



# Aprendizaje por Contexto

Luis Miranda, Estudiante PhD  
[lmirandn@uc.cl](mailto:lmirandn@uc.cl)

# Outline

1. ¿Qué es Aprendizaje por Contexto y qué es un prompt?
2. Tipos de prompting:
  - Zero-shot y Few-Shot
  - Cadena de Pensamiento
3. Buenas prácticas
4. Estrategias avanzadas
  - Generación por Recuperación Aumentada (RAG)
  - Auto Consistencia (Self-Consistency)
  - Razonar y Actuar (ReAct)
5. Consideraciones
6. Conclusiones

# ¿Qué es Aprendizaje por Contexto?

Aprendizaje por Contexto o In-Context Learning (ICL) es la capacidad de un LLM de generalizar y resolver tareas, sin necesidad de ajustar sus parámetros internos.

- No requiere entrenamiento.
- La precisión mejora al presentar ejemplos bien estructurados.
- Se basa en mecanismos de atención que detectan y replican patrones.

# ¿Qué es Aprendizaje por Contexto?

Aprendizaje por Contexto o In-Context Learning (ICL) es la capacidad de un LLM de generalizar y resolver tareas, sin necesidad de ajustar sus parámetros internos.

- No requiere entrenamiento.
- La precisión mejora al presentar ejemplos bien estructurados.
- Se basa en mecanismos de atención que detectan y replican patrones.

Aspect	Fine-Tuning	In-Context Learning
Performance	High precision for domain-specific tasks	Flexible, great for prototyping and low-resource tasks
Cost	High training cost, low inference cost	Low training cost, high inference cost
Data Requirements	Substantial labeled data required	Minimal data, relies on examples in the prompt

# ¿Qué es un prompt?

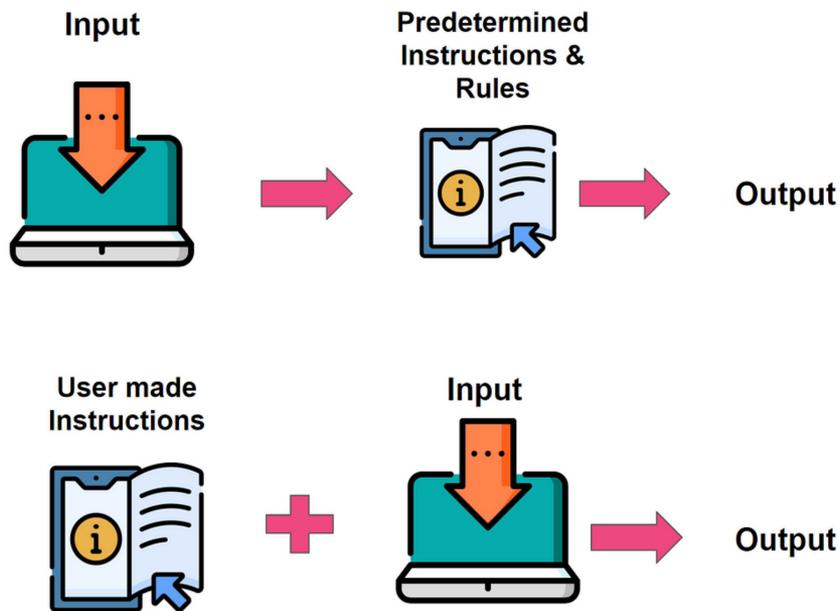
Un *prompt* en ICL es una entrada en lenguaje natural que se entrega a un modelo de lenguaje para especificar una tarea, proporcionar contexto y guiar la generación de una respuesta.

- Es la forma principal de interactuar con un LLM.
- La calidad del output depende fuertemente del diseño del prompt.
- Permite adaptar un modelo general a tareas específicas sin modificarlo.

# ¿Qué es un prompt?

Un *prompt* en ICL es una entrada en lenguaje natural que se entrega a un modelo de lenguaje para especificar una tarea, proporcionar contexto y guiar la generación de una respuesta.

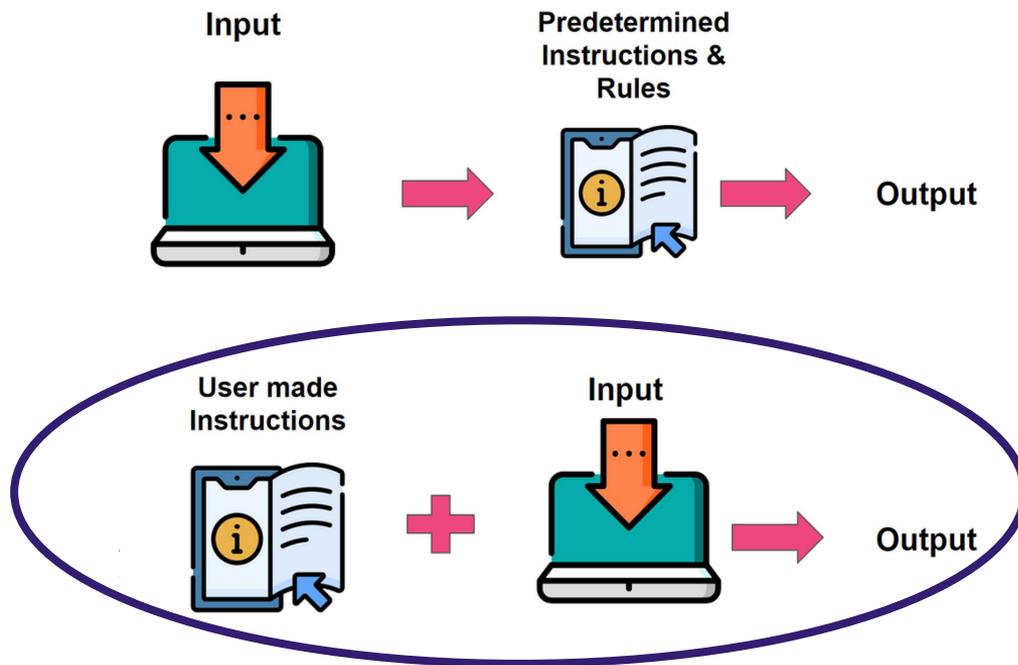
- Es la forma principal de interactuar con un LLM.
- La calidad del output depende fuertemente del diseño del prompt.
- Permite adaptar un modelo general a tareas específicas sin modificarlo.



# ¿Qué es un prompt?

Un *prompt* en ICL es una entrada en lenguaje natural que se entrega a un modelo de lenguaje para especificar una tarea, proporcionar contexto y guiar la generación de una respuesta.

- Es la forma principal de interactuar con un LLM.
- La calidad del output depende fuertemente del diseño del prompt.
- Permite adaptar un modelo general a tareas específicas sin modificarlo.



# Prompt Engineering

Prompt Engineering es la práctica de diseñar, estructurar y optimizar instrucciones en lenguaje natural para guiar de manera precisa el comportamiento de un modelo de lenguaje, logrando que genere respuestas útiles, coherentes y alineadas con una tarea específica.

## Tipos de Prompting

- Zero-Shot
- Few-shot
- Cadena de Pensamiento

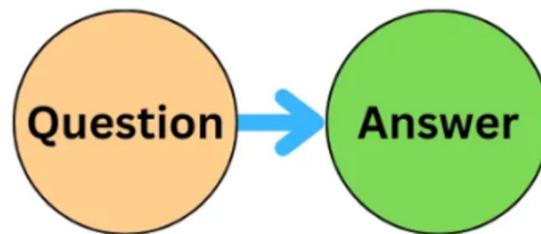
# Zero-Shot Learning

Zero-Shot learning es una técnica en la que se proporciona al modelo de lenguaje una instrucción o tarea específica **sin incluir ejemplos previos** en el prompt. El modelo debe resolver la tarea basándose únicamente en su conocimiento aprendido durante el entrenamiento y la interpretación semántica del enunciado.

# Zero-Shot Learning

Zero-Shot learning es una técnica en la que se proporciona al modelo de lenguaje una instrucción o tarea específica **sin incluir ejemplos previos** en el prompt. El modelo debe resolver la tarea basándose únicamente en su conocimiento aprendido durante el entrenamiento y la interpretación semántica del enunciado.

- Sin demostraciones
- Dependencia del pre entrenamiento
- Rápido de implementar
- Menor precisión en tareas complejas
- Amplio uso cotidiano



**Zero-shot**

# Zero-Shot Learning

¿Consideran correcta esta respuesta?

*“Paciente con fiebre, tos y dificultad respiratoria. Clasifícalo.”*

**Modelo:** Sistema Respiratorio

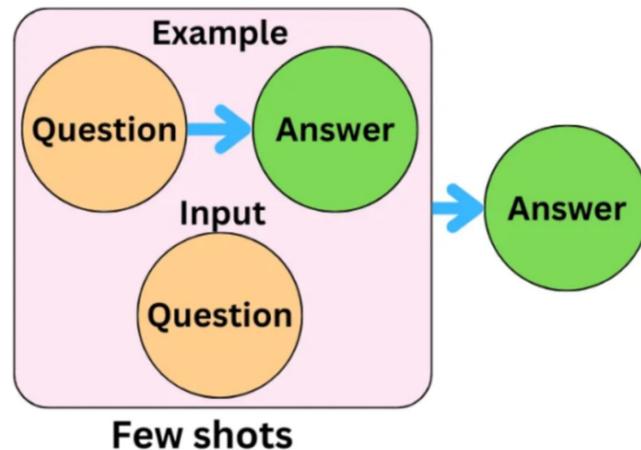
# Few-Shot Learning

Few-Shot learning consiste en proporcionar una pequeña cantidad de ejemplos de entrada y salida dentro del prompt, para que el modelo infiera la estructura y lógica de la tarea antes de generar una respuesta para un nuevo caso.

# Few-Shot Learning

Few-Shot learning consiste en proporcionar una pequeña cantidad de ejemplos de entrada y salida dentro del prompt, para que el modelo infiera la estructura y lógica de la tarea antes de generar una respuesta para un nuevo caso.

- Ayuda al modelo a entender el formato, tipo de salida y tarea.
- Puede reducir errores y ambigüedades.
- Proporciona un marco más claro que el zero-shot.
- Con 2 o 3 ejemplos basta.



# Few-Shot Learning

Ejemplo:

Tarea: Categoriza los siguientes síntomas:

Ejemplo 1: síntomas: fiebre + exantema → categoría: infeccioso

Ejemplo 2: síntomas: dolor lumbar + disuria → categoría: urológico

Nuevo caso: síntomas: dolor torácico + palpitaciones →

# Few-Shot Learning

Ejemplo:

Tarea: Categoriza los siguientes síntomas:

Ejemplo 1: síntomas: fiebre + exantema → categoría: infeccioso

Ejemplo 2: síntomas: dolor lumbar + disuria → categoría: urológico

Nuevo caso: síntomas: dolor torácico + palpitaciones → **categoría: cardiaca**

# Ejercicio práctico: Clasificación

- Actividad: define 2 ejemplos que ayuden a resolver la tarea
- Probar zero-shot vs few-shot en Chat GPT [chatgpt.com](https://chatgpt.com)

```
prompt = """Transforma una nota clínica en texto libre en un formulario estructurado con los siguientes campos:
```

```
1-Motivo de consulta 2-Especialidad requerida 3-Edad del paciente 5-Sexo del paciente
```

```
input:
```

```
El paciente dice que tiene mucho dolor en la zona lumbar desde hace dos semanas. Ha tomado paracetamol sin mejoras. Pide ver a un especialista pronto. Es un hombre de unos 50 años"""
```

**output esperado:**

```
{
  "Motivo de consulta": "Dolor lumbar intenso desde hace dos semanas sin mejoría con paracetamol",
  "Especialidad requerida": "Traumatología",
  "Edad": 50,
  "Sexo": "Masculino"
}
```

# Cadena de Pensamiento

“Cadena de Pensamiento” es una técnica que induce al modelo a generar explícitamente pasos intermedios de razonamiento antes de dar una respuesta final, ayudando en tareas que requieren lógica o toma de decisiones complejas.

# Cadena de Pensamiento

“Cadena de Pensamiento” es una técnica que induce al modelo a generar explícitamente pasos intermedios de razonamiento antes de dar una respuesta final, ayudando en tareas que requieren lógica o toma de decisiones complejas.

- Se activa con frases como “piensa paso a paso” o dándole ejemplos en donde la respuesta contenga una cadena de pensamiento.
- Útil para obtener una explicación del resultado.
- Mejora el rendimiento en tareas de razonamiento lógico.
- Hace más transparente el proceso de generación.

# Cadena de Pensamiento

“Cadena de Pensamiento” prompting es una técnica que induce al modelo a generar explícitamente pasos intermedios de razonamiento antes de dar una respuesta final, ayudando en tareas que requieren lógica o toma de decisiones complejas.

## Standard Prompting

### Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The answer is 27. ❌

## Chain of Thought Prompting

### Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. ✅

# Outline

1. ¿Qué es Aprendizaje por Contexto y qué es un prompt?
2. Tipos de prompting:
  - Zero-shot y Few-Shot
  - Cadena de Pensamiento
- 3. Buenas prácticas**
4. Estrategias avanzadas
  - Generación por Recuperación Aumentada (RAG)
  - Auto Consistencia (Self-Consistency)
  - Razonar y Actuar (ReAct)
5. Consideraciones
6. Conclusiones

# Entregar un rol

Especificar un rol al modelo dentro del prompt consiste en indicarle explícitamente una identidad o especialidad para influir en el tono, vocabulario y profundidad de la respuesta.

# Entregar un rol

Especificar un rol al modelo dentro del prompt consiste en indicarle explícitamente una identidad o especialidad para influir en el tono, vocabulario y profundidad de la respuesta.

 Prompt 

Eres un experto en comunicaciones. Redacta un correo para un cliente informándole sobre un retraso en el calendario de entrega debido a problemas logísticos.

 AI Output 

**Asunto: Actualización Importante de su Pedido**

Estimado/a [Nombre del Cliente]: Espero que este correo le encuentre bien. Me comunico con usted para notificarle sobre un retraso en la entrega de su pedido. Lamentablemente, hemos encontrado un problema logístico que está causando una pequeña demora. Aunque estamos haciendo todo lo posible por resolver esta situación rápidamente, anticipamos que su pedido será entregado dentro de [nueva fecha estimada de entrega]. Pedimos disculpas por cualquier inconveniente ...

 Prompt 

Eres un experto en marketing. Redacta un correo para un cliente informándole sobre un retraso en el calendario de entrega debido a problemas logísticos.

 AI Output 

**Asunto: Un Poco Más de Espera por Sus Maravillosos Productos**

Estimado/a [Nombre del Cliente]: ¡Esperamos que este mensaje le encuentre lleno de entusiasmo por su próximo pedido! Sin embargo, incluso los planes mejor organizados a veces enfrentan contratiempos inesperados. Hemos tenido un inconveniente logístico ...

# Respuesta estructurada y Restringida

Solicitar una respuesta estructurada consiste en pedir al modelo que entregue la salida en un formato predefinido, como listas, tablas, o formularios, para facilitar su interpretación o integración.

- Ejemplo: “Da el diagnóstico y tratamiento de la consulta, pero **responde en el siguiente formato:** {diagnóstico, tratamiento}”

# Respuesta estructurada y Restringida

Solicitar una respuesta estructurada consiste en pedir al modelo que entregue la salida en un formato predefinido, como listas, tablas, o formularios, para facilitar su interpretación o integración.

- Ejemplo: “Da el diagnóstico y tratamiento de la consulta, pero **responde en el siguiente formato**: {diagnóstico, tratamiento}”

A veces, queremos que el modelo responda un set de respuestas posibles. Para esto podemos pedir que su respuesta se acote a un set de respuestas posibles.

- Problema de clasificación: “Responde con una de estas respuestas: NO GES, GES “
- Pregunta de Alternativas : “Responde con a), b) , c) o d)”

# Plantillas reutilizables

Son estructuras predefinidas de prompts con partes fijas e identificadores de contenido (slots), que permiten aplicar el mismo esquema a diferentes entradas de manera consistente.

- Definir templates con slots: "Nota: {input}. Resumen clínico:"
- Asegura uniformidad y ahorro de tiempo

```
notas_clinicas = ["Paciente viene por ...", "Paciente presenta dolor ...", ...]

for nota in notas_clinicas:

    input = nota

    prompt_template = f"Nota: {input}. Resumen clínico:"

    ...
```

# Temperatura

La *temperatura* es un parámetro que controla la aleatoriedad en la generación de texto. Valores bajos (p. ej., 0.2) hacen al modelo más conservador; valores altos (p. ej., 0.8) lo hacen más creativo.

- En salud, se recomienda usar 0–0.3 para evitar alucinaciones.
- Controla la diversidad de las respuestas.
- No afecta la longitud, sólo la variedad.

## LLM Temperature Settings

Deterministic, repetitive



0



More creative, random



1

# Outline

1. ¿Qué es Aprendizaje por Contexto y qué es un prompt?
2. Tipos de prompting:
  - Zero-shot y Few-Shot
  - Cadena de Pensamiento
3. Buenas prácticas
- 4. Estrategias avanzadas**
  - **Generación por Recuperación Aumentada (RAG)**
  - **Auto Consistencia (Self-Consistency)**
  - **Razonar y Actuar (ReAct)**
5. Consideraciones
6. Conclusiones

# Retrieval Augmented Generation (RAG)

RAG es una técnica que combina generación de lenguaje con recuperación de documentos relevantes (p. ej., guías clínicas), integrando esa evidencia en el prompt para fundamentar mejor la respuesta.

- Incluye fragmentos relevantes en el prompt con una fase previa de búsqueda
- Reduce alucinaciones y aporta evidencia

# Retrieval Augmented Generation (RAG)

Ejemplo:



El paciente vino al Hospital ABC tiene fiebre, qué farmaco hay que darle?

Hospital ABC:

doc\_1

doc\_2

doc\_3

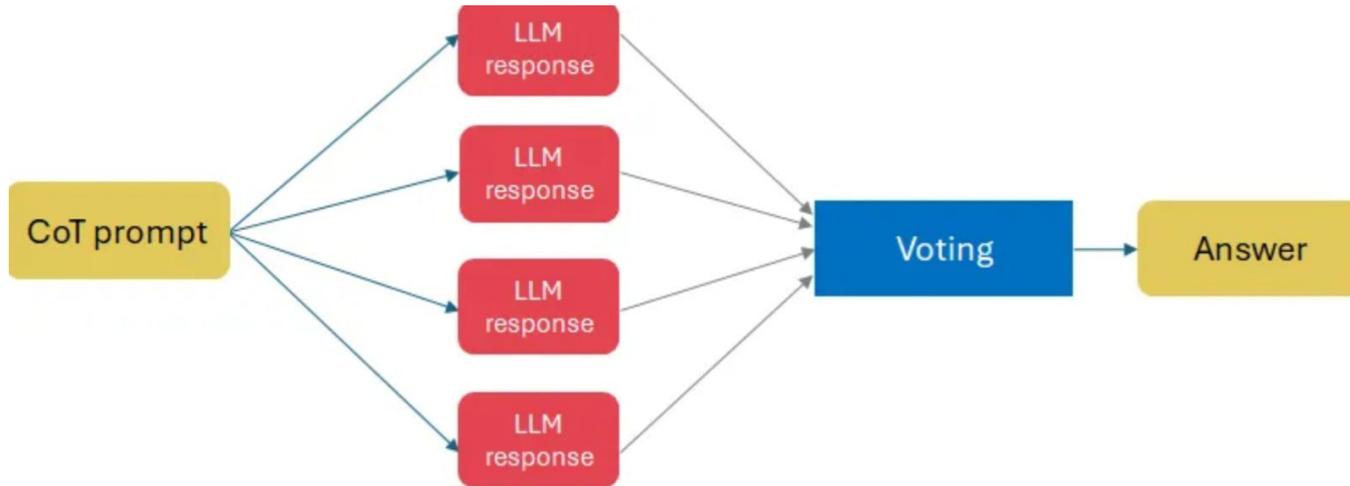
...

doc\_3: En hospital ABC se tiene como fármaco de primera línea para la fiebre el metamizol

**Metamizol**

# Self-Consistency

- Generar múltiples CoT y extraer respuestas
- Votar la respuesta mayoritaria
- Aumenta robustez en respuesta



# Reasoning and Acting (ReAct)

ReAct es un enfoque que interleva razonamiento textual (Chain-of-Thought) con acciones externas (consultas a APIs, bases de datos, motores de búsqueda) dentro de un mismo flujo de respuesta.

- Alterna pasos de pensamiento y acción (e.g., búsqueda).
- Permite verificación de información en tiempo real.
- Útil para tareas que requieren datos actualizados o verificados.
- Mejora la transparencia y control sobre decisiones automatizadas.
- Lo más utilizado hoy en día

# Outline

1. ¿Qué es Aprendizaje por Contexto y qué es un prompt?
2. Tipos de prompting:
  - Zero-shot y Few-Shot
  - Cadena de Pensamiento
3. Buenas prácticas
4. Estrategias avanzadas
  - Generación por Recuperación Aumentada (RAG)
  - Auto Consistencia (Self-Consistency)
  - Razonar y Actuar (ReAct)
- 5. Consideraciones**
6. Conclusiones

# Costos de cómputo

## GPT-4.1

El modelo más inteligente para tareas complejas

### Precio

#### Entrada:

USD 2.00/1 millón de tokens

#### Entrada en caché:

USD 0.50/1 millón de tokens

#### Salida:

USD 8.00/1 millón de tokens

## GPT-4.1 mini

Modelo asequible que equilibra velocidad e inteligencia

### Precio

#### Entrada:

USD 0.40/1 millón de tokens

#### Entrada en caché:

USD 0.10/1 millón de tokens

#### Salida:

USD 1.60/1 millón de tokens

## GPT-4.1 nano

El modelo más rápido y rentable para tareas de baja latencia

### Precios

#### Entrada:

USD 0.100/1 millón de tokens

#### Entrada en caché:

USD 0.025/1 millón de tokens

#### Salida:

USD 0.400/1 millón de tokens

## OpenAI o3

Nuestro modelo con capacidad de razonamiento más poderoso con un destacado rendimiento en programación, matemáticas, ciencia y visión

### Precio

#### Entrada:

USD 10.00/1 millón de tokens

#### Entrada en caché:

USD 2.50/1 millón de tokens

#### Salida:

USD 40.00/1 millón de tokens

## OpenAI o4-mini

Nuestro modelo con capacidad de razonamiento más rápido y rentable con un sólido rendimiento en matemáticas, programación y visión

### Precio

#### Entrada:

USD 1.000/1 millón de tokens

#### Entrada en caché:

USD 0.275/1 millón de tokens

#### Salida:

USD 4.400/1 millón de tokens

1 millón de  
tokens  $\approx$  2000  
páginas en word

# Alucinaciones

Una *alucinación* ocurre cuando un modelo de lenguaje genera información falsa, inventada o no respaldada por evidencia, a pesar de que la forma del texto parezca creíble o confiable.

- Error sintético, no intencional: el modelo no “miente”, simplemente completa texto según patrones estadísticos.
- Peligroso en salud: puede inventar diagnósticos, dosis, interacciones o citas médicas.
- Más común en zero-shot o sin recuperación (RAG).
- Reducir alucinaciones:
  - Seguir buenas prácticas!
  - Validar con expertos humanos

# Outline

1. ¿Qué es Aprendizaje por Contexto y qué es un prompt?
2. Tipos de prompting:
  - Zero-shot y Few-Shot
  - Cadena de Pensamiento
3. Buenas prácticas
4. Estrategias avanzadas
  - Generación por Recuperación Aumentada (RAG)
  - Auto Consistencia (Self-Consistency)
  - Razonar y Actuar (ReAct)
5. Consideraciones
- 6. Conclusiones**

# Conclusiones

## **El prompt es el puente entre el usuario y el modelo**

- Con instrucciones bien diseñadas, se puede automatizar tareas sin entrenar nada.

## **In-Context Learning permite adaptar el modelo en tiempo real**

- Zero-shot, few-shot y Cadena de Pensamiento son estrategias clave para lograrlo.

## **El diseño importa: buenas prácticas mejoran resultados**

- Roles, estructuras y plantillas hacen las respuestas más útiles.

## **Estrategias avanzadas como ReAct, RAG o Auto-Consistencia aumentan robustez**

- razonar y actuar, incorporar evidencia externa y votar mejora la fiabilidad y calidad.

## **Cuidar alucinaciones y costos es parte de usar bien los LLMs**

- Prompting claro, recuperación de evidencia y eficiencia computacional son claves prácticas.