

## Como?

2x: Auto-complete

**5x:** Structured outputs

10x: Function Calling

20x: Chaining y Workflows

50x: Agents

100x:?





- Las herramientas de IA actuales son primitivas somos early adopters o casi...
- La IA no reemplaza al dev, amplifica su impacto
   El dev se convierte más en diseñador de soluciones
- Las empresas ya la están usando, no están esperando Google, Microsoft, Amazon y miles de startups están integrando IA generativa
- El valor de quien sabe de IA está subiendo cada mes
   Tener skills en IA es hoy una ventaja. Mañana será un requisito

...

## Objetivos

Nuestro objetivo para las próximas 3 horas es que salgan de aquí pudiendo...

- Identificar la complejidad de tus tareas (modelo de dominio inspirado en Cynefin + CoT estimado).
- @ Elegir el LLM y estrategia de Prompt adecuados (desde prompting básico hasta agentes con memoria).
- \_\_\_\_\_ Diseñar prompts efectivos y adaptarlos al modelo (4-Level Framework).
- K Esbozar workflows simples (chaining y tool use básicos).
- M Construir agentes básicos siguiendo buenas prácticas.
- Apliquen estos conceptos en ejercicios prácticos y vean demostraciones con código real.
- o al menos algo de todo lo anterior...

La meta es que sepan elegir la herramienta correcta para el problema correcto, y cómo construir la solución...

# agenda

- Introducción:

  IA Generativa fundamentos
- 2 Framework: CoT -> Modelo -> Prompt
- Demos y Práctica: ¡Hands-on! [Prompting, JSON, Chaining, ...]
- 4 Ejercicio Grupal:
  Diseño [Colaborativo]
- Best Practices y Q&A: Charla abierta



MODEL

reasoning / non-reasoning ? that is the question...

## fundamentos

- Expansión
- Compresión
- Conversión
- Búsqueda
- Acción
- Razonamiento

**PROMPT** 

son el nuevo código dominarlos es esencial **CONTEXT** 

es la RAM para los LLM el scope de la conversación



## Complejidad → Modelo → Estrategia

cómo la **complejidad** de la tarea (estimada por pasos) dicta la **elección** del tipo de **LLM** y, crucialmente, el **estilo de prompting** a utilizar.

Naturaleza de la tarea (Cynefin)

Simple

Complicado

Complejo

Caótico

### Estimación de pasos CoT

- < 3 Pasos; tarea directa</p>
- 3 5 Pasos; análisis, seguir reglas
- > 5 Pasos; creatividad, planificación
  - ; respuesta inmediata

**R:** Nivel 3 o 4

**N-R:** Nivel 1 o 2

Nivel 2

Nivel 1 o 2

N-R: detallado

R: minimalista (x los

MOE)

minimalista, zero-shot

directivo, conciso

# Adaptando tu Enfoque

3 - 5

> 5

Caótico

Complicado

Complejo

Caótico

Pasos Estimados	Dominio	Modelo	Estilo prompt	Nivel prompt
< 3	Simple	N-R	Directo, claro, few-shot (con algún ejemplo)	Nivel 2 o 3

N-R / R compensando

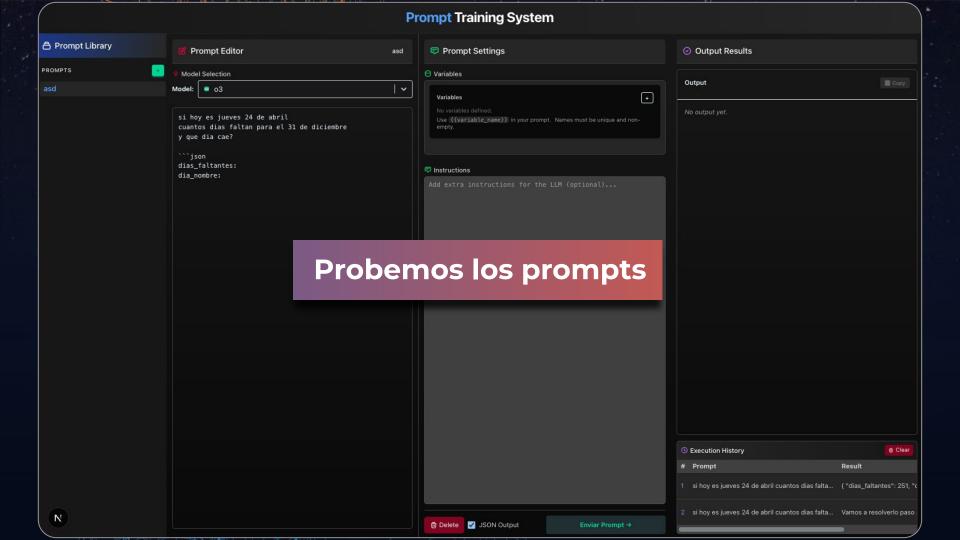
R ideal

Algún N-R rápido

# 4-Niveles de prompt

1	<ul> <li>Prompts rápidos en lenguaje natural para prototipado ágil</li> <li>Perfecto para explorar capacidades y comportamientos del modelo</li> <li>Ideal para tareas puntuales y experimentación</li> </ul>		
2	- Prompts reutilizables con propósito e instrucciones claras - Usa formato XML/estructurado para mejor rendimiento del modelo - Contiene variables estáticas que se pueden modificar - Resuelve problemas bien definidos y repetibles		
3	- Se basa en el Nivel 2 añadiendo ejemplos de salida - Los ejemplos guían al modelo para producir formatos específicos - Aumenta la consistencia y confiabilidad de las salidas - Perfecto cuando el formato de salida es importante		
4	- Prompts listos para producción con variables dinámicas - Se pueden integrar en código y aplicaciones - Base para construir herramientas y agentes potenciados por IA		





probar: https://latitude.so/

### Resumiendo algunas ideas

- La selección del modelo adecuado tiene un impacto significativo en el rendimiento de tus prompts y resultados.
- Un **propósito claro** para tu prompt es crucial para lograr resultados deseados de manera efectiva.
- Las variables dinámicas permiten la reutilización de prompts y adaptabilidad en diferentes contextos y aplicaciones.
- Incluir ejemplos concretos en los prompts ayuda a especificar el formato y estructura de la salida deseada.
- Los tipos de salida pueden ser texto o JSON, lo que influye en la confiabilidad de los resultados generados.
- La combinación de modelo, propósito, variables, ejemplos y tipo de salida crea prompts efectivos.
- Los prompts de alta calidad pueden producir el 80% de los resultados con solo el 20% del esfuerzo invertido.
- Las variables estáticas se fijan durante el desarrollo, mientras que las variables dinámicas cambian según el contexto.
- El **chaining** de prompts permite la integración de salidas de un prompt como entradas para otro.
- Las salidas en formato **JSON** proporcionan datos estructurados, fundamentales para construir sistemas y flujos de trabajo de IA complejos.



#### Actividad

Ver la *Plantilla de tarea grupal*, armar grupos si quieren, proponer una task para integrar con IA, completar la planilla.

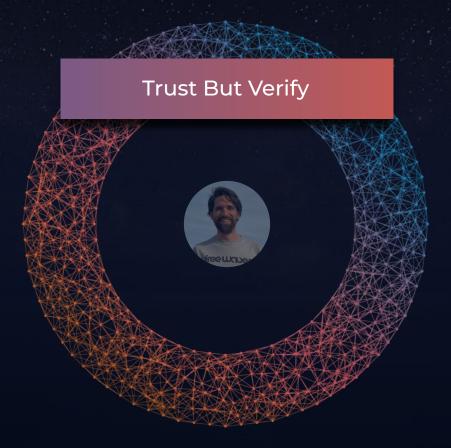
Mientras pueden preguntar, discutir, proponer ideas alrededor.

Por último pueden exponer el resultado para su tarea al resto.



#### Sources

- [OpenAi API docs] https://platform.openai.com/docs/overview
- [Gemini API docs] https://ai.google.dev/gemini-api/docs
- [Anthropic courses] https://github.com/anthropics/courses
- [Anthropic effective agents] https://www.anthropic.com/engineering/building-effective-agents
- [indydevdan yt ch] https://www.youtube.com/@indydevdan
- [deeplearning courses] https://learn.deeplearning.ai/
- [Cynefin model explanation] https://www.javiergarzas.com/2016/07/entendiendo-modelo-cynefin.html
- [HF Smolagents] https://huggingface.co/docs/smolagents/
- [Prompting guide] https://www.promptingguide.ai/



Matias Gummá - Agrosistemas