (por ejemplo, una base de datos, apuntes, PDFs, normativa, Wikipedia o documentos del centro educativo).

2. **Un modelo de lenguaje generativo**, que utiliza esos documentos como contexto para generar la respuesta final.

En lugar de responder solo “de memoria”, el sistema **consulta primero información real**, la incorpora al *prompt* y después genera una respuesta basada en esos datos. De este modo, la IA no solo “habla bien”, sino que **habla apoyándose en fuentes concretas**.

¿Por qué es necesaria la técnica RAG?

Los modelos de lenguaje de propósito general funcionan muy bien para tareas como:

- resumir textos,
- clasificar información,
- analizar sentimientos,
- responder preguntas generales.

Estas tareas no suelen requerir conocimientos externos adicionales. Sin embargo, cuando la tarea exige:

- datos actualizados,
- conocimiento especializado,
- información concreta de un documento,
- o fidelidad factual,

el modelo puede cometer errores o **“alucinar”**, es decir, generar respuestas plausibles pero incorrectas.

RAG surge precisamente para **reducir este problema** y aumentar:

- la **consistencia factual**,
- la **fiabilidad de las respuestas**,
- y la **transparencia del proceso**.

¿Cómo funciona RAG de forma sencilla?

El funcionamiento de RAG puede entenderse en cuatro pasos:

1. El usuario plantea una pregunta o tarea.

2. El sistema busca documentos relevantes en una fuente externa.
3. Esos documentos se añaden como contexto al *prompt*.
4. El modelo genera la respuesta basándose en ese contexto.

Lo importante es que **no es necesario reentrenar el modelo** cada vez que cambia la información. Basta con actualizar los documentos a los que tiene acceso el sistema. Esto hace que RAG sea especialmente útil en contextos donde el conocimiento evoluciona con el tiempo.

Ejemplo educativo de RAG

Imaginemos un centro educativo que quiere usar IA para responder preguntas sobre su **normativa interna** o sobre un **currículo oficial**.

Pregunta del alumno:

> “ ¿Cuáles son los criterios de evaluación del tema 3?

Un modelo sin RAG podría:

- inventar criterios,
- mezclar información genérica,
- o dar una respuesta poco precisa.

Con RAG, el sistema:

- busca el documento oficial del currículo,
- localiza el apartado del tema 3,
- y genera una respuesta basada únicamente en ese texto.

Así, la IA actúa como un **asistente documental**, no como una fuente inventada de conocimiento.

Aplicaciones educativas claras de RAG

En educación, RAG abre posibilidades muy valiosas:

- **Consultar apuntes y materiales del profesor**
  El alumnado puede hacer preguntas sobre documentos reales de la asignatura.
- **Trabajar con normativa educativa**
  Responder dudas basándose en leyes, decretos o instrucciones oficiales.
- **Apoyo al estudio con fuentes controladas**
  Evitar que el alumnado use información incorrecta o descontextualizada.
- **Proyectos y trabajos de investigación**
  Generar respuestas citando y usando documentos concretos.
- **Atención a la diversidad**
  Explicar el mismo contenido usando siempre la misma base documental, pero con distinto nivel de lenguaje.

RAG y pensamiento crítico

Desde un punto de vista pedagógico, RAG tiene un valor añadido muy importante: **devuelve protagonismo a las fuentes**.

El alumnado aprende que:

- las respuestas deben basarse en textos,
- la información debe poder rastrearse,
- y no todo lo que “suena bien” es verdadero.

Además, el profesorado puede exigir que la IA:

- indique en qué documento se basa,
- cite fragmentos,
- o limite sus respuestas a una fuente concreta.

Esto refuerza competencias clave como la **alfabetización informacional** y el **pensamiento crítico**.

Ventajas de RAG frente a otros enfoques

La técnica RAG permite:

- reducir alucinaciones,
- trabajar con información actualizada,

- adaptar la IA a contextos específicos,
- modificar el conocimiento sin reentrenar el modelo,
- y aumentar la confianza en las respuestas.

Por eso, RAG es una de las bases de muchos sistemas educativos avanzados basados en IA.

Idea clave para el aula

> **La IA no debe sustituir a las fuentes,**
> **sino aprender a trabajar con ellas.**

La Generación Aumentada por Recuperación convierte a la IA en una **herramienta de acceso y comprensión de documentos**, no en una autoridad absoluta. En educación, esto marca una diferencia fundamental: **aprender con información fiable es tan importante como aprender a formular buenas preguntas**.

Revision #6
Created 2025-12-17 18:25:21 CET by Maria
Updated 2026-03-15 13:56:39 CET by Maria

Copyright 2026