

IHC

2026

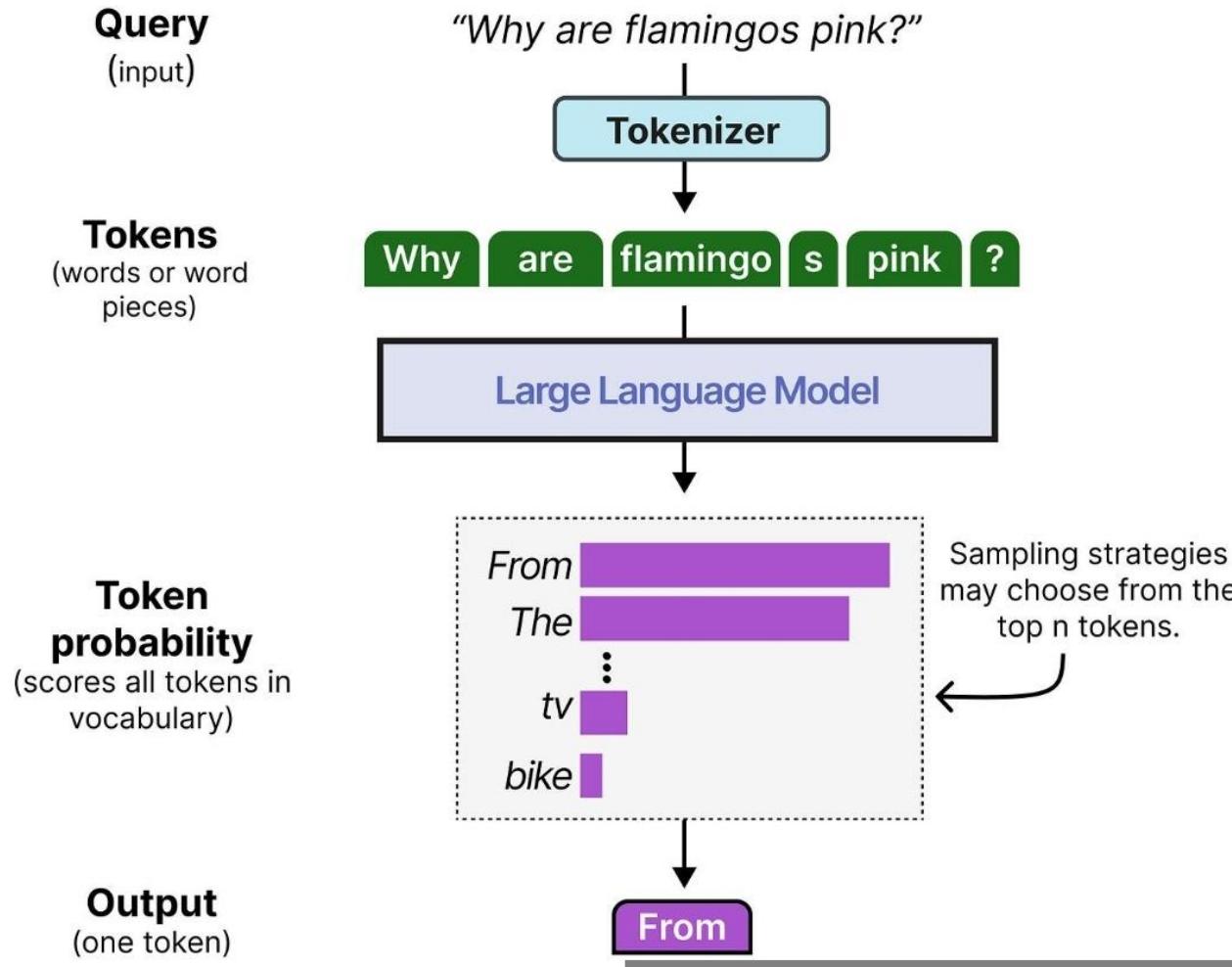
CLI -> GUI -> NUI -> NLI
~70 ~80 2007 2022
ling da metáforas sentidos linguagem
máquina natural

Projeto: NLI

Agentes (LM, tools, memory and planning)
DSPy

Objetivo: Data Agent (só vale opensource)





I saw **6** flamingos. **2** flew away. **1** hid behind a tree. How many can I see?

“regular” LLM

↓
answer only

You see
3 flamingos

Reasoning LLM

↓
reasoning

- You saw **6** flamingos
- **2** flew away, you can see **4**
- **1** hid, you can see **3**

↓
answer

You see
3 flamingos

Why are flamingos pink?



LLM

Show reasoning



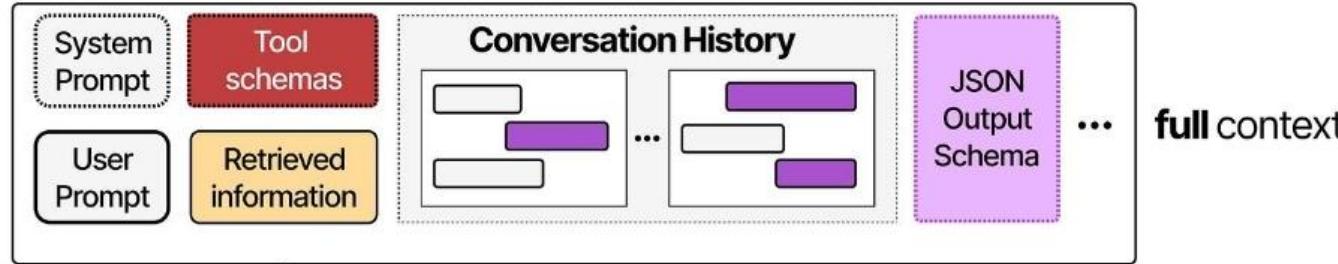
Because of pigments
in their diet.

reasoning

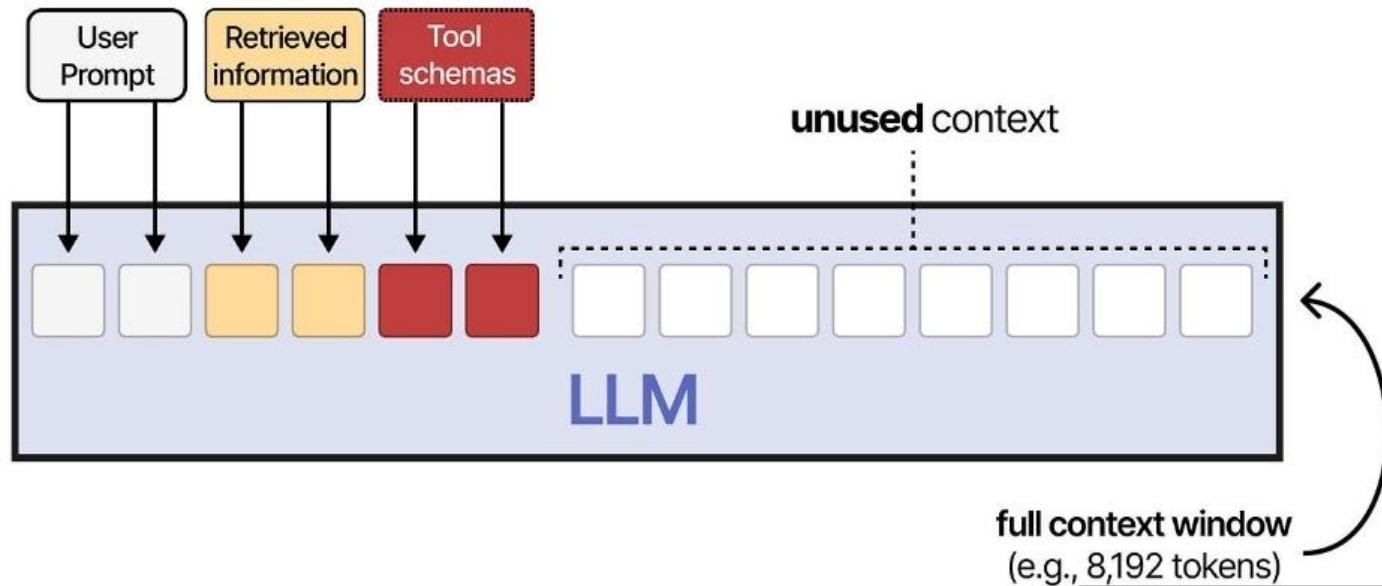
Flamingos are pink because their diet includes algae and brine shrimp, which contain pigments that get metabolized and deposited in their feathers, skin, and beaks.

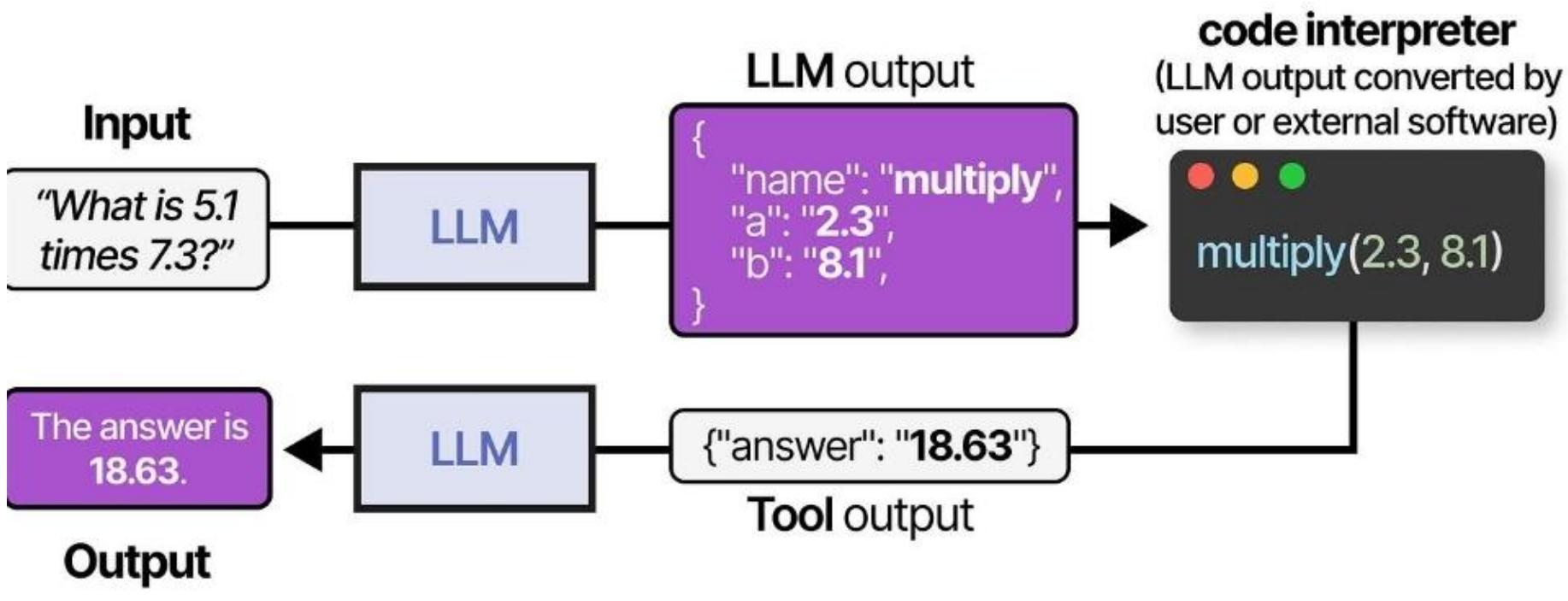


The full **reasoning** is typically hidden whereas the **answer summarizes** the reasoning instead.



select context





Query

Research
“AI Agents”



Output (a plan)

To research **AI Agents** I have to perform the following tasks:

- [] **Google** for relevant information
- [] Research papers on **ArXiv**
- [] **Summarize** the results

Plan

To research **AI Agents** I have to perform the following tasks:

- [x] **Google** for relevant information
- [] Research papers on **ArXiv**
- [] **Summarize** the results

Reasoning LLM

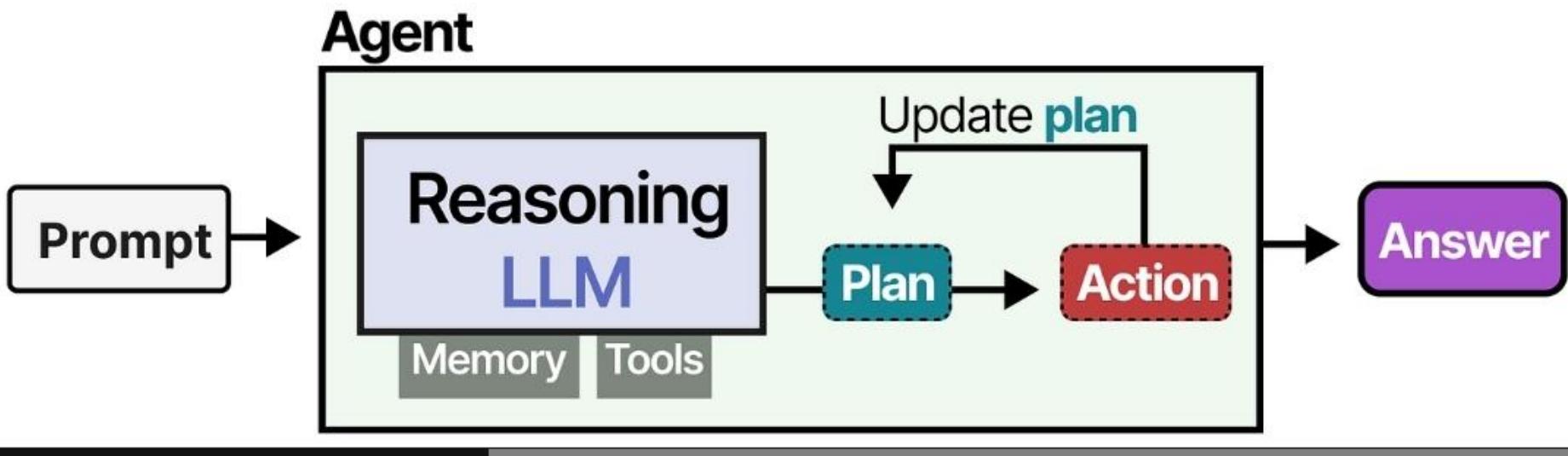
Memory

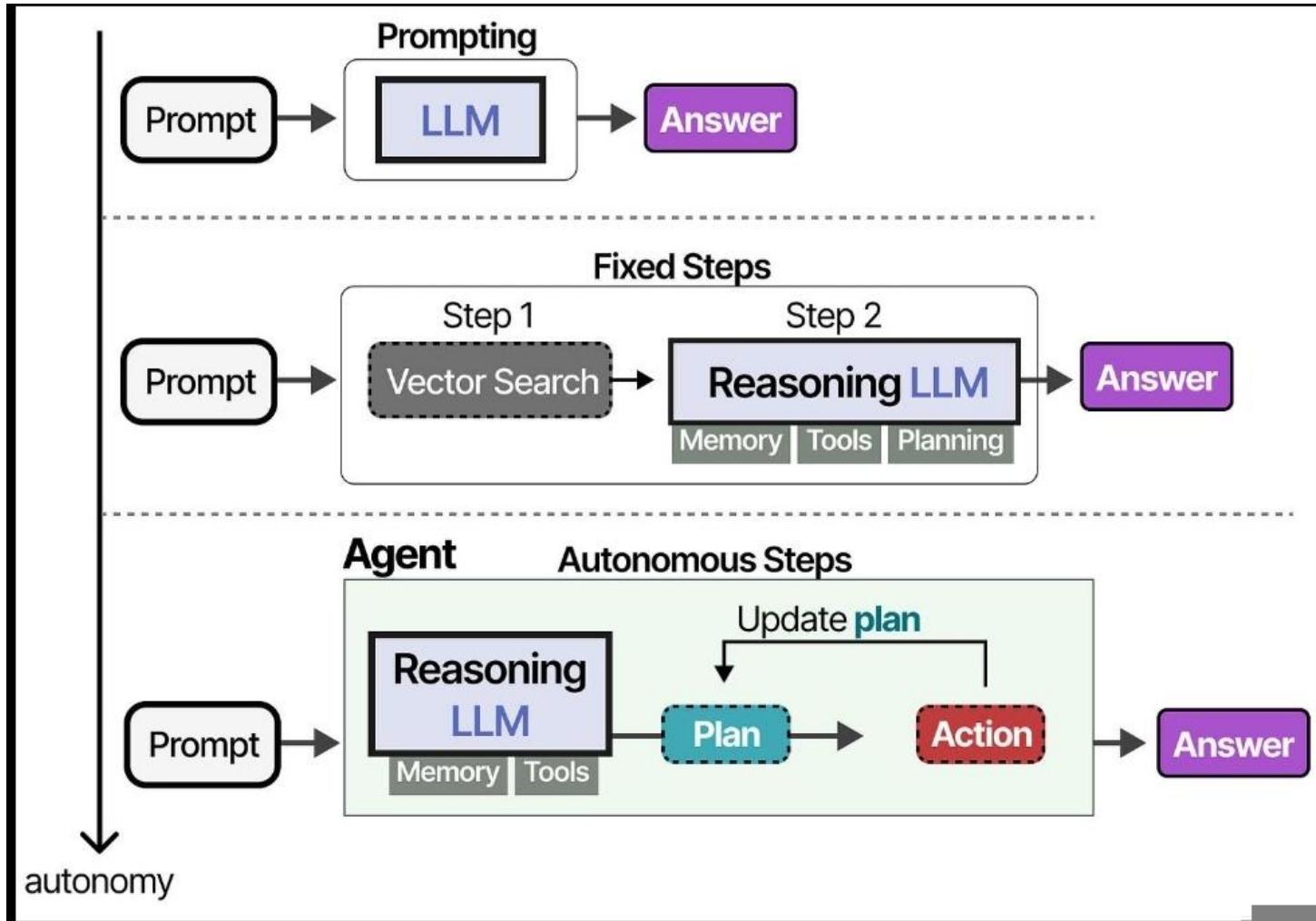
Tools

Planning

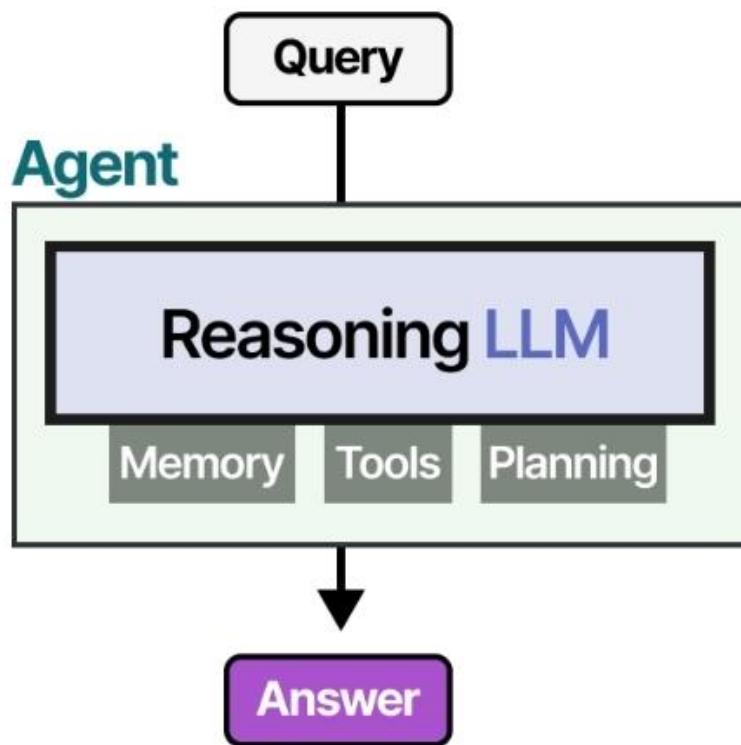
Next Steps

I have **finished** the **Google** search. I should now research papers on **ArXiv**.

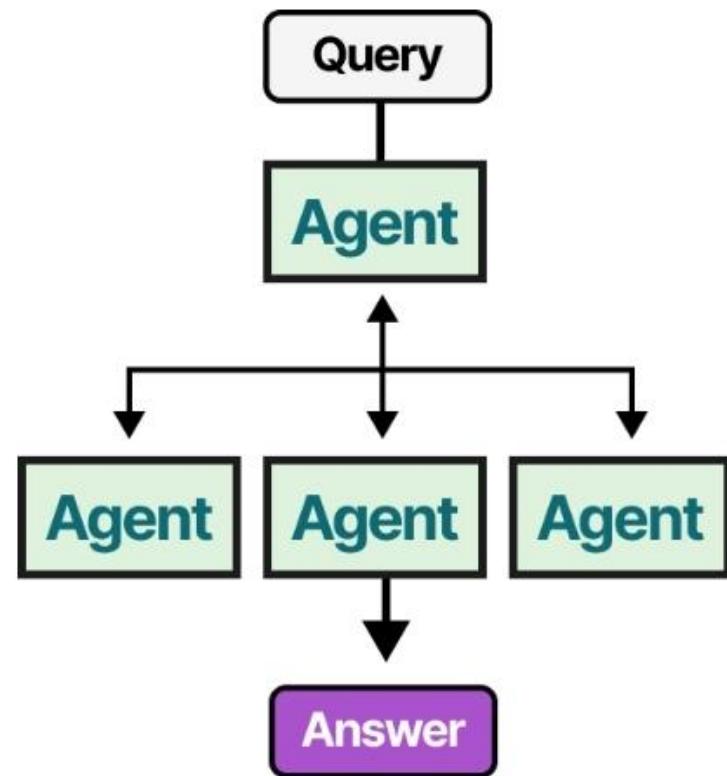




Single Agent



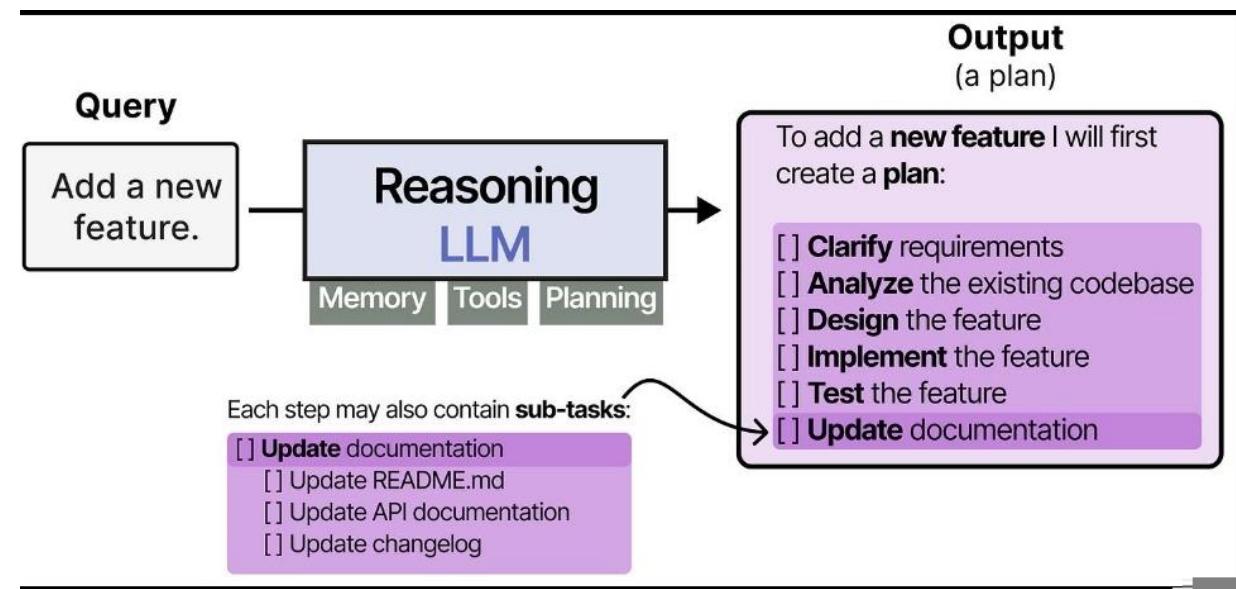
Multi-Agent

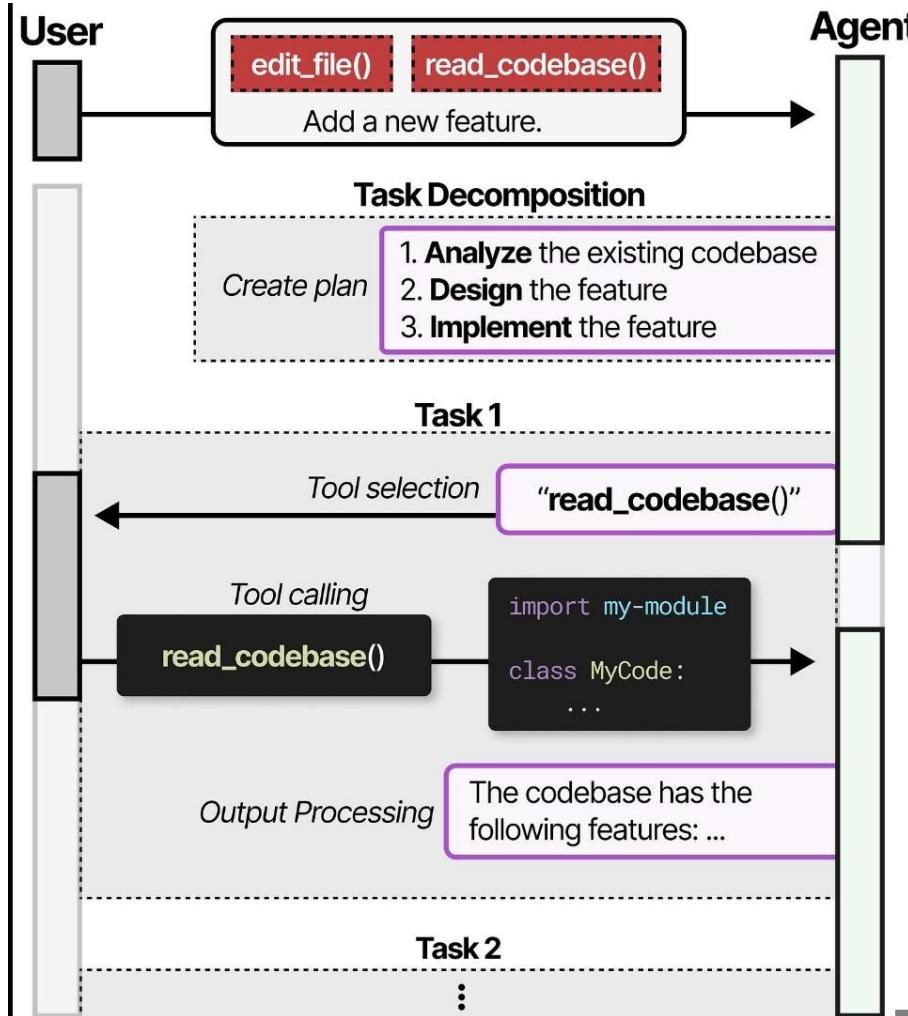


There is a wide variety of methodologies to enable planning in reasoning LLMs effectively.

Task Decomposition

The first step in planning is to decompose an initial query into subtasks. Like the example at the beginning of the chapter, when you ask an Agent to create a certain feature, it will split that query up into smaller tasks to execute. This is called Task Decomposition and allows the Agent to simplify complicated tasks.¹ Shown in Figure is an example of the reasoning LLM creating this plan, where each task may also have a set of sub-tasks to complete.





Chain-of-Thought

A major component of task decomposition, aside from having a reasoning LLM, is prompt engineering. We covered several of them in Chapter 3 from the perspective of reasoning traces, where several “thoughts” are sampled. We covered Chain-of-Thought (CoT) as a common approach to create these reasoning traces. Remember that CoT is a process where the LLM is asked to solve problems step-by-step. This can be achieved by either providing examples to the LLM (few-shot prompting) or by simply stating “Let’s think step-by-step” (zero-shot prompting). This step-by-step process is, in fact, task decomposition. The original query is processed through separate reasoning traces that rely on the ones that came before (seen in Figure). In other words, it breaks down the problem into sequential reasoning substeps.

Question

I saw 6 flamingos. 2 flew away. 1 hid behind a tree. How many can I see?
Let's think step-by-step.

Reasoning steps

First, you saw **6** flamingos.

2 flew away and you can see **4**.

1 hid so now you can see **3**.

Final answer

3

task decomposition

User: Add a new feature to my existing codebase. Let's think step-by-step.

LLM (CoT):

First, I need to clarify the requirements. What kind of feature is needed?

Second, I should analyze the existing codebase to understand the architecture.

Third, I need to design the feature and make sure nothing is left out.

Fourth, I should implement the feature in the existing codebase but without changing its original functionality.

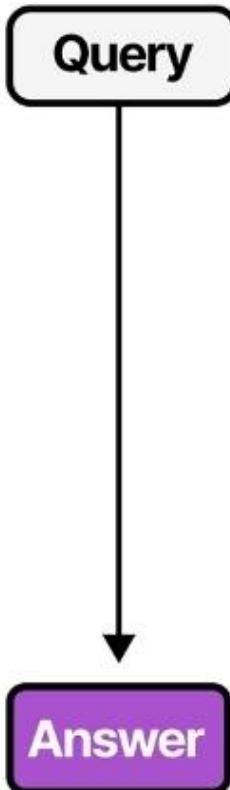
Fifth, I must test the feature thoroughly through unit and integration tests.

Finally, I need to update the documentation to explain how the new feature works.

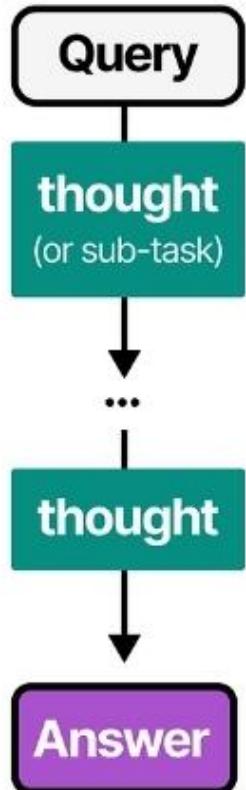
LLM (Plan):

The plan will be as follows:

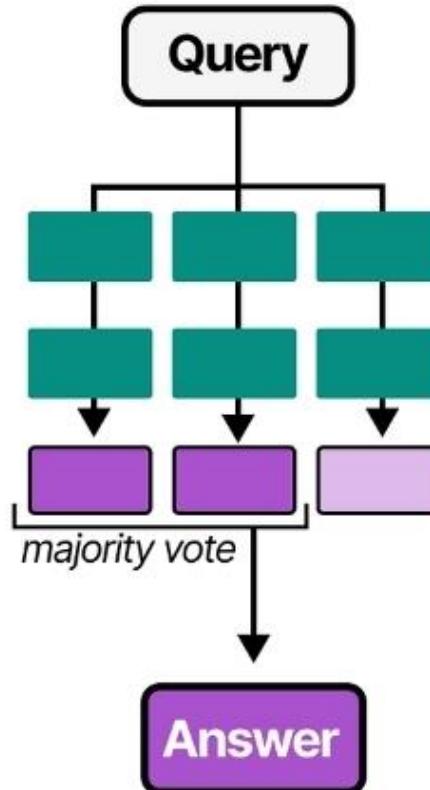
- [] Clarify requirements
- [] Analyze the existing codebase
- [] Design the feature
- [] Implement the feature
- [] Test the feature
- [] Update documentation



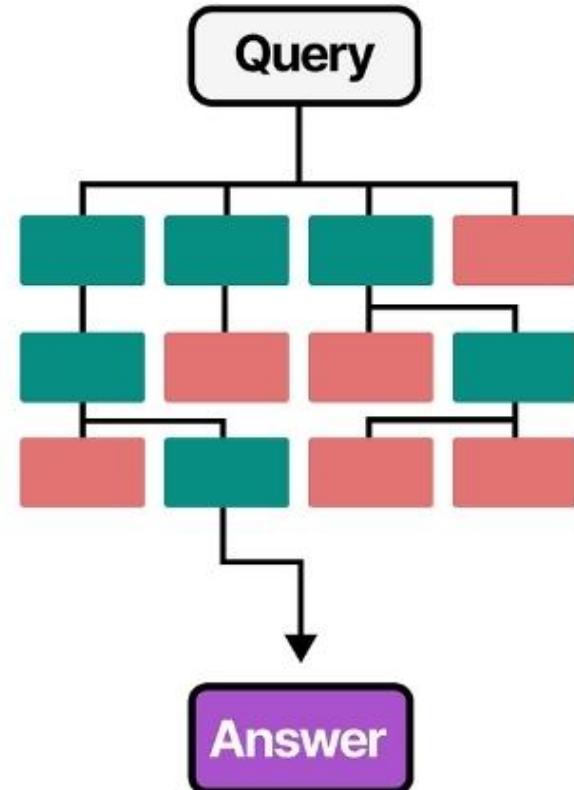
regular
prompting



Chain-of-Thought (CoT)



Self-consistency with CoT



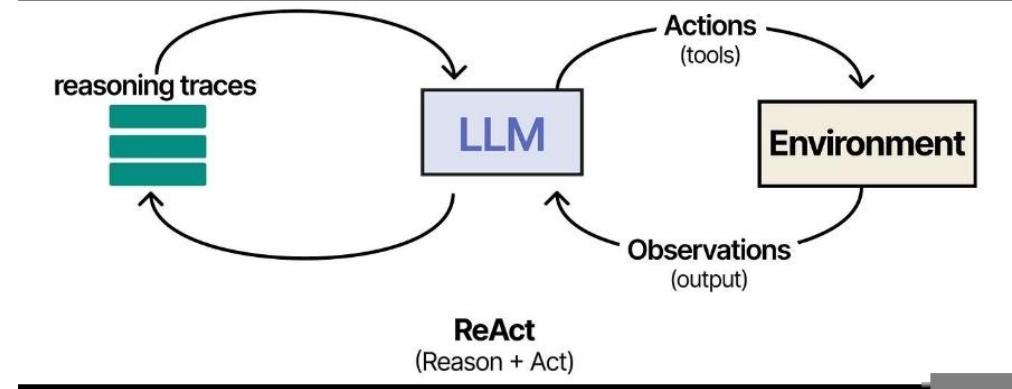
Tree of Thoughts (ToT)

The interleaving of reasoning and actions creates a feedback loop in which the model repeatedly cycles through a thought-action-observation process. In each loop, the model is asked to separate its textual output into three components:

Thought
A reasoning step about the current situation

Action
A set of actions to execute (e.g., tools)

Observation
A reasoning step about the result of the action



<https://github.com/haasonsasa/dspy-0to1-guide>

Projeto: Data Agent (com dspy)

primeiro passo: text2sql

depois: visualizações e etc