

# **I Context Engineering: Core**

Versión V1.0

# Índice

00 Recordatorio

01 Contexto del Código

02 Ingeniería del Contexto (Context Engineering)

# Leyenda

## A nivel de Contenido



Cuadros de Contexto



Problemas

Texto Objetivos

Texto Ejemplos explicativos



Detalles

## A nivel de Tipo de Slide



Tipo "Curiosidades"



Tipo "Ejemplo"

Los recursos utilizados se han obtenido de

- <https://storyset.com> : Ilustraciones personalizables gratuitas
- <https://lexica.art> : IA que genera imágenes
- <https://www.freepik.es>: Iconos

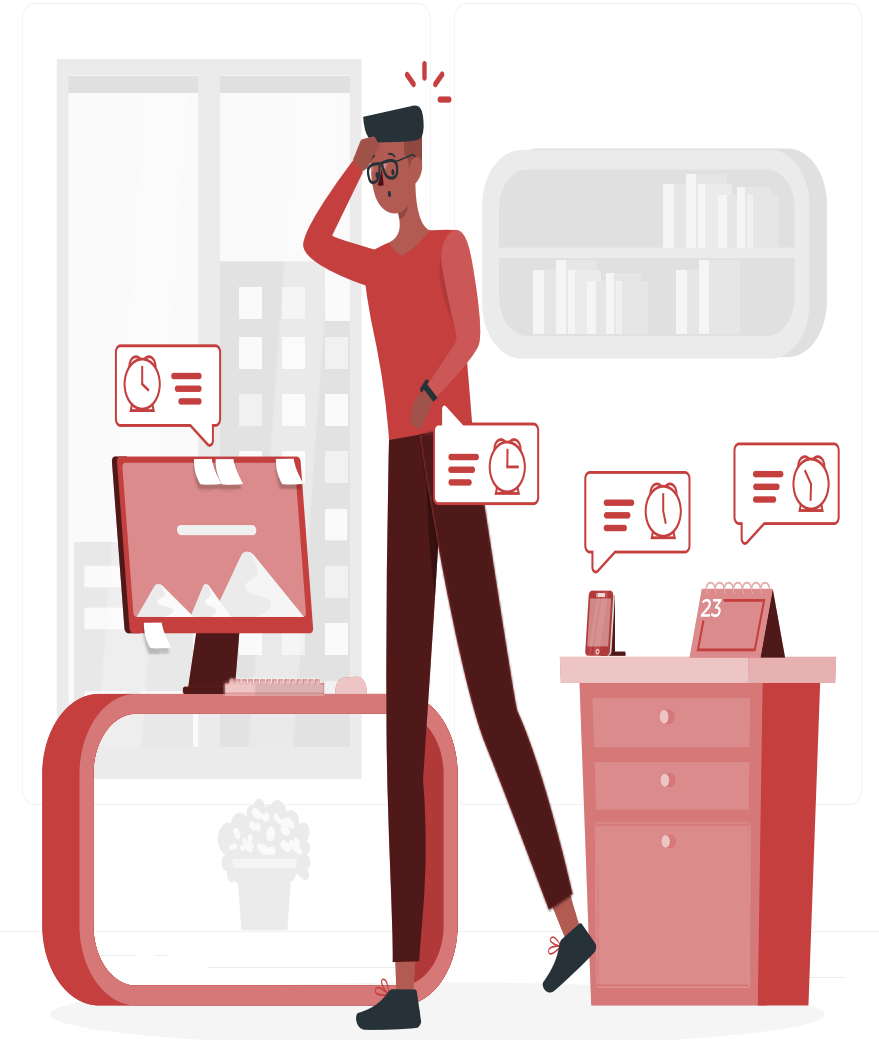
00 Recordatorio

00

# Recordatorio

Desarrollo  
Asistido por  
IA

Conceptos  
generales de  
IA

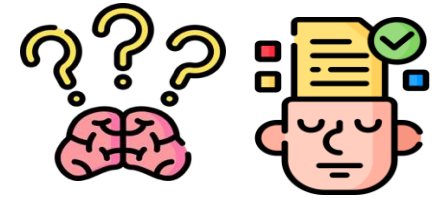


01 Contexto del código

01

# No me cunde el día

Contexto del código



La productividad media de un desarrollador es de **10-50** líneas por día

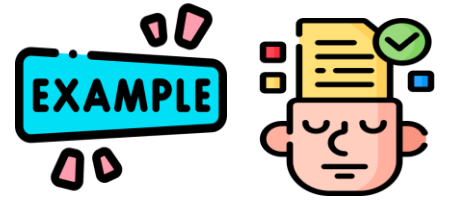
- Escribir código es el 10%
- Leer, probar y debugar es el 90%

---

**Paradoja: “Los desarrolladores NO tienen tiempo de trabajar rápido”**

# El problema de la “X”

Contexto del código



## ¿Qué significa este código?

```
if x < 10 {  
    ++X  
}
```

### Problemas:

- El naming de una variable como X es malo
- No se indica qué es X
- No se sabe que hacemos con X ante la condición
- No hay documentación
- ...

### Investigación

- Entender el antes y el después del código
- Leer todo el fichero
- Entender el nombre del fichero
- ...

# ¿Qué es el contexto del código en IA?

## Contexto del código

**Conjunto** de **elementos** del **entorno de desarrollo** que un **asistente de código IA** (como Github Copilot) **analiza** para **entender** lo que estas programando y así **generar sugerencias** más útiles y precisas

Proporciona:

- *Cuanto más contexto, mejor hace las sugerencias*
- *El contexto incluye muchos factores*
- *Aprendizaje del historial de desarrollador e incluso del equipo*

**Objetivo:** Mejorar las propuestas realizadas en base a la realidad

**Ejemplo:** Es como si el asistente fuera un programador que está leyendo todo el proyecto y no solo una línea. Su ayuda será mejor / más inteligente si llega a entender todo el panorama completo



# Aspectos que se tienen en cuenta

## Contexto del código



### Línea de código actual

Lo que se está escribiendo en tiempo real

### Contenido del archivo

El código completo del archivo actual: comentarios, variables, funciones, clases, ...

### Archivos relacionados

Otros archivos del proyecto abiertos o importados (uso de una función definida en otro módulo)

### Estructura del proyecto

Información sobre el lenguaje, framework, dependencias, arquitectura general, etc.

### Comentarios y Documentación

Comentarios implementados por el desarrollador o docstring explicativos

### Paquetes y librerías instaladas

Se considerarán todas las dependencias

### Estilo de codificación

El estilo del código ya escrito se utiliza para aprender y replicarlo (convenciones locales, patrones de uso, nombre de variables, etc.)

### Historial de Edición y patrones previos

Durante la sesión de uso se puede considerar todo lo escrito

# Comprender el Contexto del Código

Contexto del código



## Beneficios

Ofrece sugerencias de valor

Evita repetir código

Identifica posibles errores

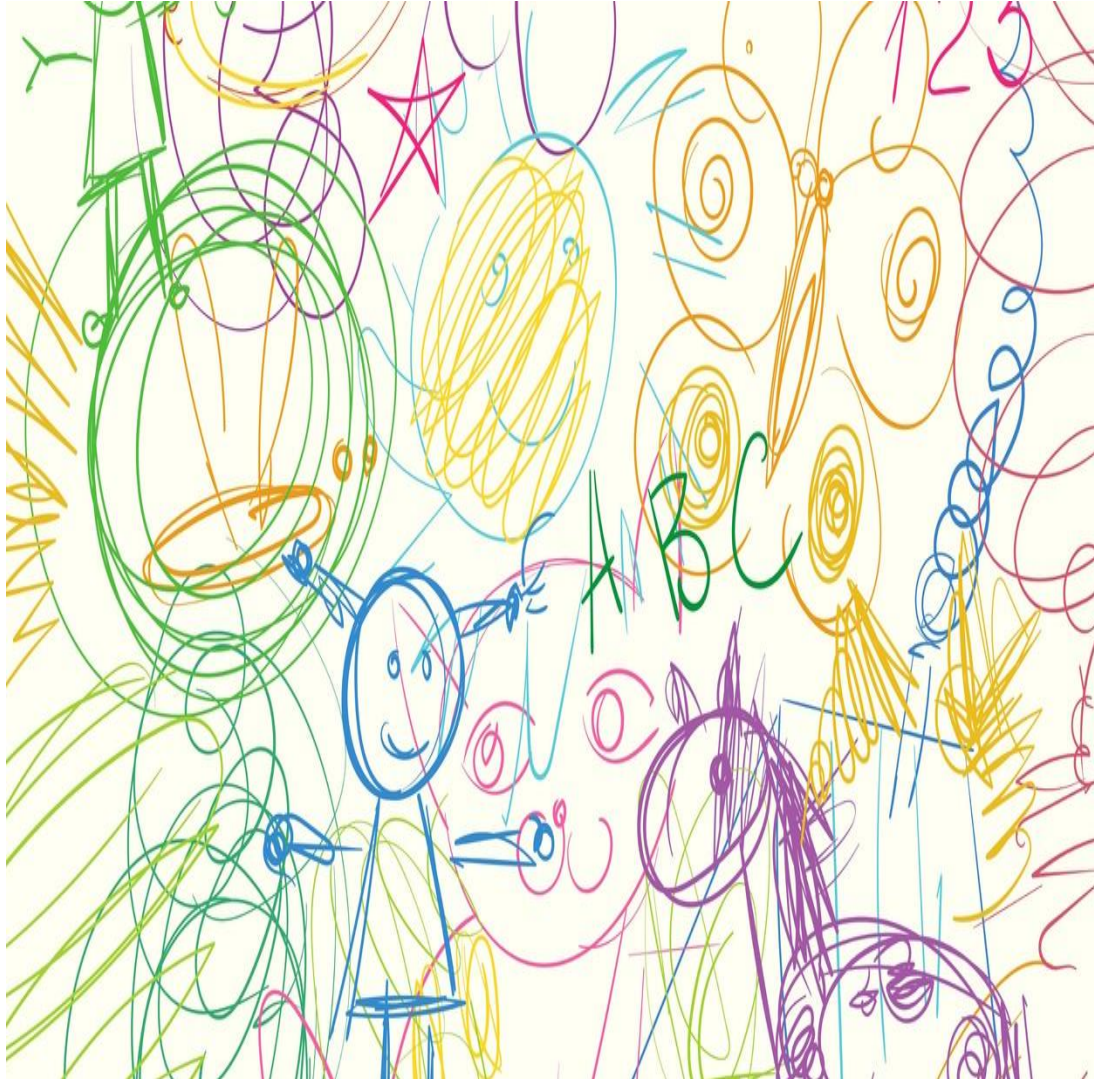
Mejora colaboración humano-IA

Mejora la adherencia a un estilo y/o estructura

**Ejemplo:** Es como pedirle ayuda a un asistente sin contexto es como pedirle a alguien que complete una frase que no ha escuchado desde el principio

# El problema de las imágenes y los diagramas

## Contexto del Código



La **mayoría** de las **herramientas de IA** actuales **presenta problemas** a la hora de **interpretar** directamente **imágenes, diagramas o esquemas visuales**

Esto se debe principalmente a varios motivos

- *Las herramientas de IA han sido entrenadas principalmente con texto (código, comentarios, documentación)*
- *La mayoría de ellas no tienen capacidades de visión computacional integrada (como los modelos multimodales)*
- *Falta comprensión en el significado semántico visual de un diagrama sin ayuda adicional*
  - *Problemas con interfaces de dibujo, diagramas UML, etc.*

**Ejemplo:** Es como si alguien es muy bueno leyendo libros, pero completamente ciego. Si no lo escribes, no lo va a entender

Solución: Convertir estos elementos a texto estructurado o código

# ¿Qué es DaaC (Diagram as a Code)?

## Contexto del código

**Práctica** que consiste en crear **diagramas mediante código** en lugar de dibujarlos manualmente. Para ello hace uso de lenguajes y/o herramientas específicas

Término que se utiliza para almacenar el código fuente de una imagen de diagrama como archivo de texto

Proporciona:

- *Diagramas sean versionables, reproducibles y mantenibles como parte del proyecto*
- *Todo esta escrito con texto*
- *Automatizable para mantener la documentación viva y actualizada*
- *Solicitar ayuda a una IA para modificaciones*
- *Reutilizable*

**Objetivo:** Integrar los diagramas en el flujo de trabajo del desarrollo de SW, de forma automática, mantenible y colaborativa, igual que el código fuente

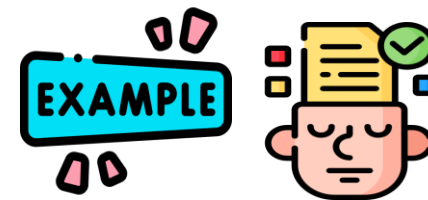
**Ejemplo:** Es como programar un PowerPoint, en lugar de dibujar las formas, se definen en código y estos se dibujan solas

Ejemplo Mermaid, PlantUML, Diagrams, etc.



# Ejemplo de DaaC

## Contexto del código



diagrams.mingrammer.com

Diagrams Docs Guides Nodes GitHub Sponsoring

```
from diagrams import Cluster, Diagram
from diagrams.aws.compute import ECS, EKS, Lambda
from diagrams.aws.database import Redshift
from diagrams.aws.integration import SQS
from diagrams.aws.storage import S3

with Diagram("Event Processing", show=False):
    source = EKS("k8s source")

    with Cluster("Event Flows"):
        with Cluster("Event Workers"):
            workers = [ECS("worker1"),
                       ECS("worker2"),
                       ECS("worker3")]

        queue = SQS("event queue")

        with Cluster("Processing"):
            handlers = [Lambda("proc1"),
                       Lambda("proc2"),
                       Lambda("proc3")]

    store = S3("events store")
    dw = Redshift("analytics")

    source >> workers >> queue >> handlers
    handlers >> store
    handlers >> dw
```

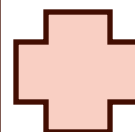
A diagram illustrating an Event Processing architecture. It shows a flow from a 'k8s source' (Kubernetes icon) to a cluster of 'Event Workers' (ECS icons labeled worker1, worker2, worker3). These workers connect to an 'event queue' (SQS icon). The queue then feeds into a 'Processing' cluster containing three Lambda functions (proc1, proc2, proc3). The outputs of these functions are directed to two destinations: 'events store' (S3 icon) and 'analytics' (Redshift icon). The entire central part is labeled 'Event Processing'.

# El contexto lo es todo

Contexto del código



Contexto Técnico



Contexto Empresarial

# Hay un límite en el contexto

Contexto del código



## 02 Ingeniería del Contexto (Context Engineering)

---

02

# El diablo se encuentra en los detalles

Ingeniería del contexto (Context Engineering)



# Contexto de un LLM

## Ingeniería del contexto (Context Engineering)

Se dice que un **bot conversacional "recuerda"** el **contexto**, pero lo que hace es **meter como prompt del último input todo lo que ha generado en respuestas anteriores**

- Implementado como "Prompt Recursivo Automático"
- Se usa sobre todo en los de tipo ChatBot (IAs de Diálogo)

**Objetivo:** Ser la memoria del LLM en las conversaciones

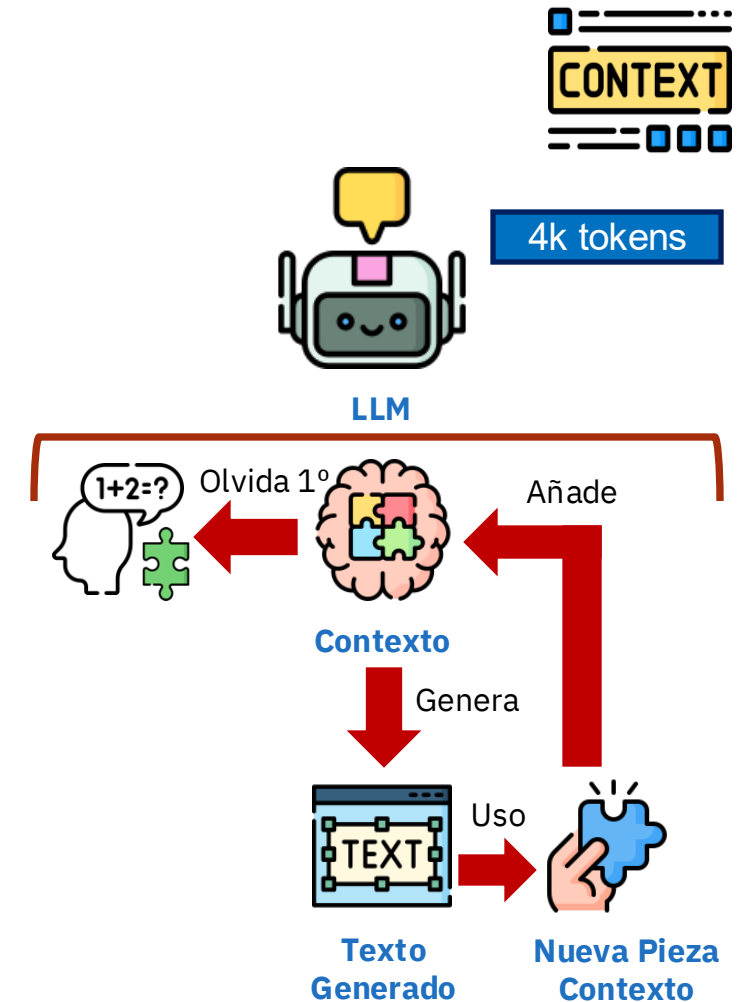
Uso de los **tokens** para su **funcionamiento**

Cada **modelo guarda** una **cantidad de texto limitada** y **diferente entre ellos**

- Se pretende resolver en un futuro
- **Ejemplos**
  - GPT-4 tiene dos modelos : 8K y 32k
  - ChatGPT con Davinci, son 4K tokens (unas 3K palabras en inglés)
  - 2K – 3K palabras NO son suficientes para mantener el contexto para escribir una novela

### Detalles

- Si la **conversación** es **muy larga** entonces **NO recuerda** que es lo **primero** que dijo
- **Mayor cantidad** de **texto "in-memory"** -> **Mayor cantidad** de **tiempo** manteniendo la **coherencia**
- La **primera interacción** suele tener una **longitud mayor** al incluir el **contexto**
- "Toda conversación anterior no cabe en el prompt y tampoco es eficiente incluirla toda cada vez"



### Problema de Olvidos



- No recuerda bien el contexto
- Retroalimentación con el prompt anterior
- Límites diferentes por modelo

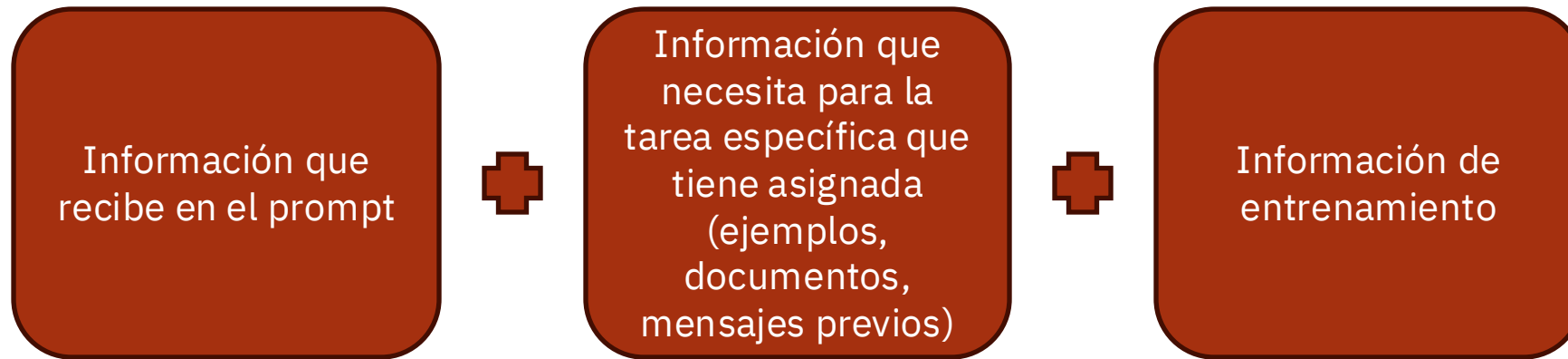
# ¿Qué es la ventana de contexto?

Ingeniería del contexto (Context Engineering)



**Límite** de **cuántas palabras** o **tokens** (unidades mínimas de texto) **pueden procesar un modelo** a la vez en una ejecución

Se considera la máxima cantidad de texto (input + output)



# ¿Qué es la ingeniería del contexto?

Ingeniería del contexto (Context Engineering)



**Conjunto** de **técnicas** y **estrategias** utilizadas para **adaptar sistemas, productos** o **servicios** a las **condiciones específicas** del **entorno** en el que serán utilizados, con el objetivo de mejorar su efectividad, personalización o relevancia

Se encarga de **proporcionar** dinámicamente **toda** la **información** y los **antecedentes** necesarios para que la IA puede responder

Diseñar y optimizar esa información de apoyo: decidir qué incluir, cómo formatearlo y cómo entregarlo al modelo

Características:

- *Orientación al usuario (adaptación al contexto del usuario: momento, dispositivo, ubicación, etc.)*
- *Multidisciplinar*
- *Dinámica (se puede basar en datos que pueden cambiar)*
- *Soporte con IA*
- *Requiere dedicar tiempo*

**Objetivo:** Capacidad de seleccionar y organizar la información exacta para llenar la ventana de contexto con información relevante para el resultado

**Ejemplo:** Es como un camarero que recuerda qué te gusta, si tienes alergias, cuánto tiempo llevas esperando, la temperatura ambiente ... y adapta el servicio en función de todo esto

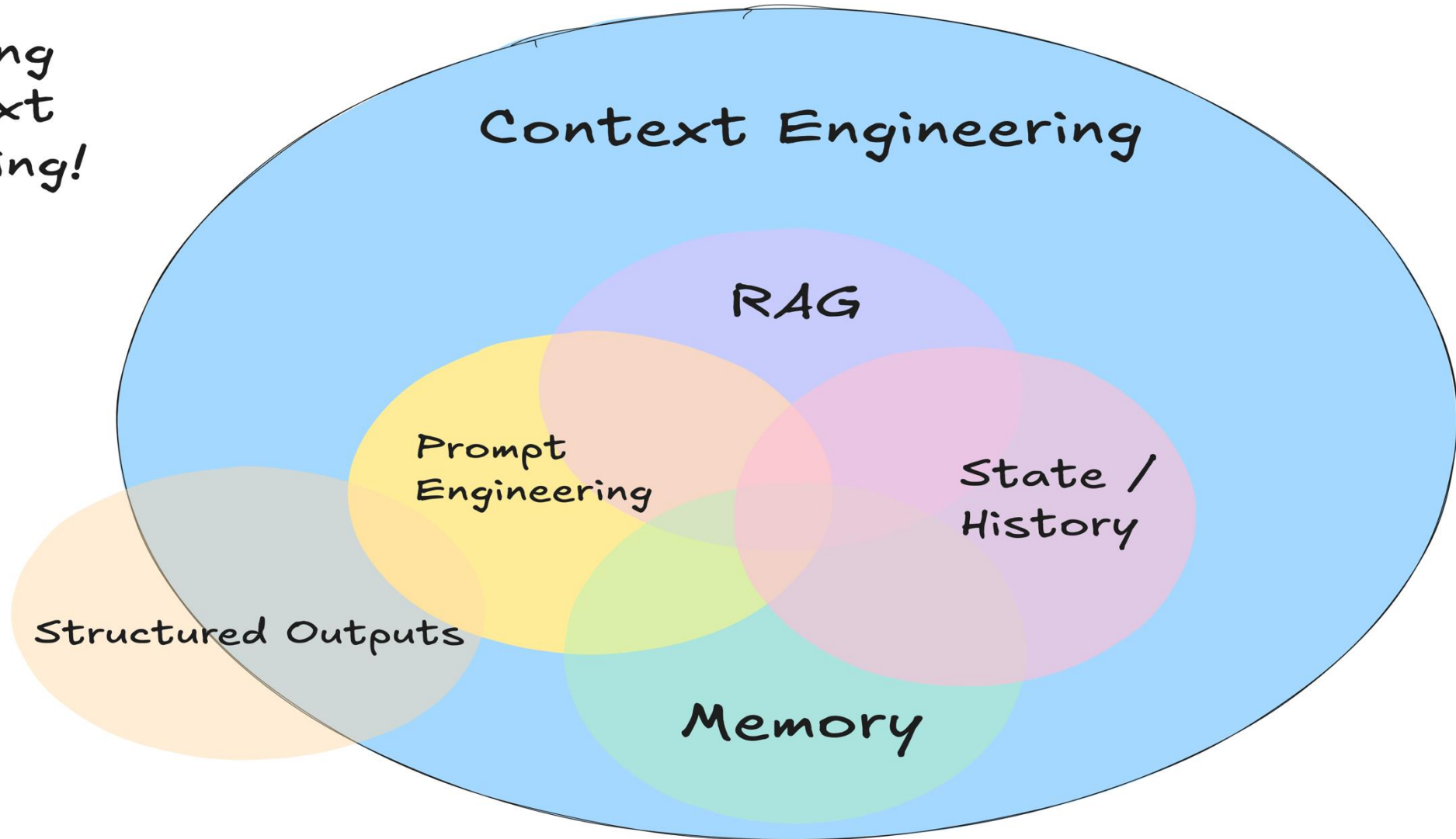
Garantiza que la IA sea precisa y realmente útil

# ¿Qué es la ingeniería del contexto?

Ingeniería del contexto (Context Engineering)



Everything  
is Context  
Engineering!



# Aspectos que se tienen en cuenta

Ingeniería del contexto (Context Engineering)



Instrucciones  
usuario  
(prompts)

Ejemplos  
(Few-shot)

Instrucciones  
sistemas

Datos  
relevantes

Recuperación  
de información  
(RAG)

Salidas de  
aplicación o  
APIs

Estado e  
historial

Compactación  
de información

Metadatos del  
entorno

# Capas de la Ingeniería del Contexto

## Ingeniería del contexto (Context Engineering)



### Ingeniería de prompts

- *Capa táctica / Tactical Layer*
- *Se basa en el diseño de templates y cue para guiar la salida del modelo*
- *Incluye: definición de roles, estrategias de cadena de pensamiento, formato de respuestas, etc.*

### Instrucciones y sistemas

- *Capa operativa/ Operational Layer*
- *Incluye: tono, la experiencia en el dominio, tolerancia al riesgo, etc.*

### Retrieval-Augmented Generation (RAG)

- *Capa informacional / Informational Layer*
- *Obtiene conocimiento en tiempo real o estructurado en el momento de la inferencia*
- *Con la ingeniería del contexto se garantiza que esta información, se seleccione, se resuma y se inyecte*

### Gestión de Memoria

- *Capa temporal / Temporal Layer*
- *Simular una memoria a largo plazo mediante bases de datos, embeddings y logs de sesión*
- *Qué recordar, cómo recordarlo y cuándo olvidarlo*

### Modelado del entorno y la entidad

- *Capa semántica / Semantic Layer*
- *Incluye: intención del usuario, el comportamiento previo, los niveles de acceso*
- *Uso en sistemas multiusuario o agentes de IA*

### Flujos de trabajo

- *Capa arquitectura / Architectural Layer*
- *El contexto debe de fluir de forma sencilla entre tareas, modelos y herramientas*

# Impactos en la Industria

Ingeniería del contexto (Context Engineering)



Mejora de la  
eficiencia

Reducir  
alucinaciones

Colaboración  
entre humanos  
e IA

Seguridad

Límite de  
tokens

Latencia

Privacidad

Debugging

Estandarización

knowmad mood



Spain · Portugal · Italy · United Kingdom · United States · Uruguay. Morocco

Email: [example@knowmadmood.com](mailto:example@knowmadmood.com)  
[www.knowmadmood.com](http://www.knowmadmood.com)