



Universidad
Rey Juan Carlos



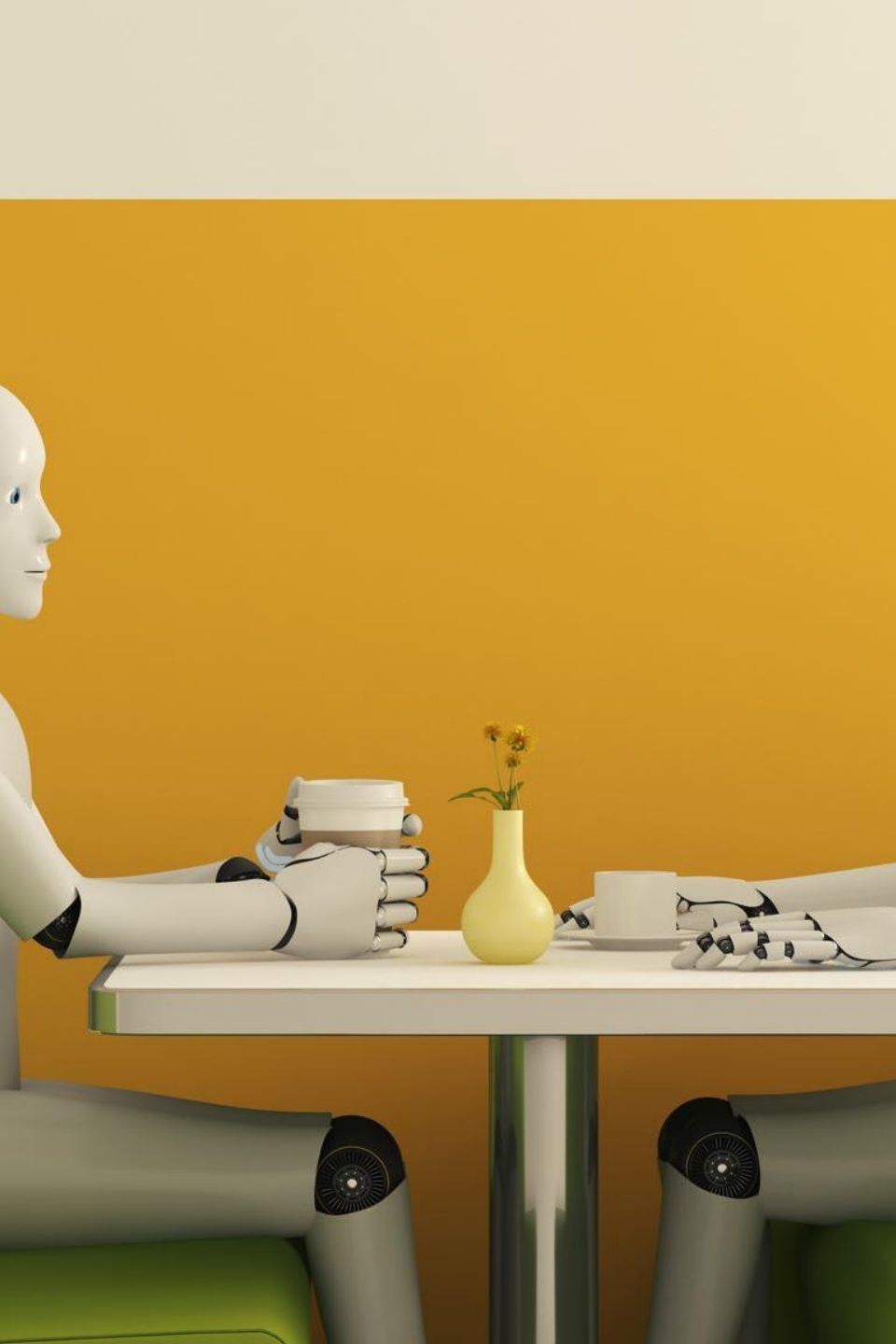
Planificación y Sistemas Cognitivos

Arquitecturas cognitivas. Modelos del Lenguaje

FRANCISCO JOSÉ ROMERO RAMÍREZ
FRANCISCO.ROMERO@URJC.ES

Índice de contenidos

1. Introducción
2. Large Language Models
3. Agentes LLMs
4. Arquitecturas cognitivas para agentes LLMs
5. Futuro



1. Introducción

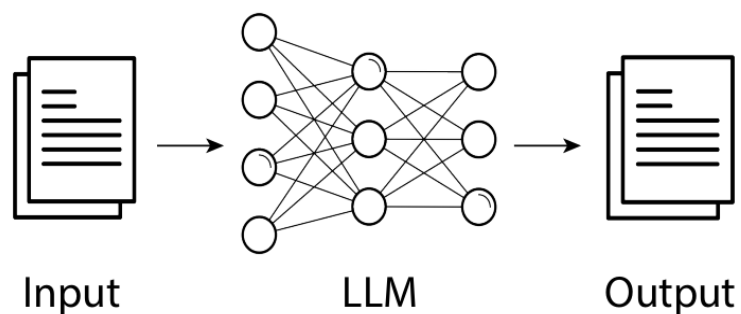
1. Introducción

- **Cognición y lenguaje** se encuentran estrechamente entrelazados.
- El lenguaje como herramienta cognitiva:
 - Permite **organizar experiencias** y **representar información** abstracta mediante la formación de conceptos.
 - **Facilita el pensamiento** más allá de lo sensorial inmediato, apoyando el razonamiento abstracto y la resolución de problemas.
 - El lenguaje modula nuestros procesos de **codificación, almacenamiento y recuperación de información**.
 - Desarrollo de las **habilidades sociales**.

1. Introducción

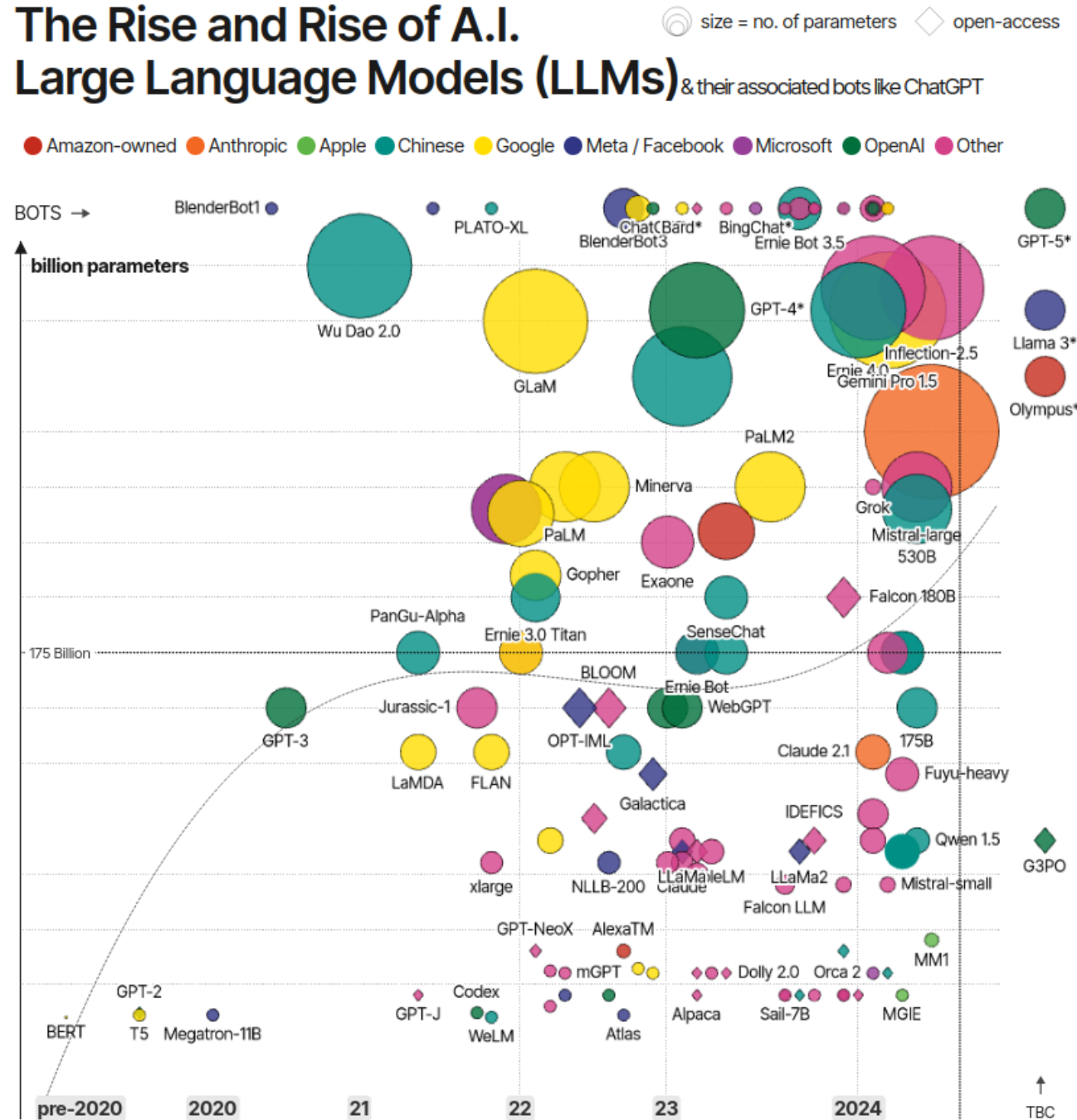
- **Representaciones simbólicas clásicas**
 - Lógica y reglas de producción
 - Redes semánticas y frames
- **Transición conexionista**
 - Redes neuronales simples
 - Modelos recurrentes (RNN, LSTM)
- **Incorporación de la probabilidad**
 - Modelos estadísticos de lenguaje. N-gramas, Modelos Ocultos de Markov (HMM)
 - Redes Bayesianas y Modelos Gráficos Probabilísticos (CRFs)
- **Hacia las redes de atención y Transformers**
 - Atención y auto-atención (Vaswani et al., 2017)

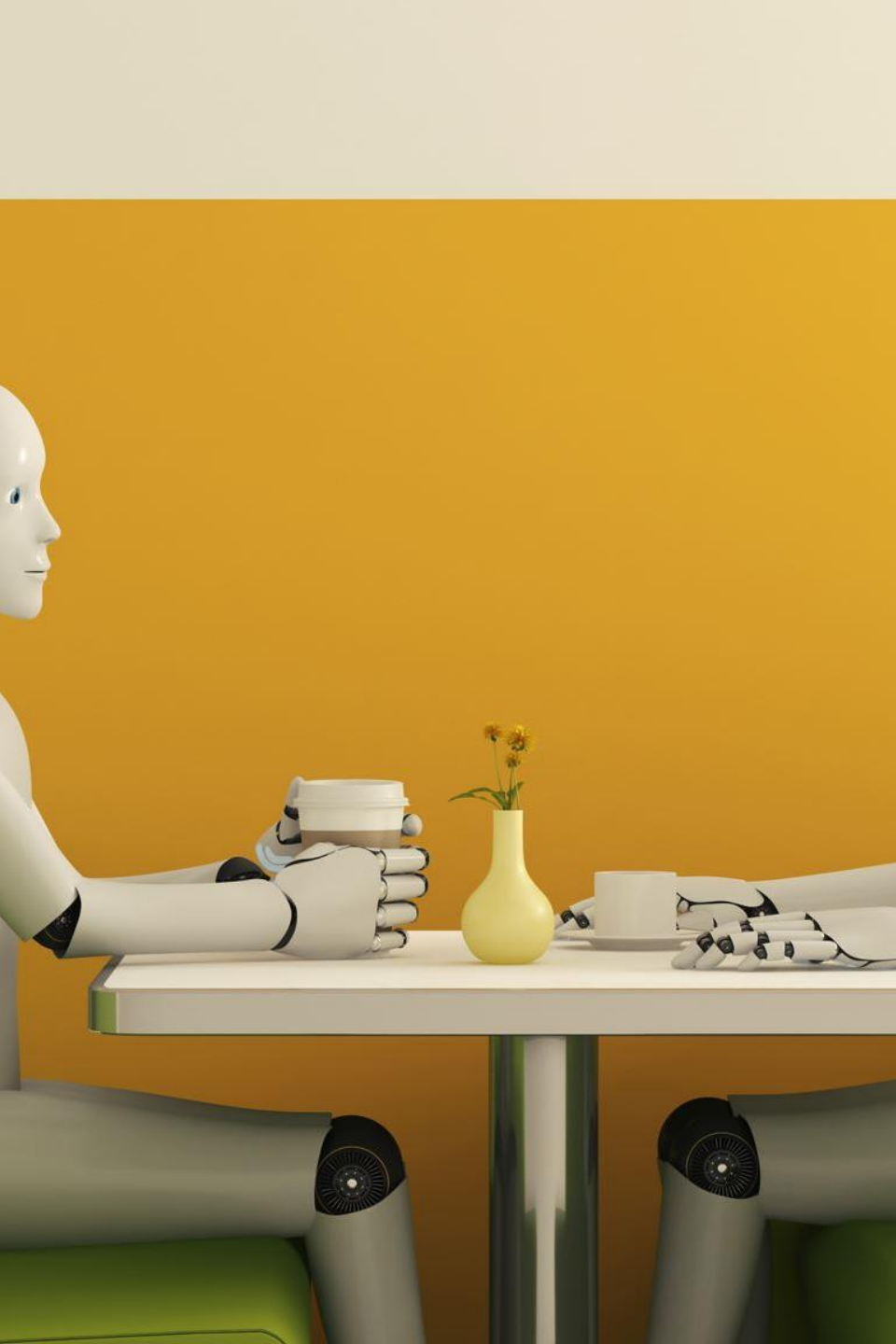
1. Introducción



Natural Language Processing (NLP)

<https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/>

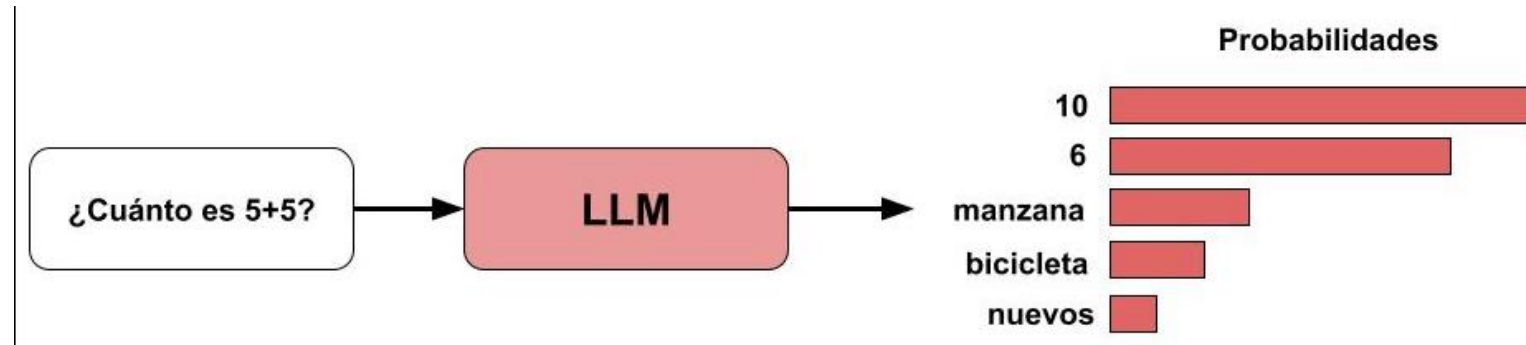




2. Large Language Models

2. LLMs

- LLMs son "transformer models" entrenados para predecir la siguiente palabra (token)



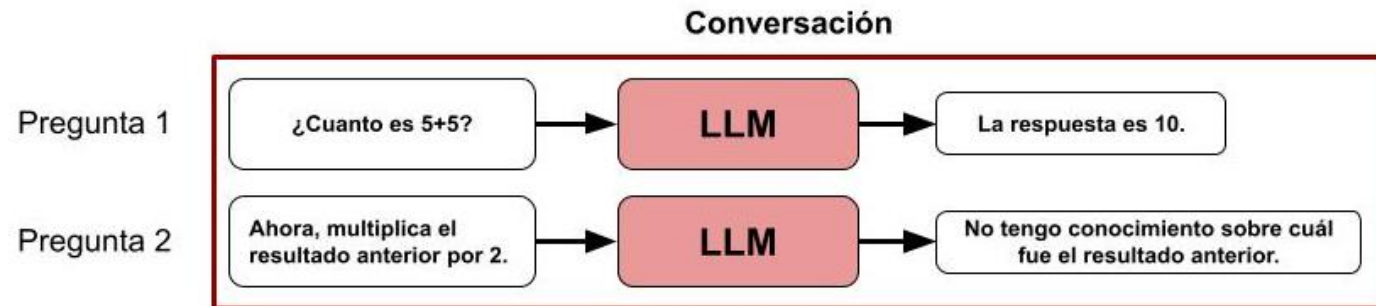
- Entrenado con millones de sentencias, como pueden ser todos los textos de internet.
- Podemos pensar, que entrenar LLMs es como enseñar a un loro para que imite el lenguaje humano.

2. LLMs

- Si muestreamos múltiples tokens consecutivos, es posible imitar conversaciones y usar el LLM para generar respuestas más detalladas.

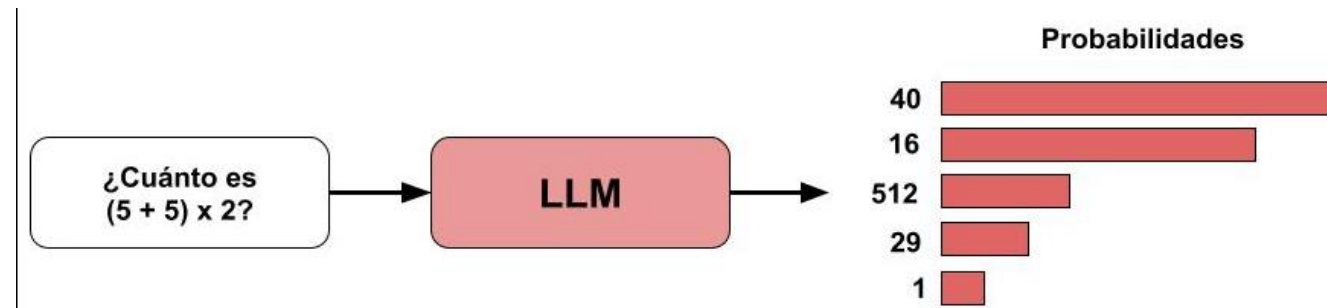


- Sin embargo, si continuamos la conversación, cualquier LLM mostrará su principal limitación. No recuerda conversaciones.



2. LLMs

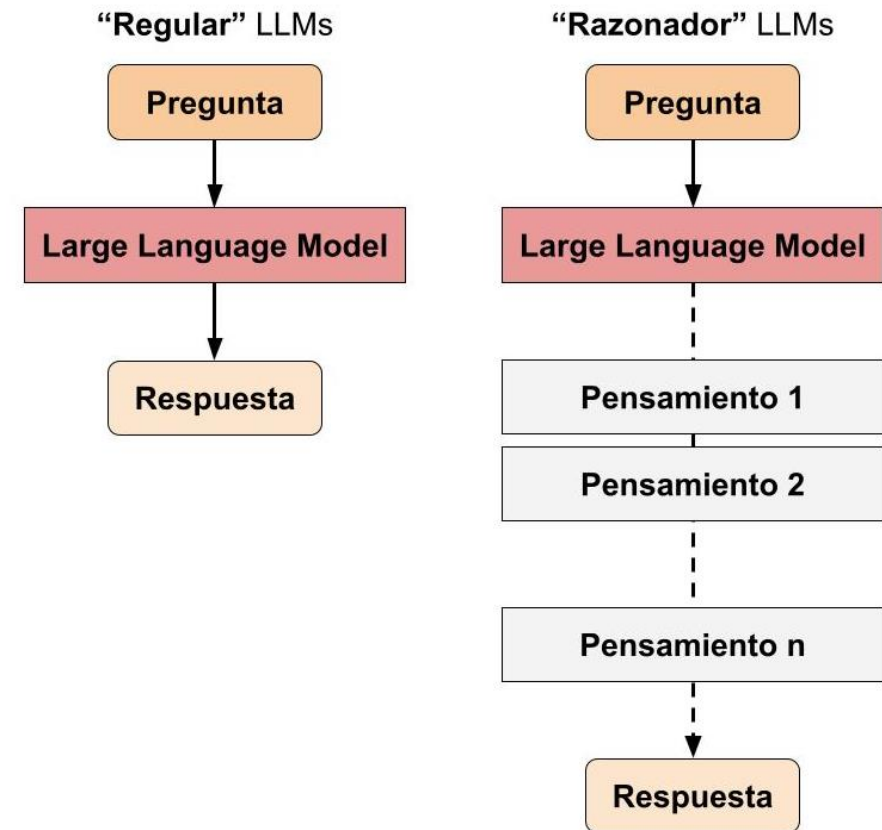
- Hay muchas otras tareas en las que los LLMs suelen fallar, incluidas matemáticas básicas como la multiplicación y la división.



- Para resolver estas limitaciones, las herramientas basadas en LLMs han incorporado mecanismos para realizar un "**proceso de razonamiento**", antes de dar la respuesta. DeepSeek-R1, OpenAI o3-mini, and Google Gemini 2.0 Flash Thinking son ejemplos de ello.

2. LLMs

- Los LLM que incorporan mecanismos de razonamiento tienden a dividir un problema en pasos más pequeños (a menudo denominados "pasos de razonamiento" o "procesos de pensamiento") antes de responder a una pregunta determinada.



2. LLMs

- Ejemplo razonamiento:
- Prompt: ¿Cuál es resultado de $8*9*2*3/1.34$?

Primero multiplicamos los números:

- $728 \times 9 = 72$
- $72 \times 2 = 144$
- $144 \times 3 = 432$

Luego dividimos el resultado por 1.34:

- $432/1.34 \approx 322.39$

Por lo tanto, el resultado aproximado es **322.39**.

Ejemplo ChatGPT o3-mini

2. LLMs

- Ling et al 2017 en DeepMind fueron pioneros en utilizar el razonamiento del lenguaje natural para resolver problemas matemáticos «... derivar la respuesta final a través de una serie de pequeños pasos». Entrenaron un modelo secuencia a secuencia desde cero.

Problem 1:

Question: Two trains running in opposite directions cross a man standing on the platform in 27 seconds and 17 seconds respectively and they cross each other in 23 seconds. The ratio of their speeds is:

Options: A) $3/7$ B) $3/2$ C) $3/88$ D) $3/8$ E) $2/2$

Rationale: Let the speeds of the two trains be x m/sec and y m/sec respectively. Then, length of the first train = $27x$ meters, and length of the second train = $17y$ meters. $(27x + 17y) / (x + y) = 23 \rightarrow 27x + 17y = 23x + 23y \rightarrow 4x = 6y \rightarrow x/y = 3/2$.

Correct Option: B



Ling et al. [Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems](#). ACL 2017

2. LLMs

- Siguiendo el trabajo de Ling et al 2017, Cobbe et al 2021 en OpenAI construyeron un conjunto de datos de problemas matemáticos de palabras mucho mayor (GSM8K) con razonamientos en lenguaje natural, y lo utilizaron para hacer finetune de GPT3.

Problem: Ali is a dean of a private school where he teaches one class. John is also a dean of a public school. John has two classes in his school. Each class has $\frac{1}{8}$ the capacity of Ali's class which has the capacity of 120 students. What is the combined capacity of both schools?

Solution: Ali's class has a capacity of 120 students. Each of John's classes has a capacity of $120/8 = 15$ students. The total capacity of John's two classes is $15 \text{ students} * 2 \text{ classes} = 30 \text{ students}$. The combined capacity of the two schools is $120 \text{ students} + 30 \text{ students} = 150 \text{ students}$.

Final answer: 150



Cobbe et al. Training Verifiers to Solve Math Word Problems. [arXiv:2110.14168](https://arxiv.org/abs/2110.14168) [cs.LG]. 2021

2. LLMs

- Show Your Work: Scratchpads for Intermediate Computation with Language Models

Input:

2 9 + 5 7

Target:

<scratch>

2 9 + 5 7 , C: 0

2 + 5 , 6 C: 1 # added 9 + 7 = 6 carry 1

, 8 6 C: 0 # added 2 + 5 + 1 = 8 carry 0

0 8 6

</scratch>

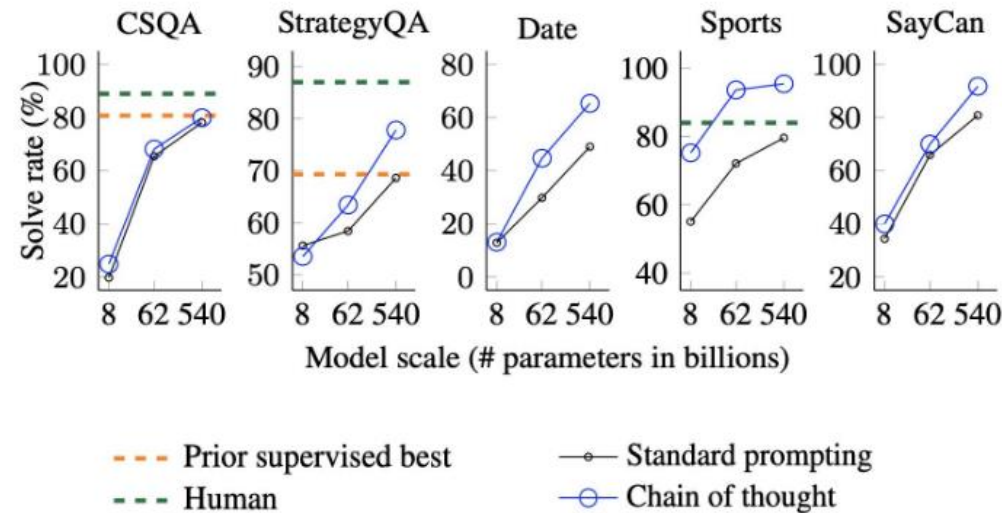
8 6



Nye et al. Show Your Work: Scratchpads for Intermediate Computation with Language Models. [arXiv:2112.00114](https://arxiv.org/abs/2112.00114) [cs.LG], 2021

2. LLMs

- Chain-of-Thought (CoT) Prompting



- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. NeurIPS 2022

2. LLMs

- Training with intermediate steps (Ling et al 2017)
- Finetuning with intermediate steps (Cobbe et al 2021, Nye et al 2021)
- Prompting with intermediate steps (Nye et al. 2021, Wei et al 2022)
- Los **pasos intermedios** es lo que realmente importa!!

2. LLMs

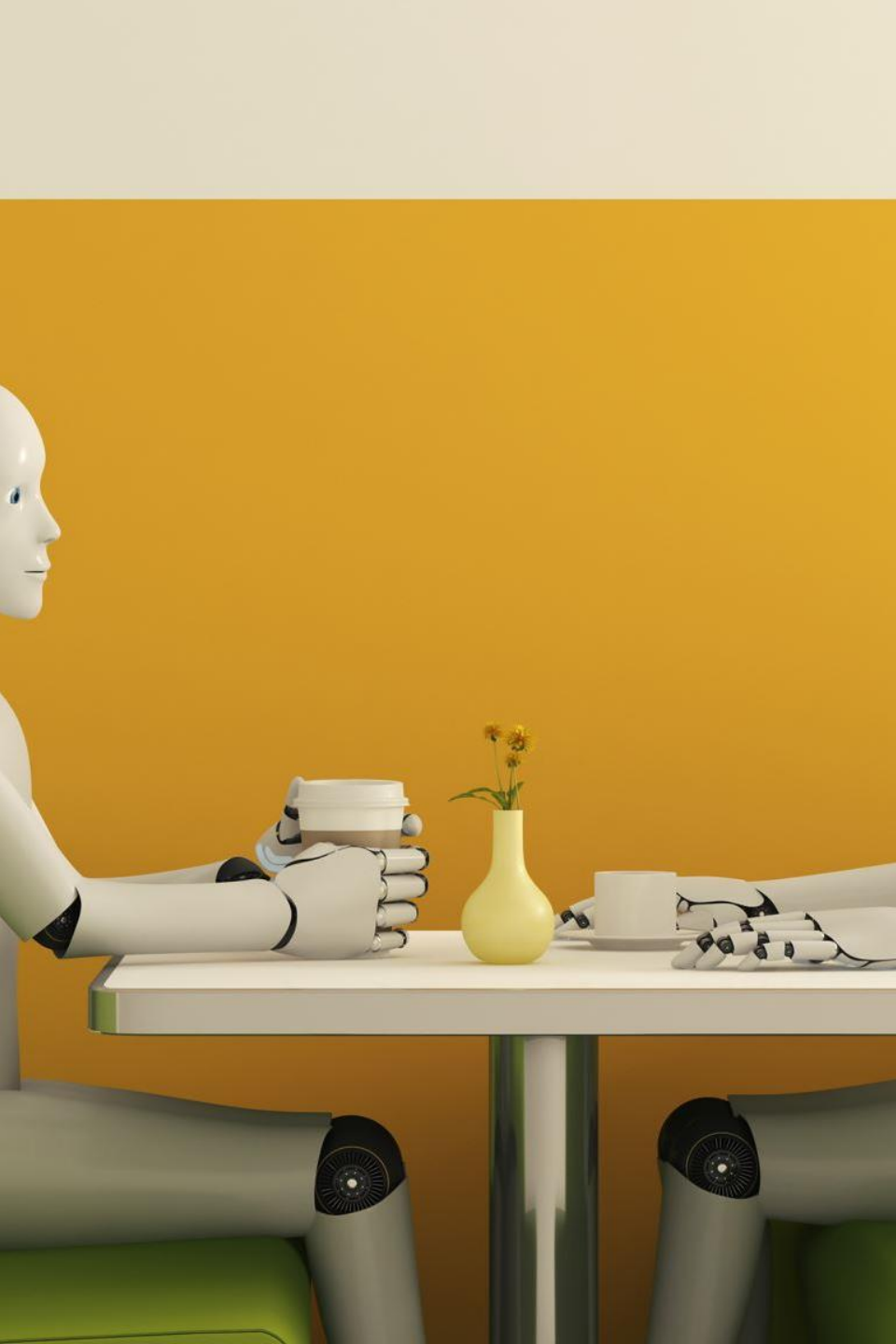
- La evolución de los modelos de razonamiento en IA generativa está pasando de una simple predicción autoregresiva de tokens a enfoques **híbridos neuro-simbólicos**, incorporando:
 - **Razonamiento basado en grafos** (ej., gráficos de conocimiento + LLMs para la toma de decisiones estructurada).
 - **Razonamiento aumentado con herramientas** (ej., LLMs que invocan APIs, ejecutan código o consultan bases de datos).
 - **Aprendizaje por refuerzo a partir de retroalimentación humana (RLHF)** para mejorar la coherencia lógica.

2. LLMs

- **Chain-of-Thought (CoT) Prompting** - El CoT fomenta que los modelos generen pasos intermedios antes de llegar a una conclusión. Este método mejora la capacidad de los modelos para resolver tareas complejas estructurando el proceso de resolución de problemas ([\[2201.11903\] Chain-of-Thought Prompting Elicits Reasoning in Large Language Models](#)).
- **Tree-of-Thought (ToT) Prompting** - Expandiendo el CoT, el ToT permite a los modelos explorar múltiples rutas de razonamiento en una estructura de árbol. Esto facilita la revisión de estrategias y mejora la flexibilidad para resolver problemas. ([\[2305.10601\] Tree of Thoughts: Deliberate Problem Solving with Large Language Models](#)).
- **Self-Consistency Decoding** - Este método implica generar varias rutas de razonamiento de manera independiente y seleccionar la respuesta más consistente. Ayuda a reducir sesgos y aumenta la fiabilidad de las respuestas del modelo. ([\[2203.11171\] Self-Consistency Improves Chain of Thought Reasoning in Language Models](#)).

2. LLMs

- **Toolformer: Language Models Can Teach Themselves to Use Tools** (*Schick et al., 2023*) – Muestra cómo los LLMs pueden aprender a usar herramientas externas para mejorar su capacidad de razonamiento ([\[2302.04761\] Toolformer: Language Models Can Teach Themselves to Use Tools](#)).
- **Reflexion: Language Agents with Verbal Reinforcement Learning** (*Shinn et al., 2023*) – Introduce el marco de Reflexion, que permite a los modelos mejorar sus respuestas mediante el aprendizaje por refuerzo verbal ([\[2303.11366\] Reflexion: Language Agents with Verbal Reinforcement Learning](#)).



3. Agentes

3. Agentes

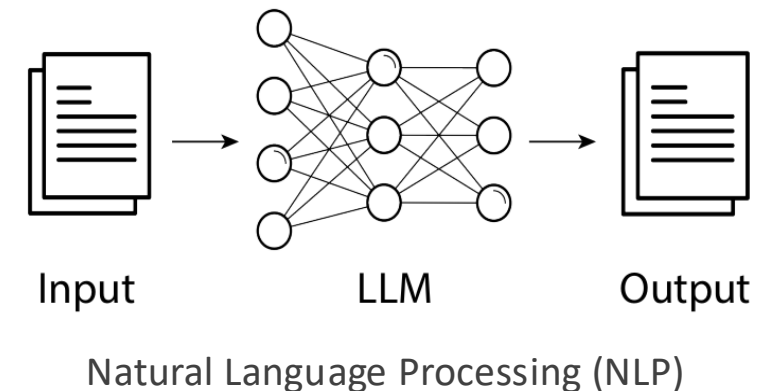
- Los **Large Language Models (LLMs)** por sí mismos no pueden realizar tareas que requieran una toma de decisiones compleja ni ejecutar acciones del mundo real.

Yo estudie robótica en la URJC... ... *y fue una experiencia increíble que realmente amplió mis horizontes.*

El clima ayer en Fuenlabrada fue... ... *Lo siento, no tengo acceso a información en tiempo real como el clima. Te recomendaría buscar en un sitio web de pronóstico del tiempo o en una aplicación meteorológica para obtener esa información específica sobre Fuenlabrada ayer.*

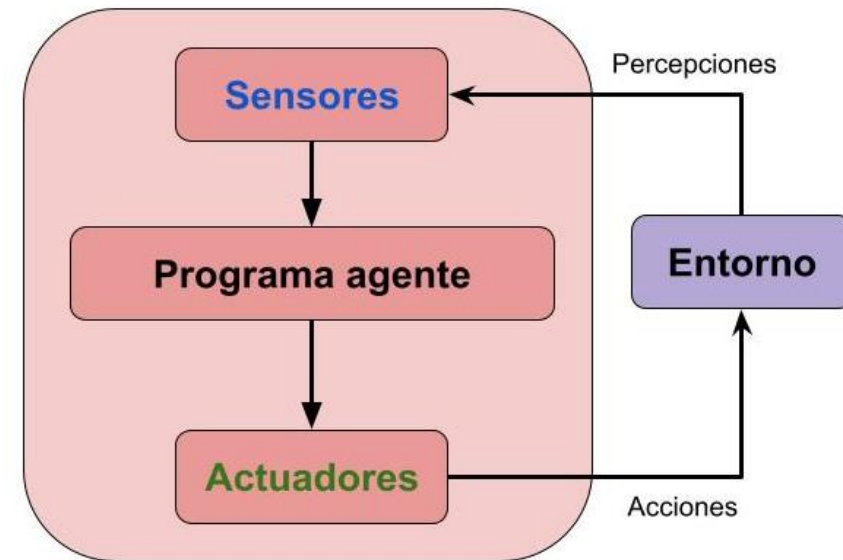
El mundo es...

- **Complejo.** Muchas reglas que manejar.
- **Estocástico.** Las reglas disponibles pueden fallar.



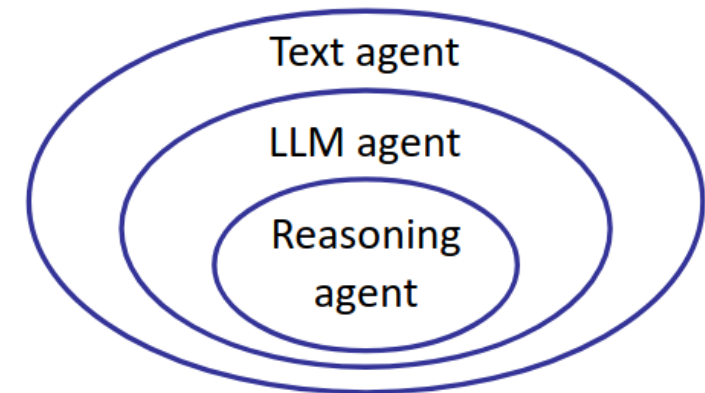
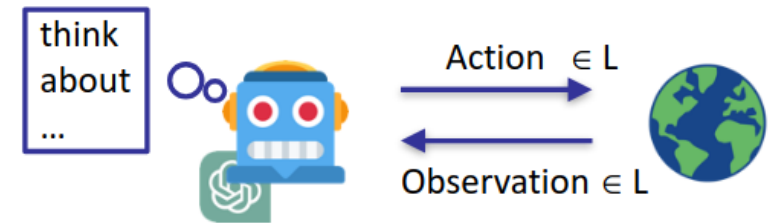
3. ¿Qué es un agente?

- Un **agente inteligente** es una entidad capaz de **percibir su entorno, procesar tales percepciones y responder o actuar en dicho entorno de manera racional**, es decir, logrando objetivos, tendiendo a maximizar un resultado esperado y adquiriendo conocimiento con su desempeño. Es capaz de percibir su medio ambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores (elementos que reaccionan a un estímulo realizando una acción). (*Wikipedia*)

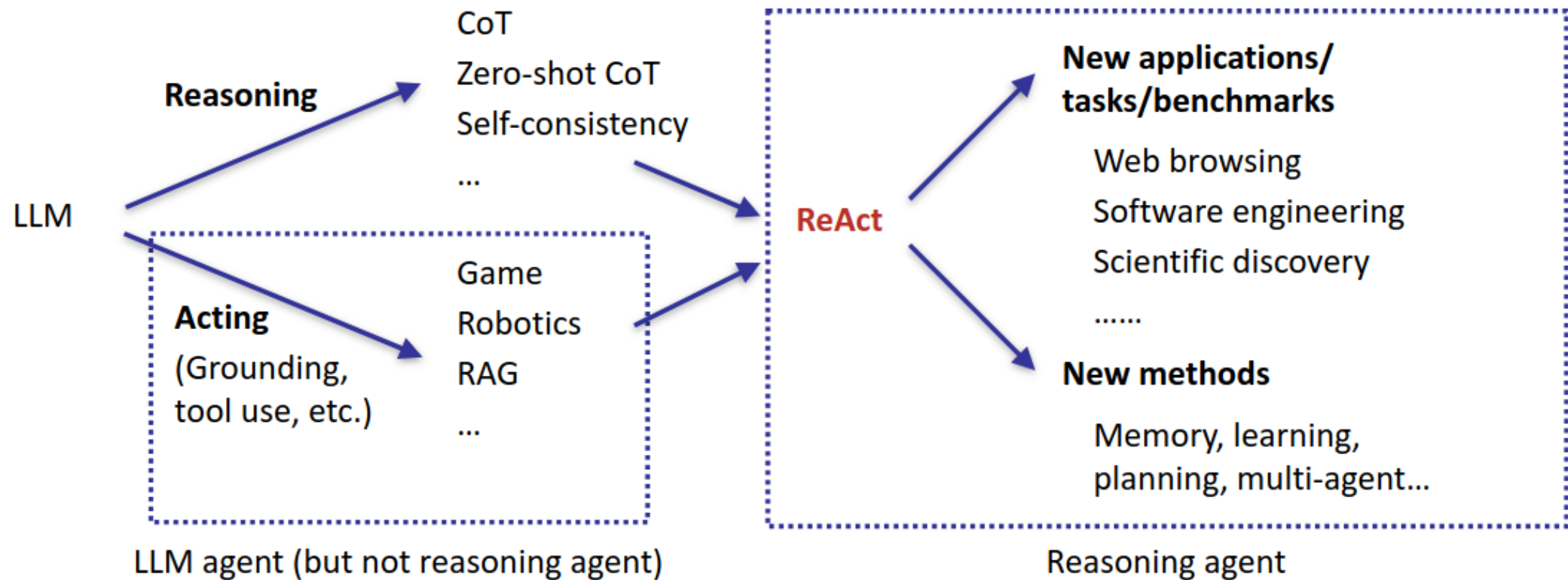


3. ¿Qué es un agente LLM?

- Nivel 1: Agente de texto. Utiliza la acción de texto y la observación
 - Ejemplos: ELIZA, LSTM-DQN
- Nivel 2: Agente LLM. Utiliza LLM para actuar.
 - Ejemplos: SayCan, Language Planner
- Nivel 3: Agente de razonamiento. Utiliza LLM para razonar para actuar.
 - Ejemplos: ReAct, AutoGPT



3. ¿Qué es un agente LLM?

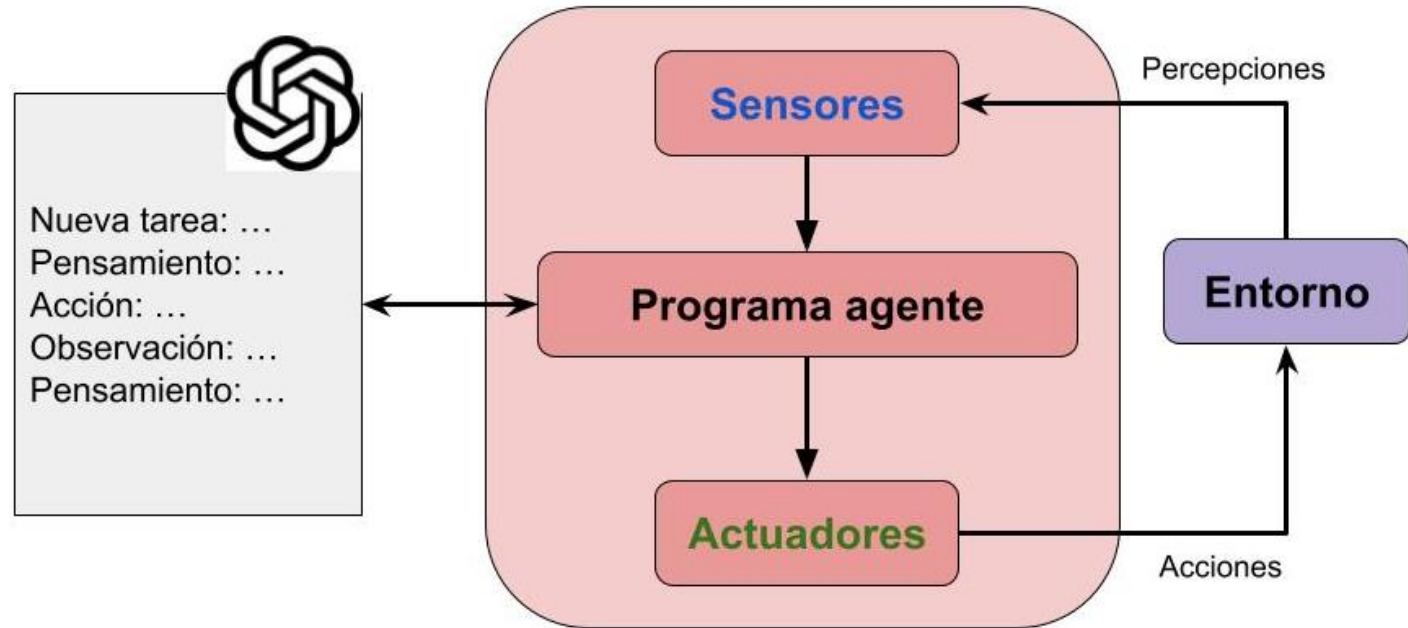


3. Language Agents

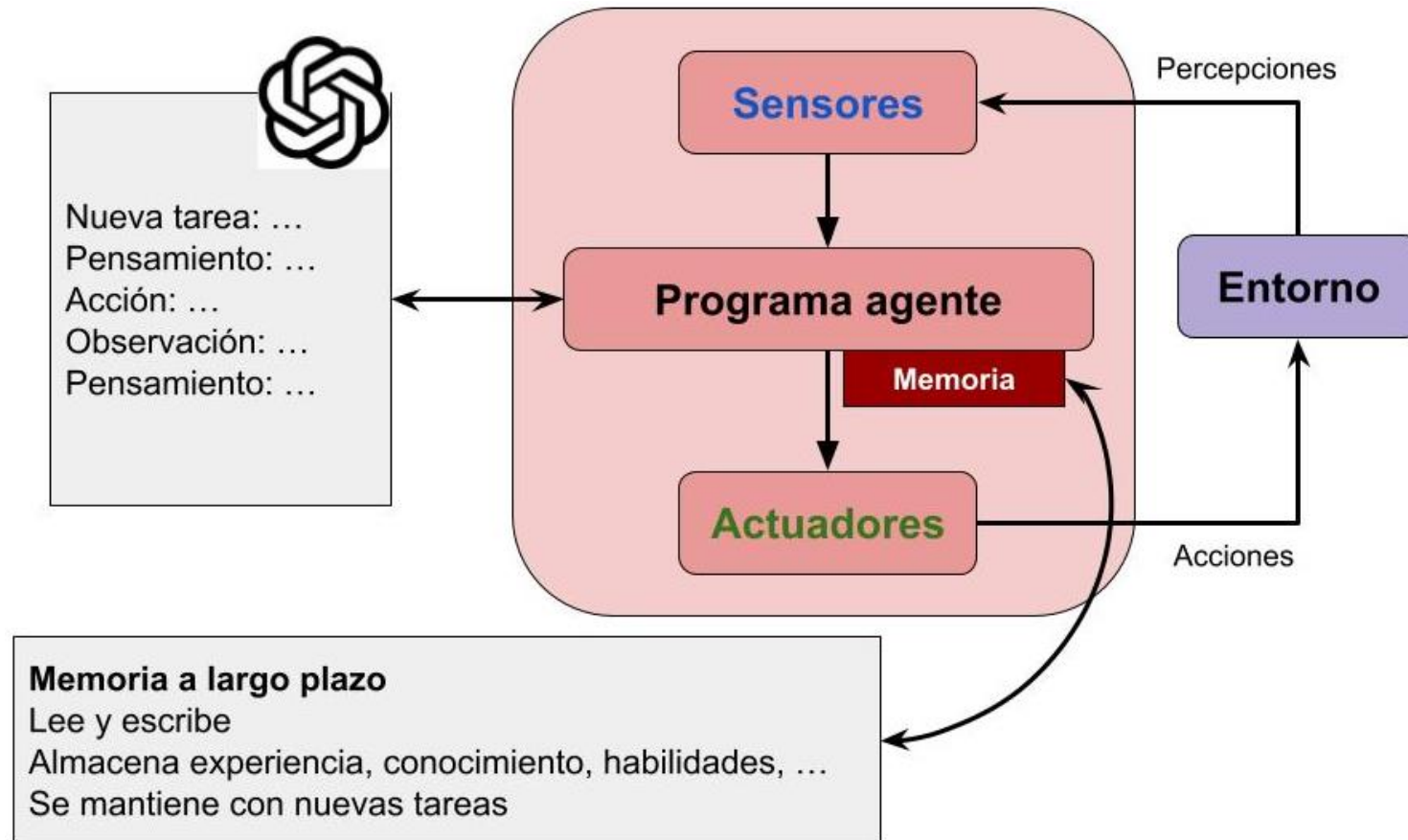


"Memoria a corto plazo"

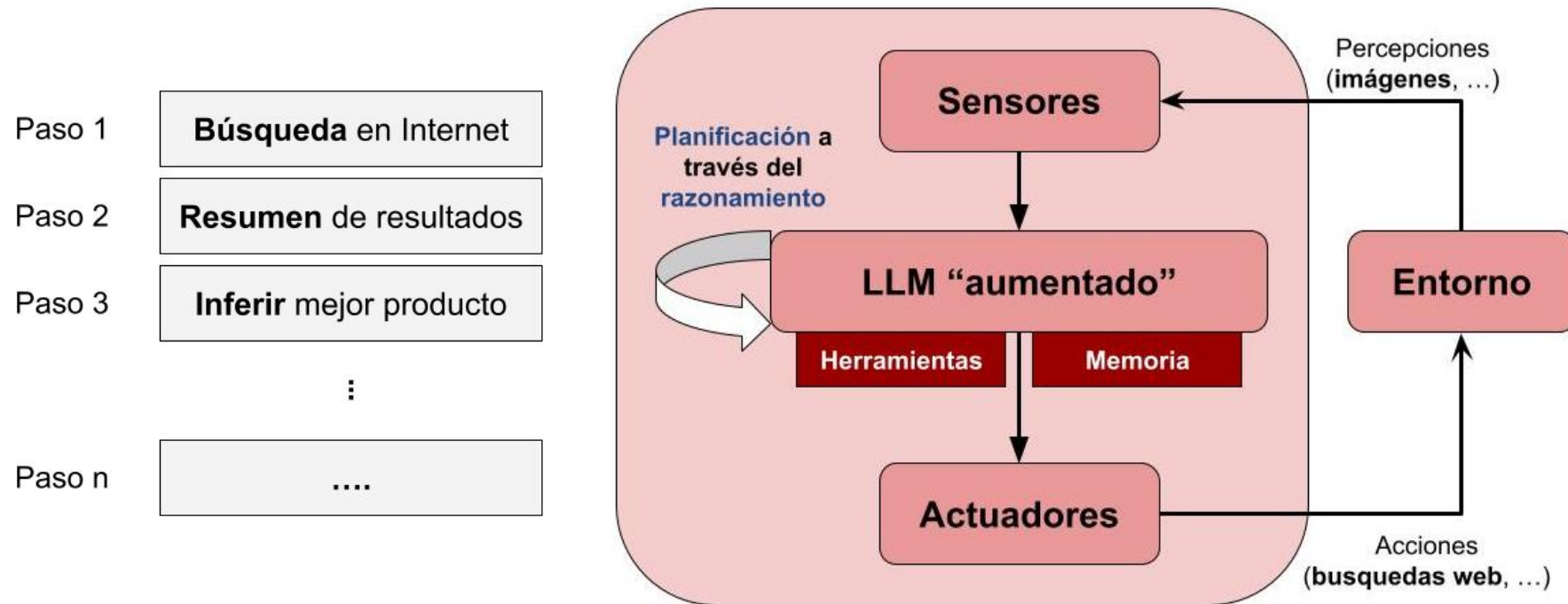
- Contexto limitado
- Atención limitada.
- No se mantiene con nuevas tareas



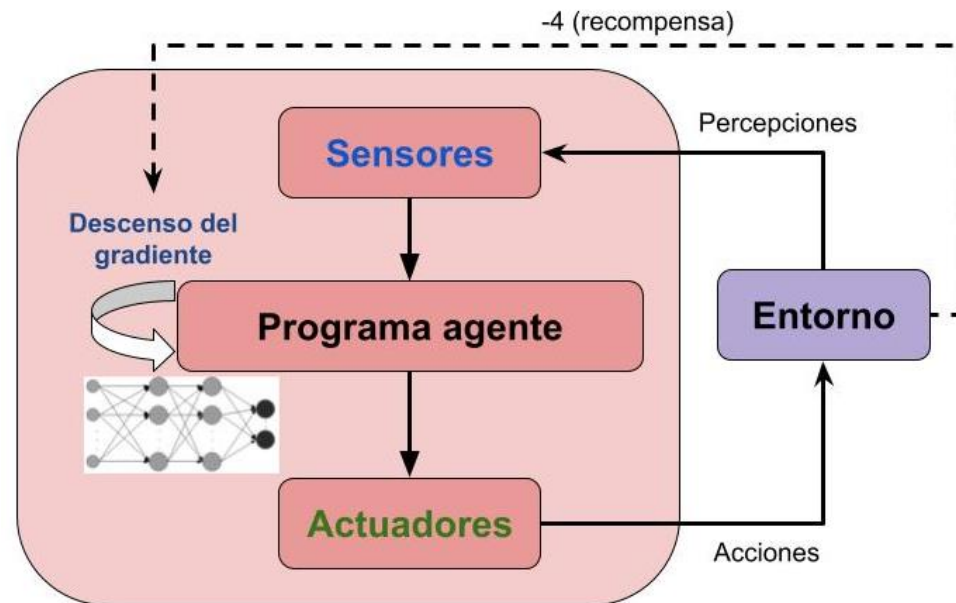
3. Language Agents



3. Language Agents

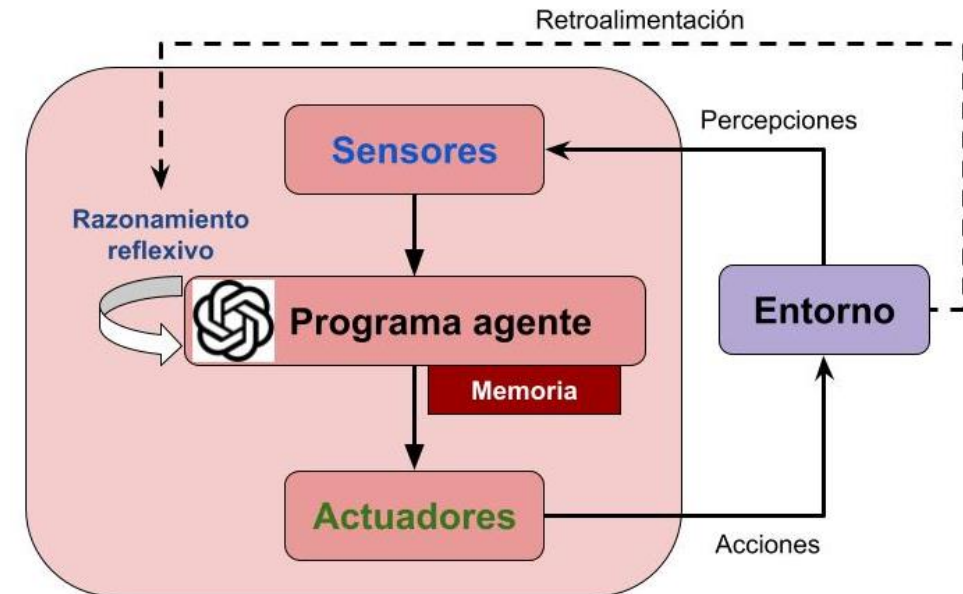


3. Language Agents



RL tradicional

- Aprendizaje (recompensa **escalar**)
- Aprendizaje mediante actualización de **pesos**

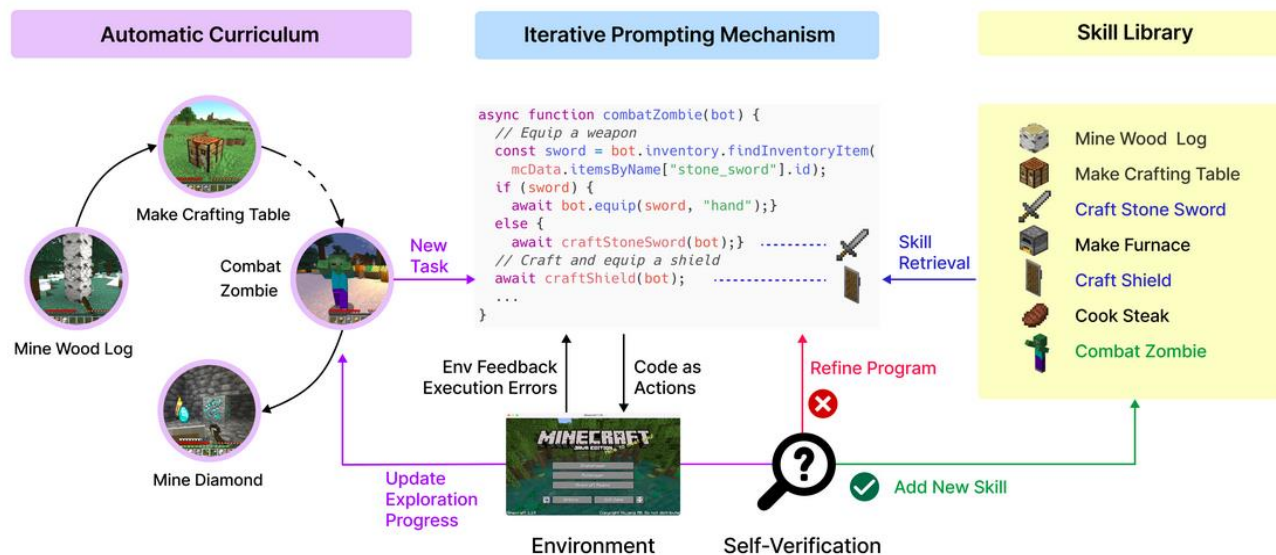


Razonamiento: RL "verbal"

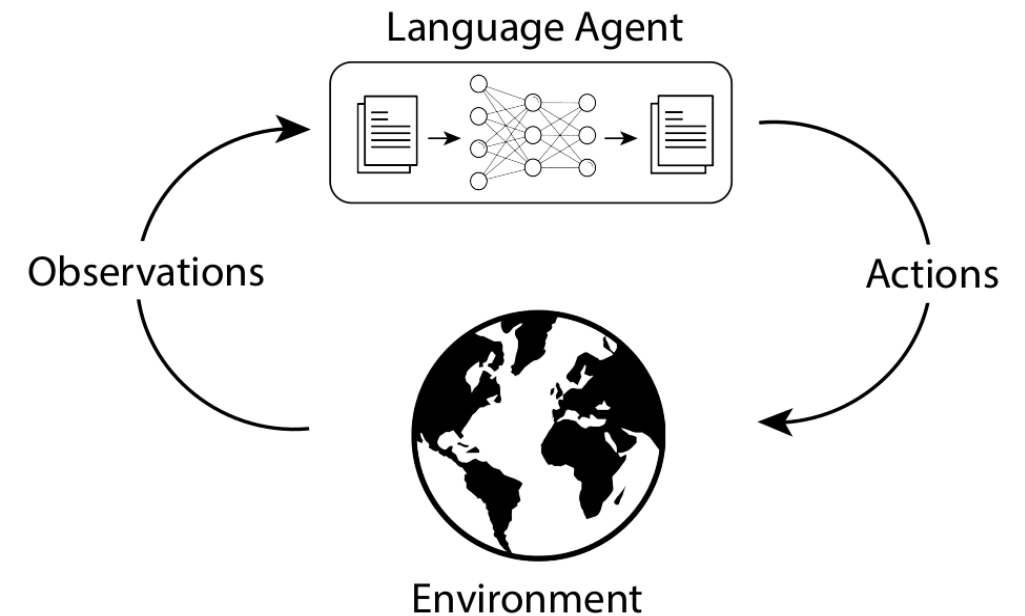
- Aprendizaje (retroalimentación **texto**)
- Aprendizaje mediante actualización del **lenguaje** (memoria a largo plazo)

3. Language Agents

Los **Language Agents** colocan al LLM en un bucle de retroalimentación directa con el entorno externo transformando las observaciones en texto y utilizando el LLM para elegir acciones.

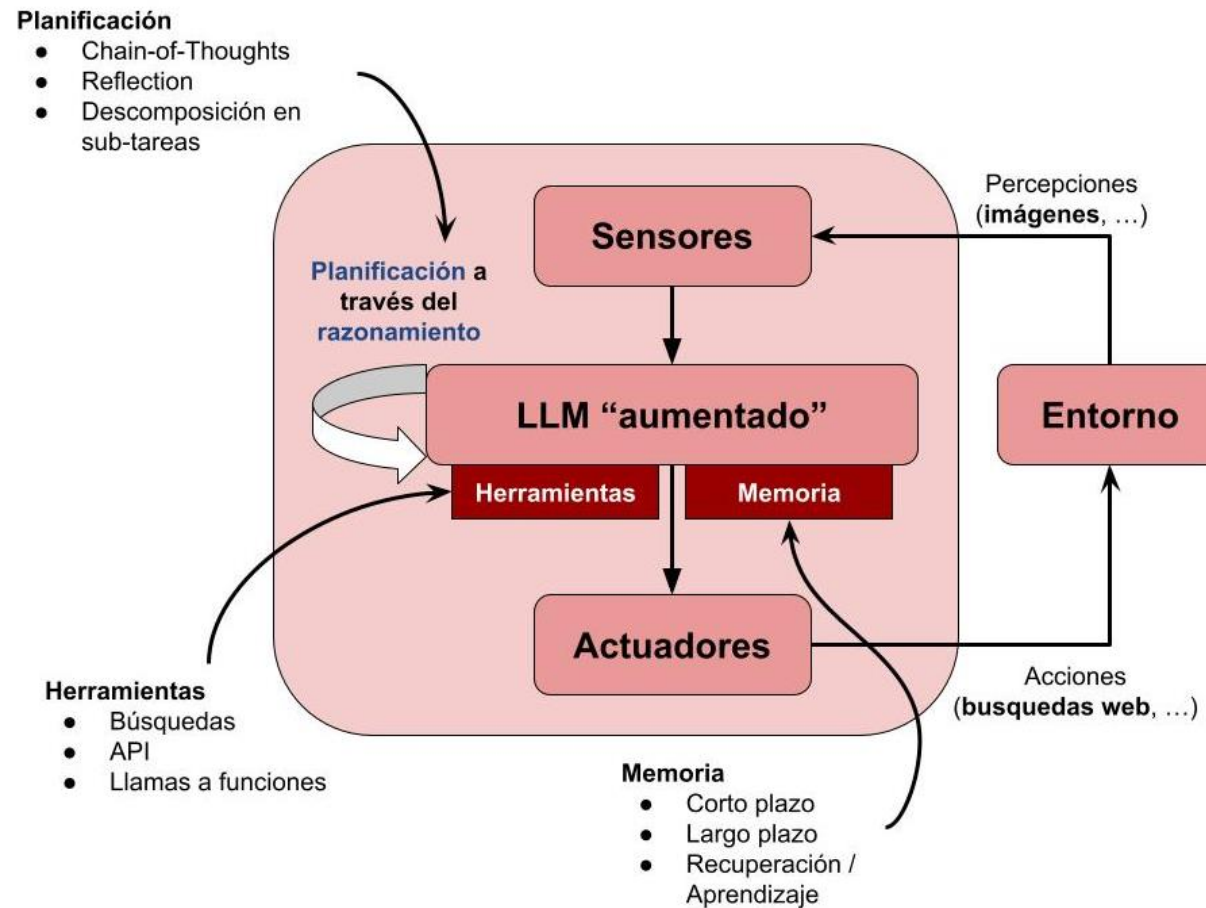


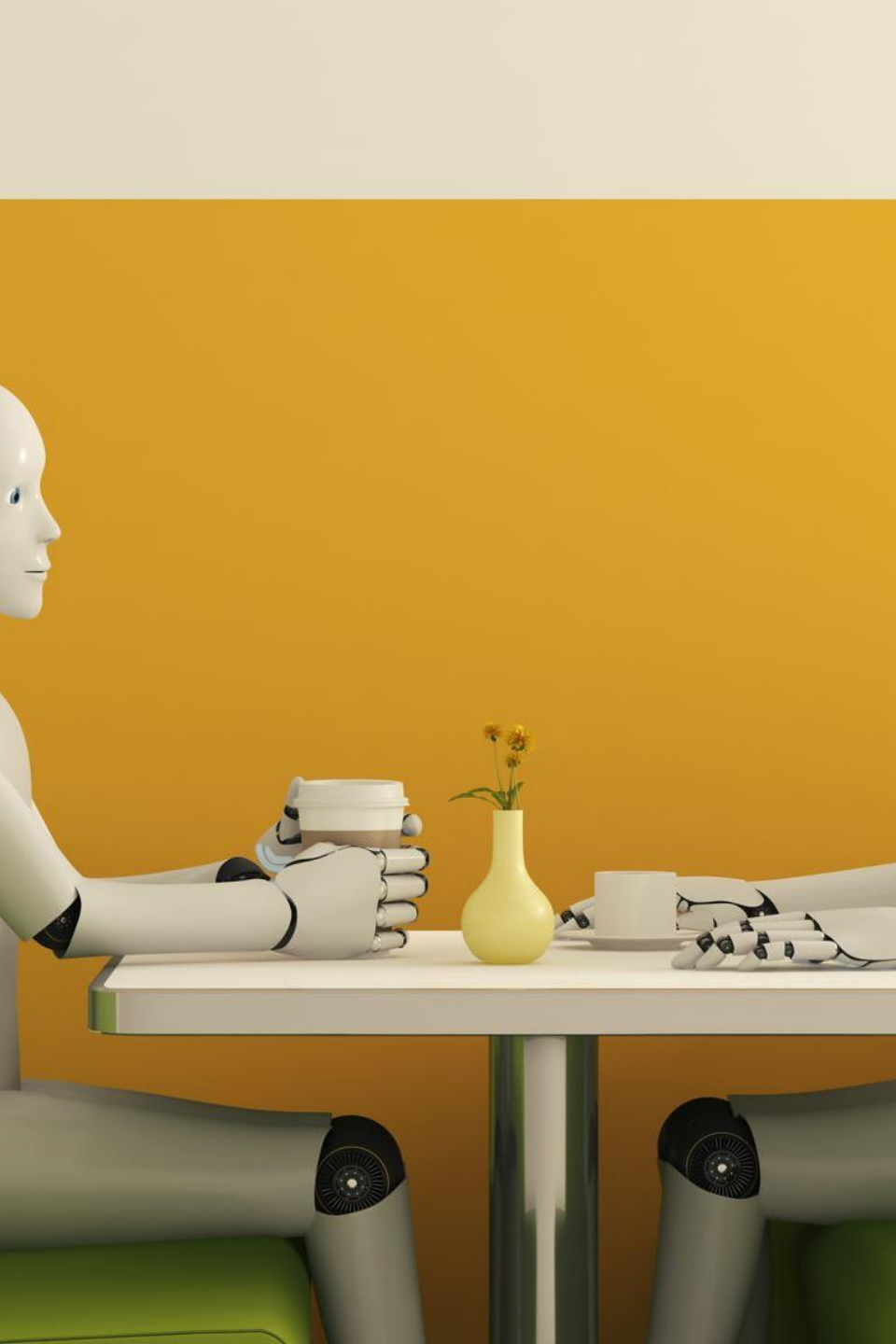
<https://voyager.minedojo.org/>



Language agents (Ahn et al., 2022; Huang et al., 2022c)

3. Language Agents

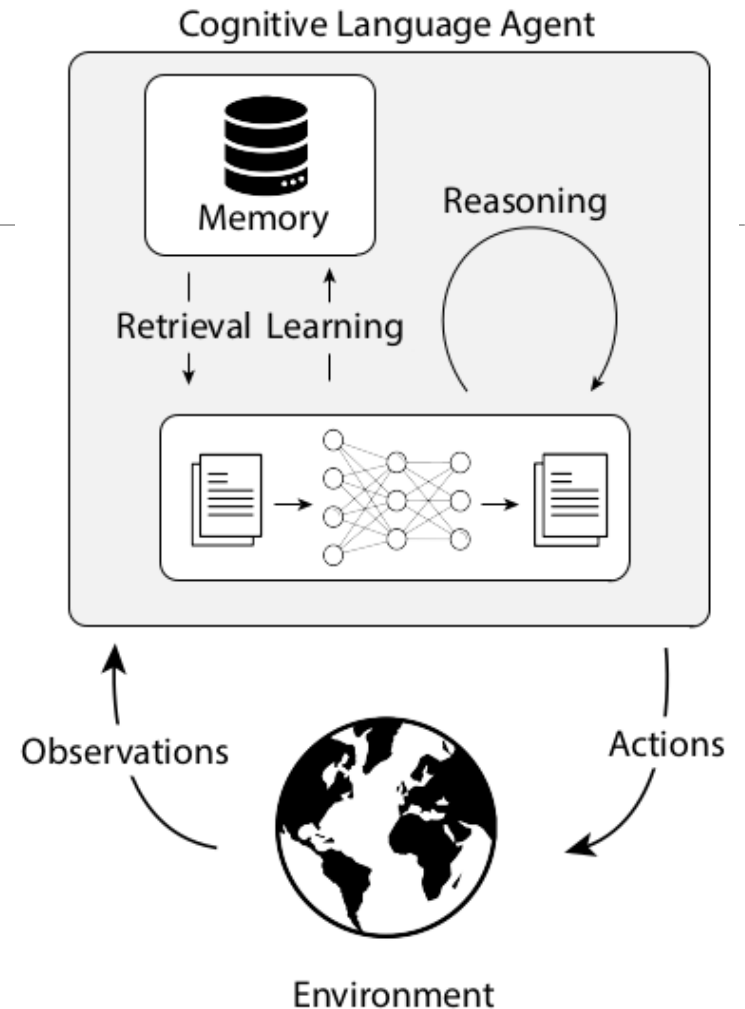




4. Arquitecturas cognitivas para agentes del lenguaje natural

4. Cognitive Architectures for Language Agents

Los **Cognitive Language Agents**, simulan una arquitectura cognitiva utilizando LLMs para gestionar el estado interno del agente a través de procesos como el aprendizaje y el razonamiento.



Cognitive language agents (Yao et al., 2022b; Shinn et al., 2023; Wang et al., 2023a)

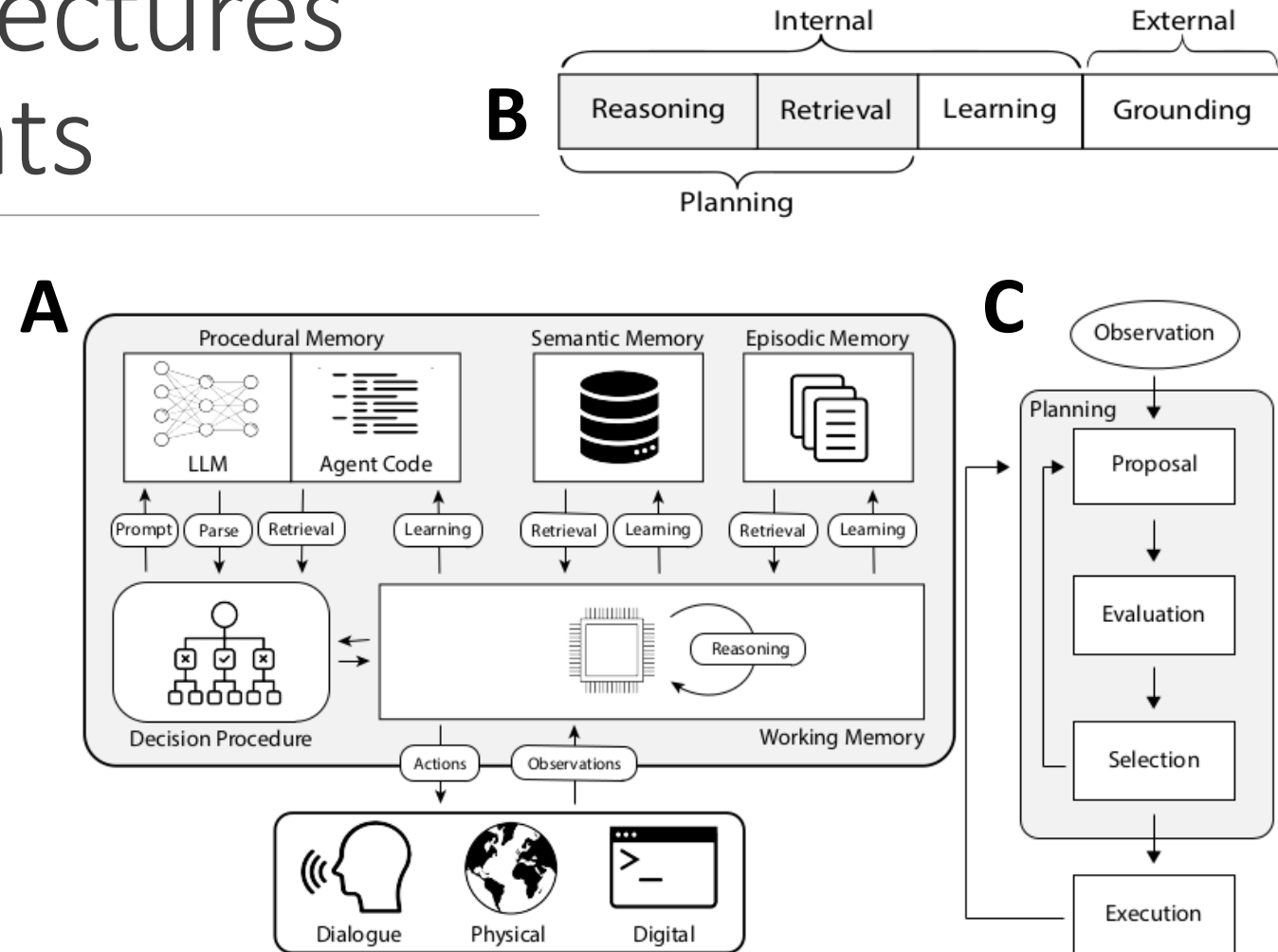
4. Cognitive Architectures for Language Agents

Cognitive Architectures for Language Agents (CoALA), un framework conceptual para caracterizar y diseñar agentes lingüísticos de propósito general.

A. Almacenamiento de información (memoria largo/corto plazo)

B. Espacio de acción (internas y externas)

C. Procedimiento de toma de decisiones (planificación y ejecución)

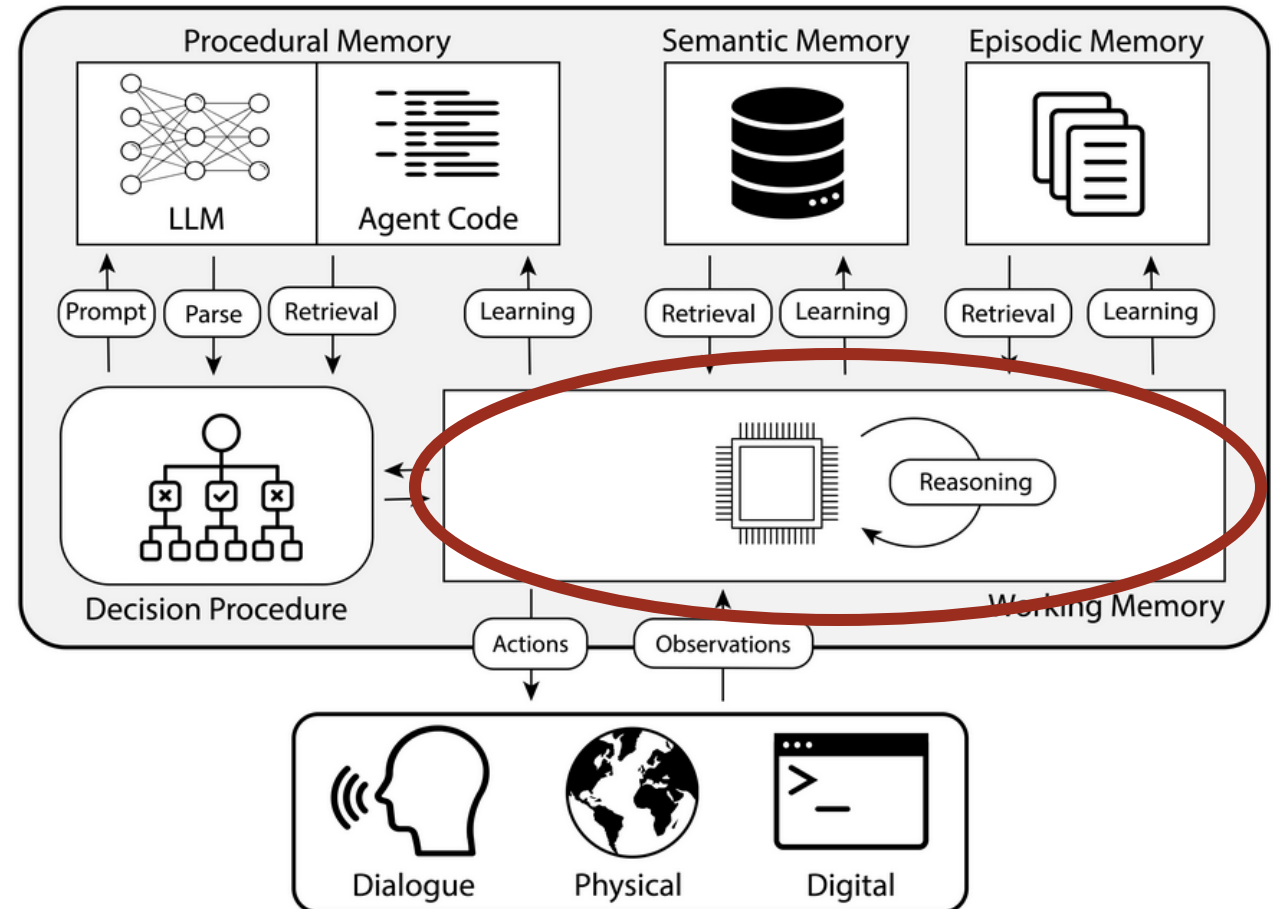


[Sumers, Theodore & Yao, Shunyu & Narasimhan, Karthik & Griffiths, Thomas. (2023). Cognitive Architectures for Language Agents.]

4. Cognitive Architectures for Language Agents

Memoria de trabajo (a corto plazo)

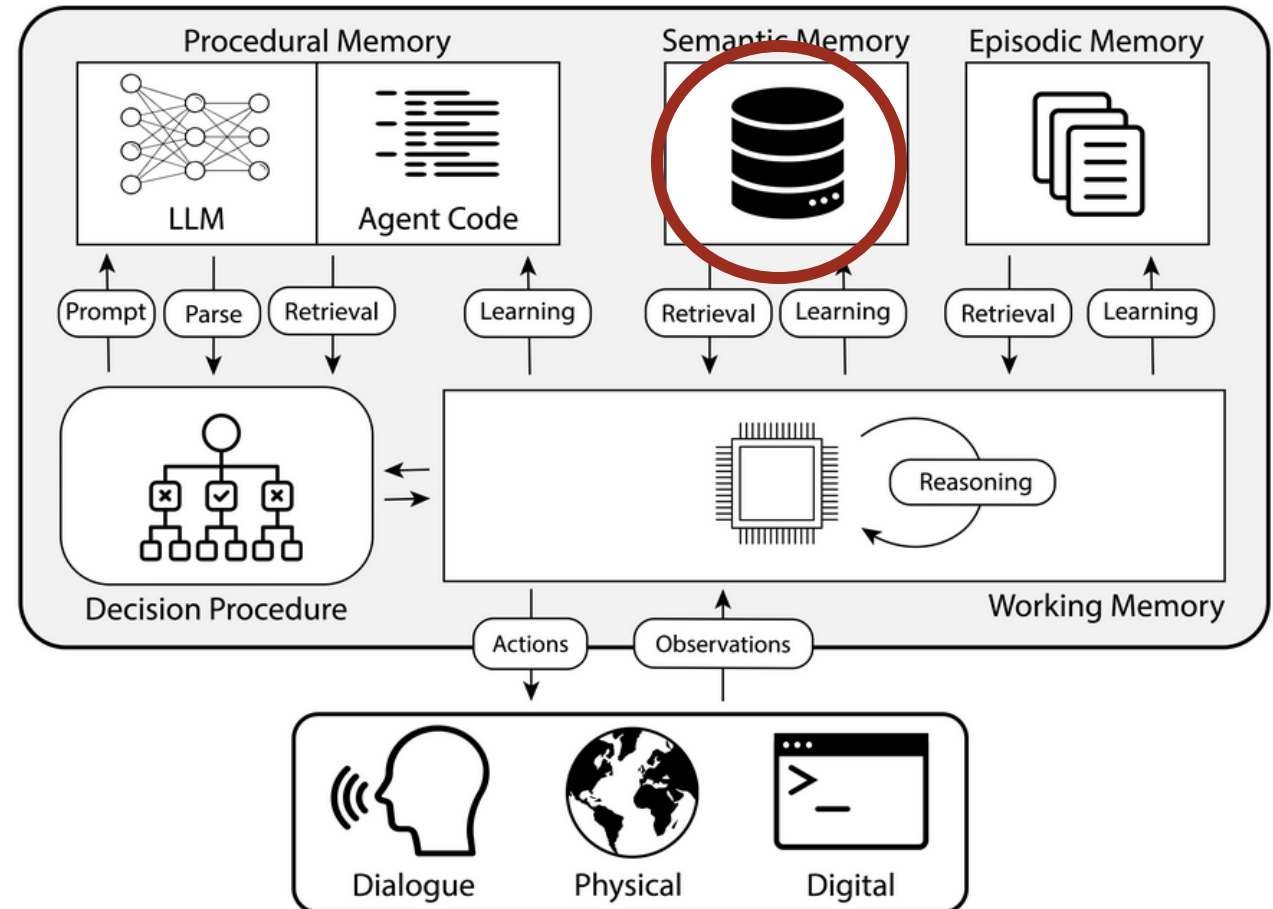
- Información relevante (variables simbólicas) para el **ciclo de decisión** actual.
- Conjunto de datos que persiste a través de las llamadas LLM (similar a un bloc de notas), que puede ser usado para razonar.
- Conectar diferentes componentes, como la memoria a largo plazo y los entornos externos.



4. Cognitive Architectures for Language Agents

Memoria semántica (a largo plazo)

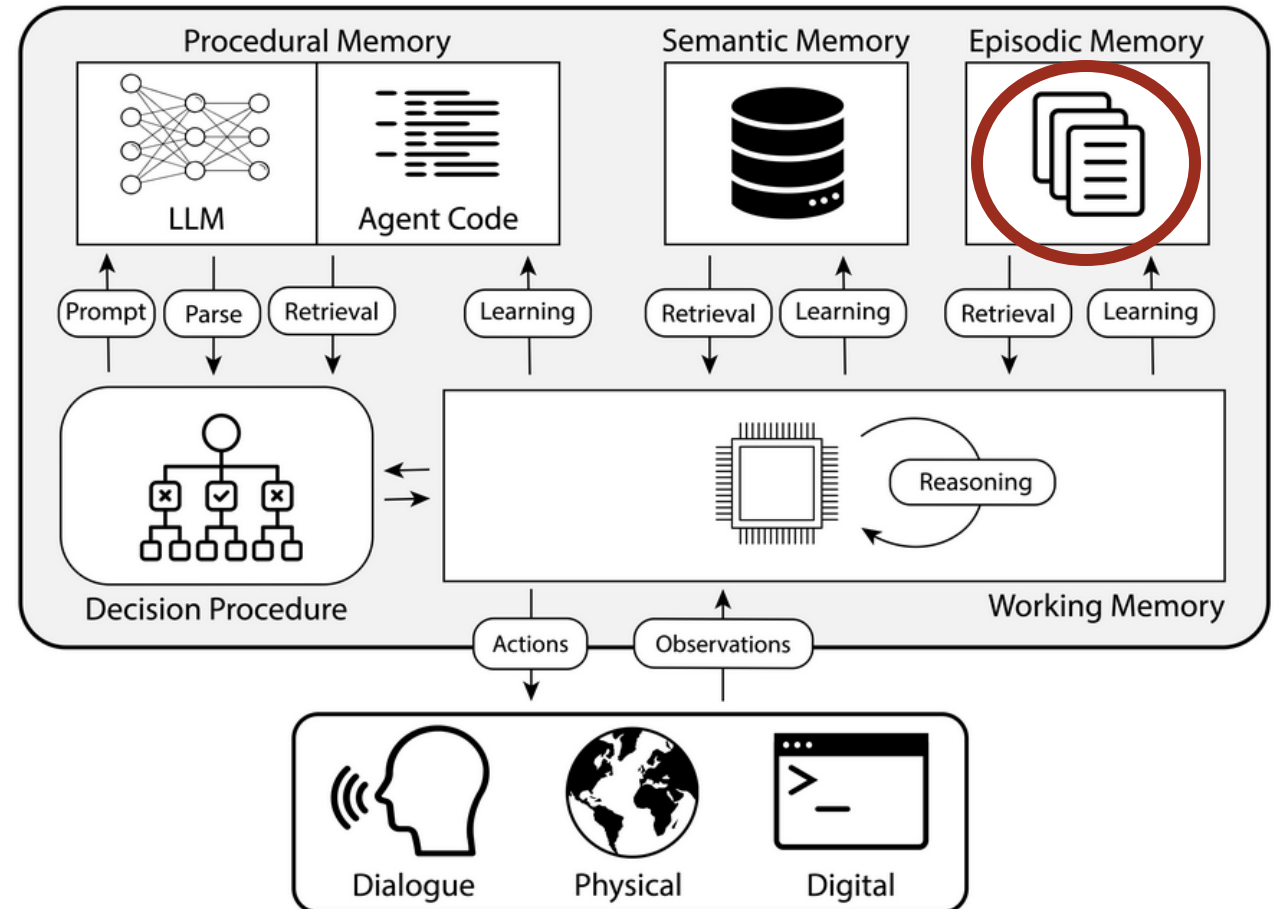
- **Conocimiento de hechos sobre él mismo y sobre mundo.**
- Utiliza fuentes de datos externas (documentos, wikis, bases de datos, etc.) para aumentar el conocimiento del LLM.
- Se puede actualizar las fuentes de datos externas o fine-tune el LLM con los aprendizajes del agente para mejorar esta memoria.



4. Cognitive Architectures for Language Agents

Memoria episódica (a largo plazo)

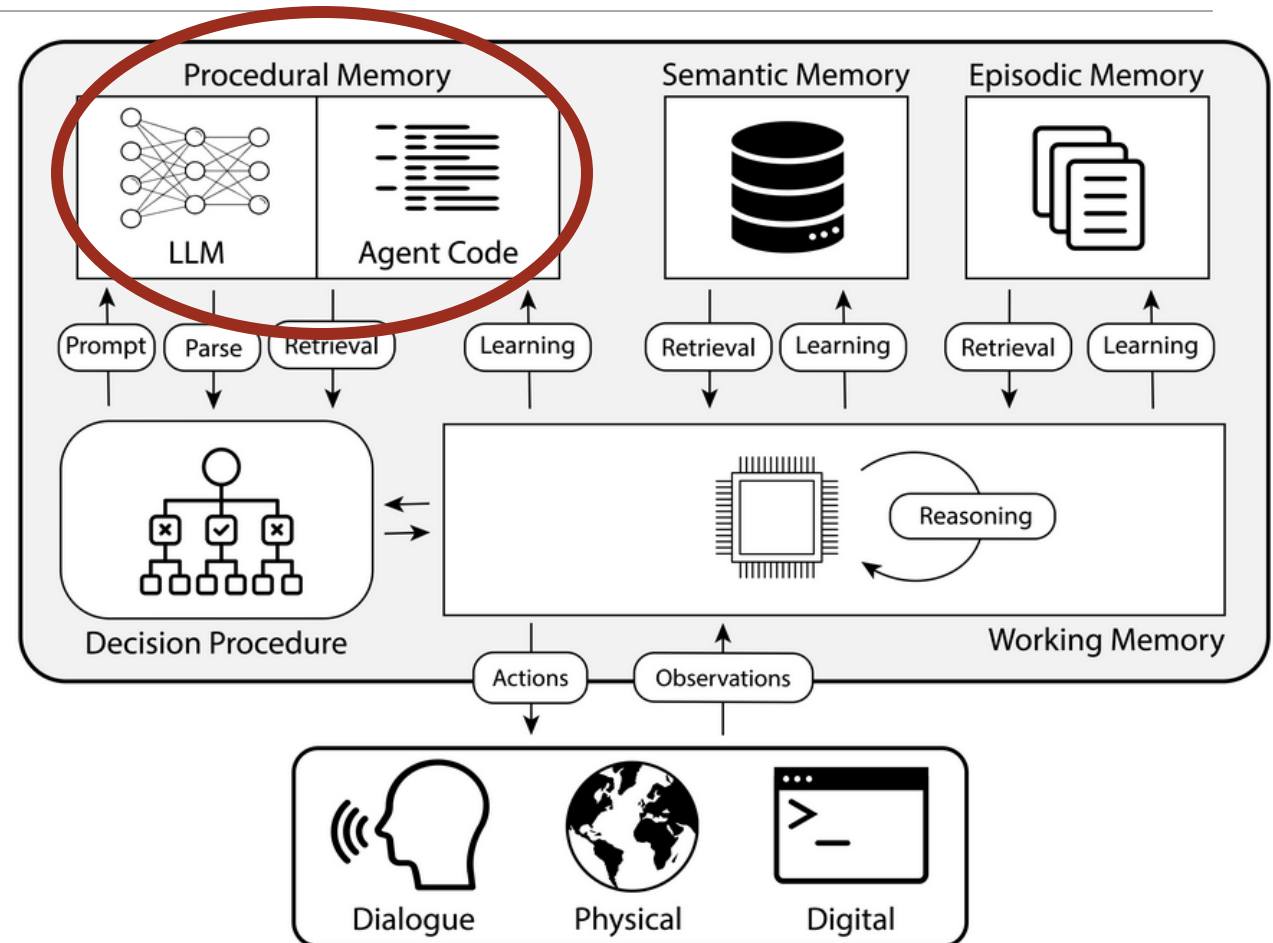
- Historial de experiencias de un agente en ciclos de decisión anteriores.
- El contenido se recupera en la memoria de trabajo y se utiliza para razonar y tomar decisiones.



4. Cognitive Architectures for Language Agents

Memoria procedimental (a largo plazo)

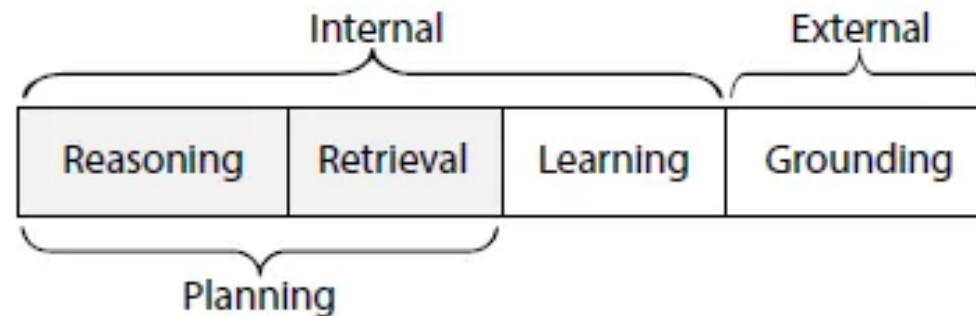
- **Conocimiento implícito**, integrado en los pesos del LLM.
- **Conocimiento explícito** escrito por el desarrollador en el código del agente. Puede implementar acciones (tales como procedimientos de razonamiento, recuperación, grounding y aprendizaje) y procedimientos para facilitar la toma de decisiones.
- Inicializada cuando el agente es puesto en marcha.



4. Cognitive Architectures for Language Agents

Acciones

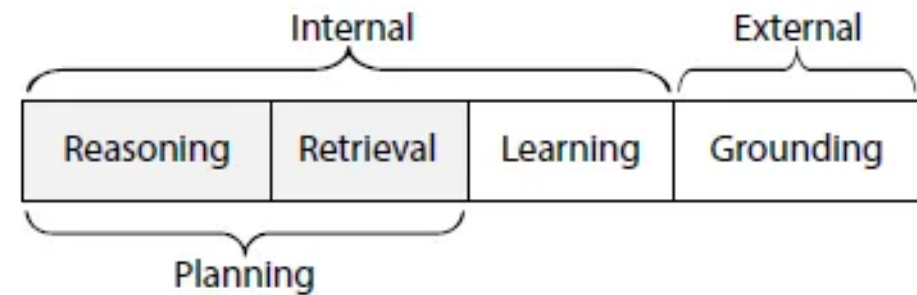
- **Acciones externas.** Interacción con entornos externos (**grounding**). Retroalimentación en la memoria de trabajo.
- **Acciones internas.** Interacción con la memoria interna (**razonamiento, recuperación, aprendizaje**). Actualización de la memoria semántica, episódica y procedimental.



4. Cognitive Architectures for Language Agents

Grounding

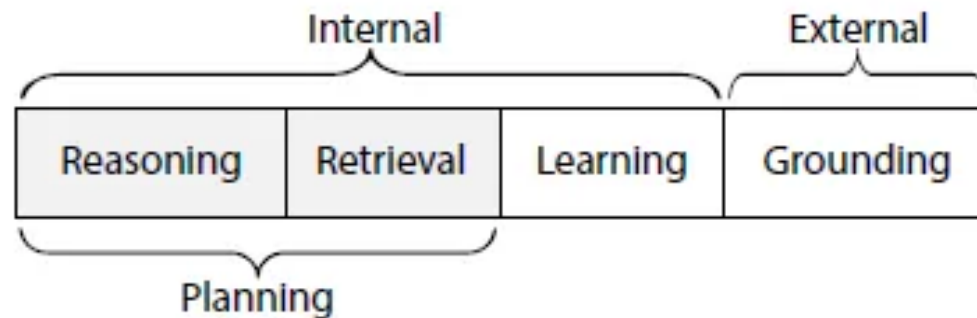
- Los procedimientos de grounding ejecutan acciones externas y procesan la información del entorno en la memoria de trabajo en forma de texto. Esto simplifica de forma efectiva la interacción del agente con el mundo exterior como un "juego de texto" con observaciones y acciones textuales.
 - Entornos físicos.
 - Diálogo con humanos u otros agentes.
 - Entornos digitales.



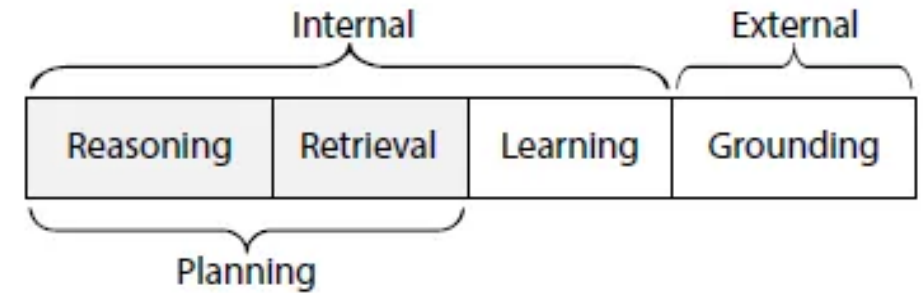
4. Cognitive Architectures for Language Agents

Recuperación

- **Lee información de las memorias de largo plazo a la memoria de trabajo.** Dependiendo de la información y del tipo de memoria, podría implementarse de varias maneras, por ejemplo, recuperación basada en reglas, escasa o densa.
- Mientras que la recuperación desempeña un papel clave en la toma de decisiones humana (Zhou et al., 2023a; Zhao et al., 2022), la recuperación adaptativa y específica del contexto sigue sin estudiarse en los agentes lingüísticos.



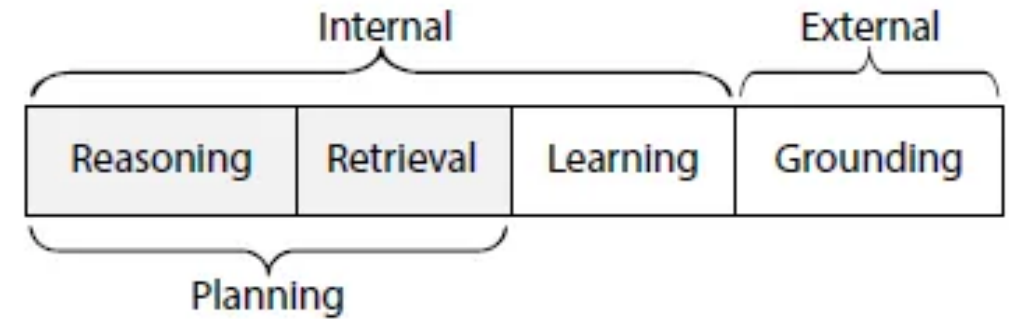
4. Cognitive Architectures for Language Agents



Razonamiento

- Procesa el contenido de la **memoria de trabajo** para generar nueva información.
- Lee de la memoria de trabajo y escribe en ella. Esto permite al agente resumir y extraer información sobre la observación más reciente (Yao et al., 2022b; Peng et al., 2023), la trayectoria más reciente (Shinn et al., 2023) o la información recuperada de la memoria a largo plazo (Park et al., 2023).
- El razonamiento puede utilizarse para apoyar el aprendizaje (escribiendo los resultados en la memoria a largo plazo) o la toma de decisiones (utilizando los resultados como contexto adicional para posteriores llamadas LLM)

4. Cognitive Architectures for Language Agents



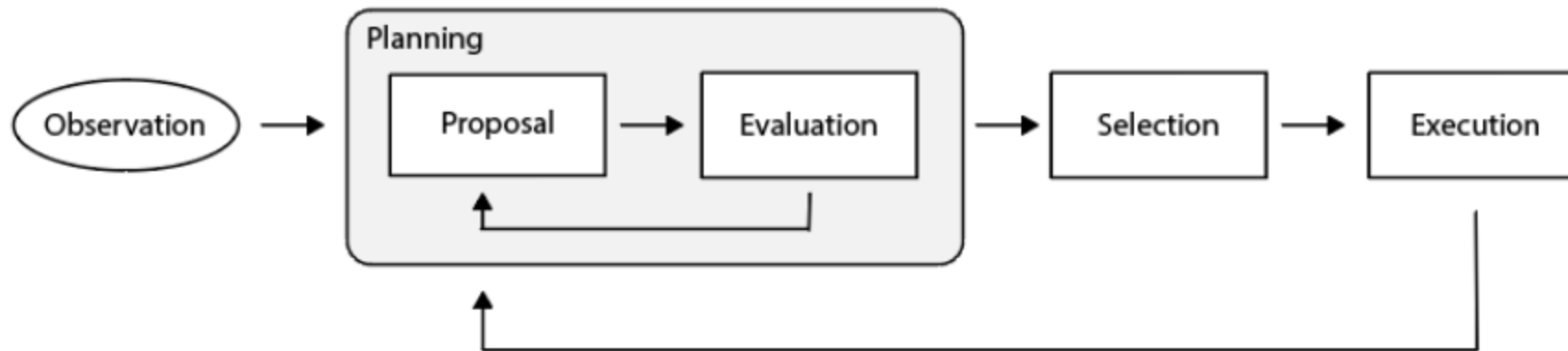
Aprendizaje

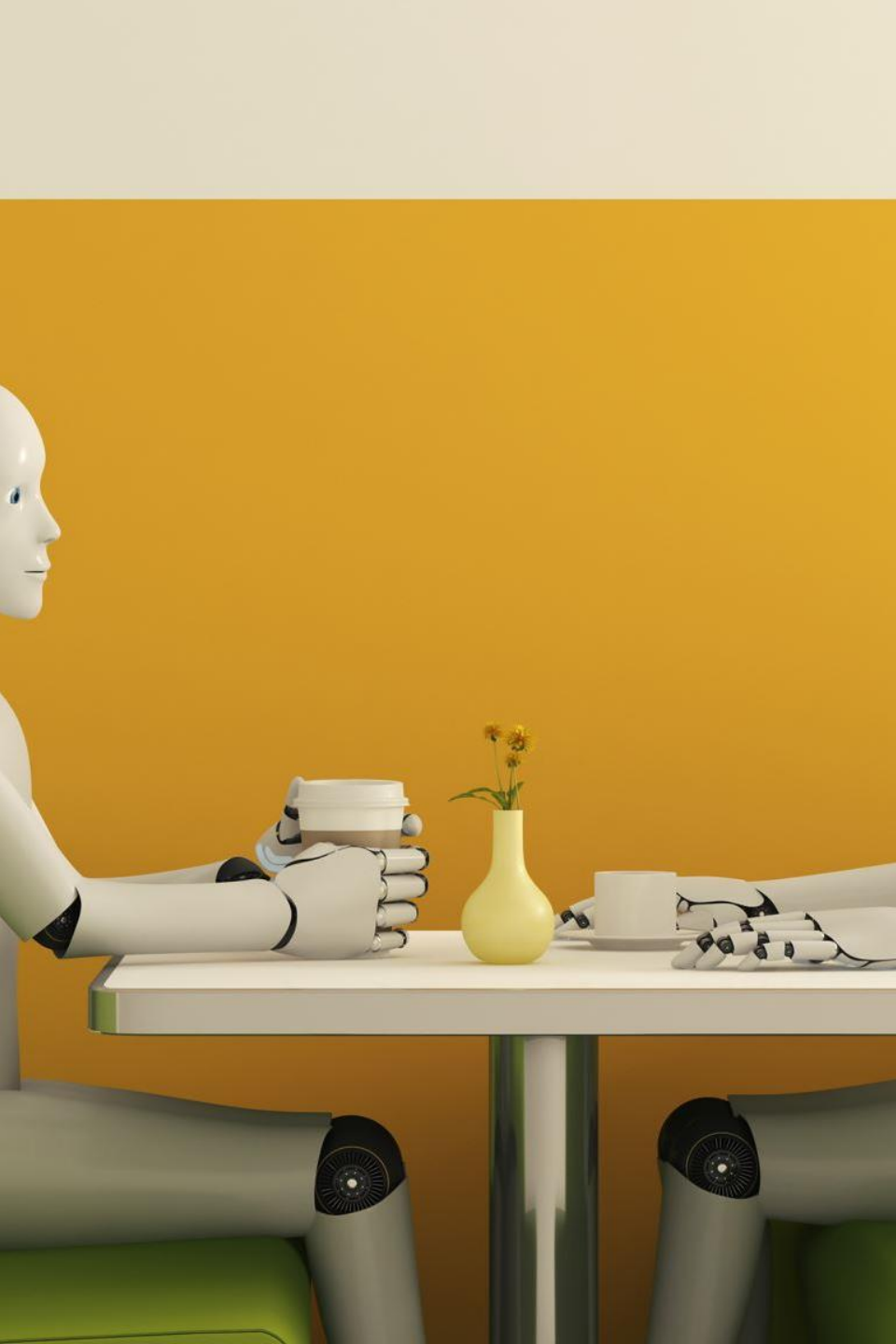
- El aprendizaje se produce escribiendo información en la memoria a largo plazo.
 - Actualización de la memoria episódica con la experiencia.
 - Actualización de la memoria semántica con el conocimiento.
 - Actualización de los parámetros LLM (memoria procedimental).
 - Actualización del código del agente (memoria procedimental). Modificación del código fuente.

4. Cognitive Architectures for Language Agents

Toma de decisiones

- Las acciones de un agente se realizan en ciclos de toma de decisiones. Cada ciclo está formado por dos etapas:
 - **Planificación.** Razonamiento/recuperación para (iterativamente) proponer y evaluar acciones.
 - **Ejecución.** La acción de aprendizaje/ground seleccionada se ejecuta para afectar a la memoria interna o al mundo externo.





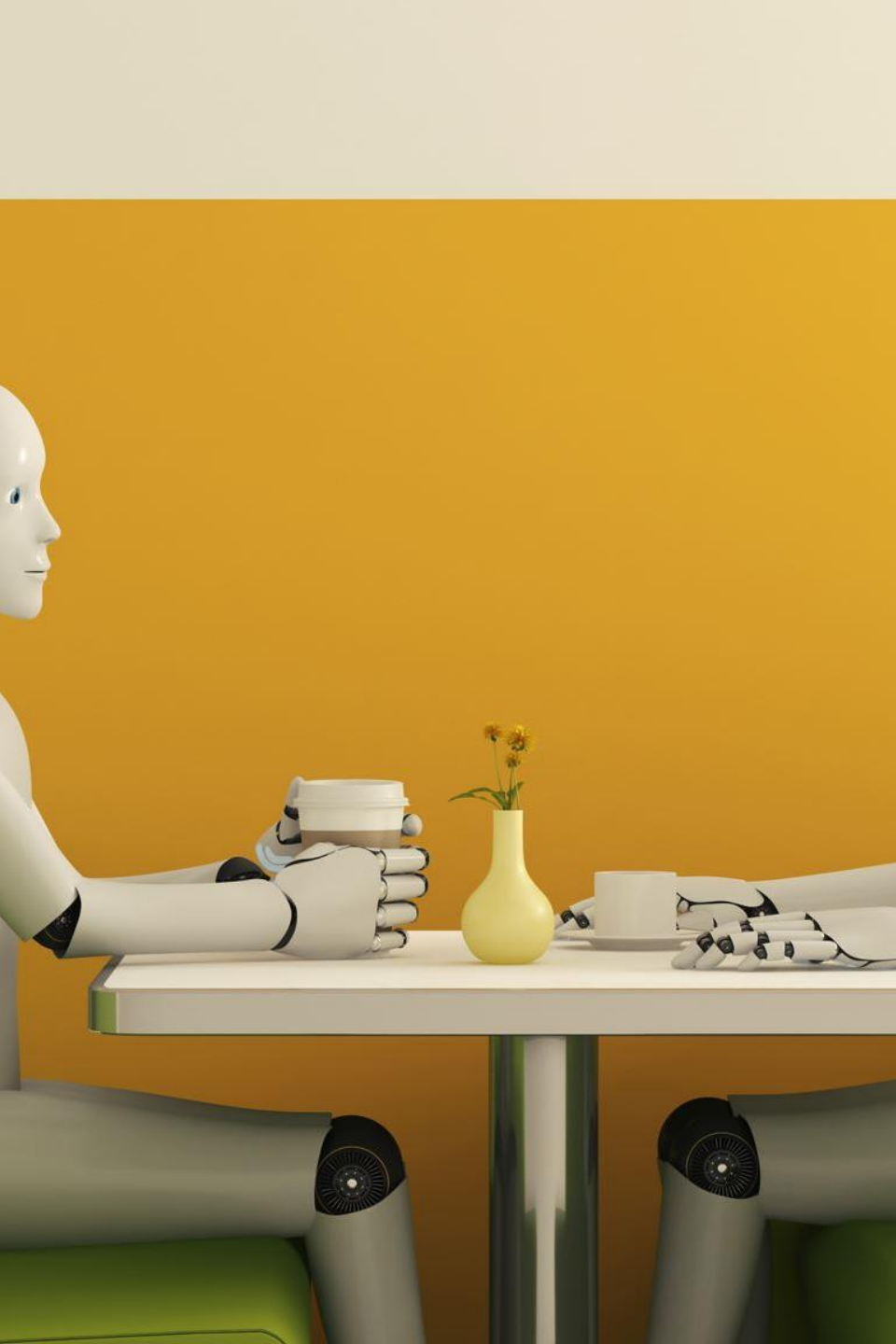
4. Futuro

*<<The best way to predict the future is to create it.>>
Abraham Lincoln.*

4. Futuro



<https://www.youtube.com/watch?v=x4O8pojMF0w>



Referencias bibliográficas

Referencias bibliográficas

- Hands-On Large Language Models. Jay Alammar, Maarten Grootendorst, O'Reilly Media, Inc. (2024).
- LLM Reasoning: Key Ideas and Limitations. Denny Zhou, Google DeepMind. UC Berkeley (2024)
- LLM Agents. Shunyu Yao. UC Berkeley (2024)
- Cognitive Architectures for Language Agents. Theodore Sumers and Shunyu Yao and Karthik Narasimhan and Thomas Griffiths, Transactions on Machine Learning Research (2024)