



I².A²

institut d'intelligence
artificielle appliquée

i2a2.academy





Fonte: <https://www.utu.fi/en/news/news/a-brief-overview-of-kievis-approach-to-using-ai>

Agentes Autônomos com Redes Generativas

Planejamento de Agentes e Ferramentas

Promovido

Orientações Gerais



Orientações

Sobre os Grupos de Trabalho

Todas as alterações solicitadas foram executadas. Em breve vocês receberão os e-mails com as alterações.

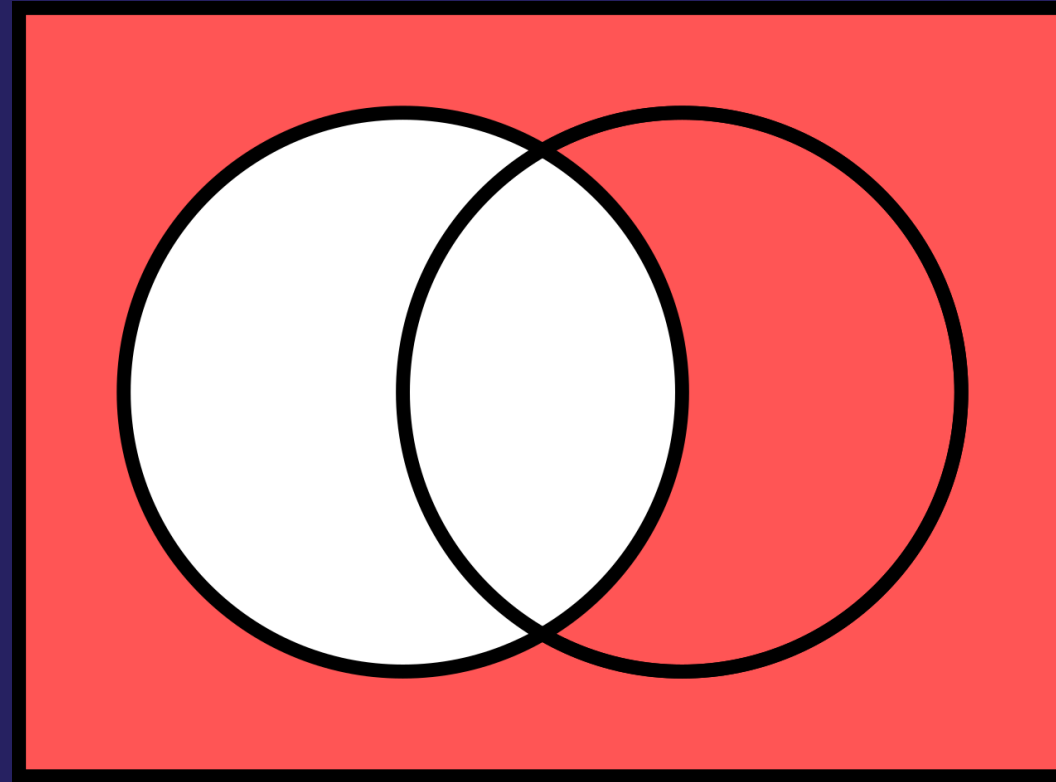
Caso ainda exista alguma divergência, por favor, avisem utilizando o e-mail **sac@i2a2.academy**.

Sobre os grupos de WhatsApp do curso

Criamos a comunidade para ter um espaço exclusivo para recados e abrigar ambos os grupos. Aparentemente o WhatsApp está tentando lidar com isto ainda transferindo participantes de um lado para o outro.

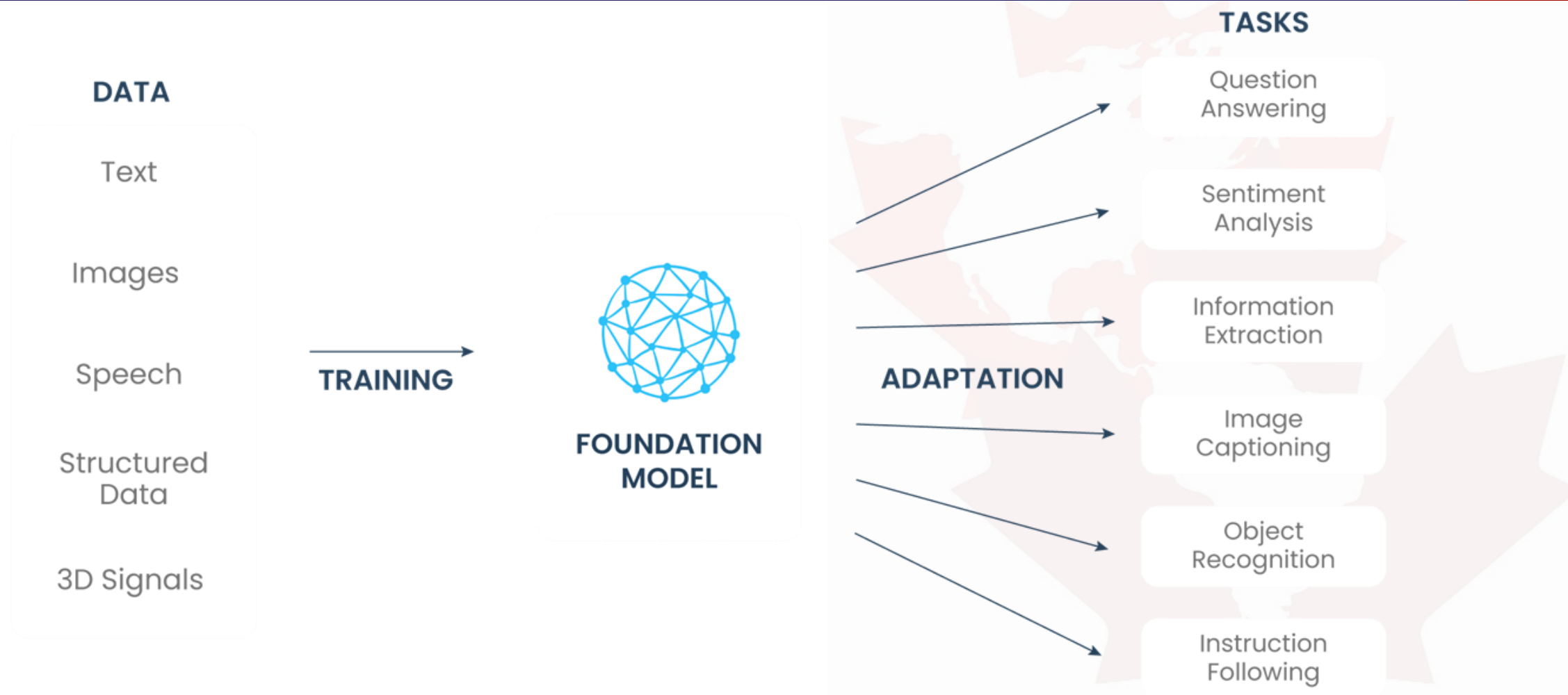
Sobre os projetos

Criamos um formulário para vocês informarem suas dúvidas. Por favor, utilizem-no.



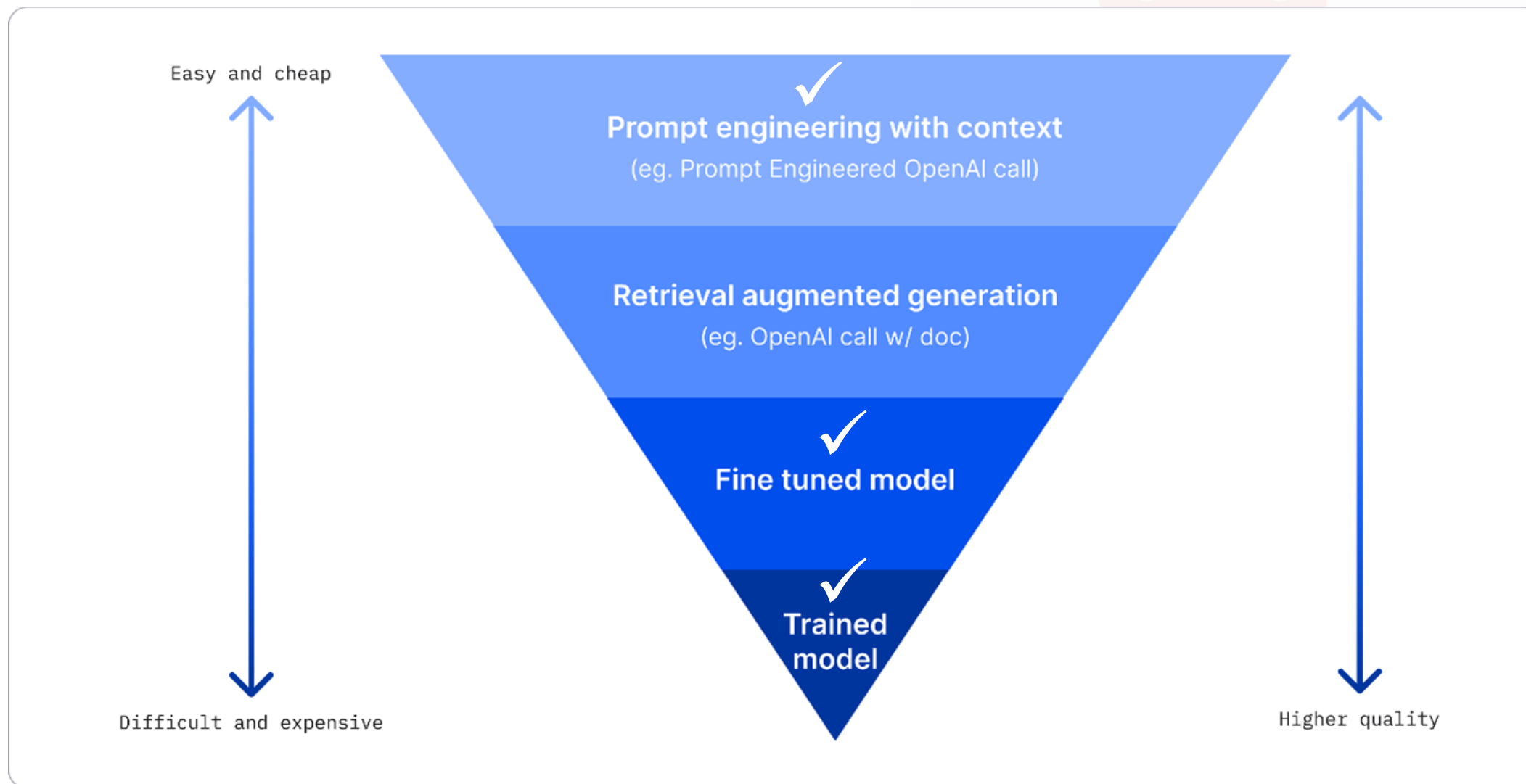
Complementando

O que é uma LLM?



Observação Importante: LLMs conhecem apenas os dados que foram utilizados durante o treinamento. Não conhecem fatos posteriores e não tem acesso à informações privadas.

Como Aprimorar uma LLM

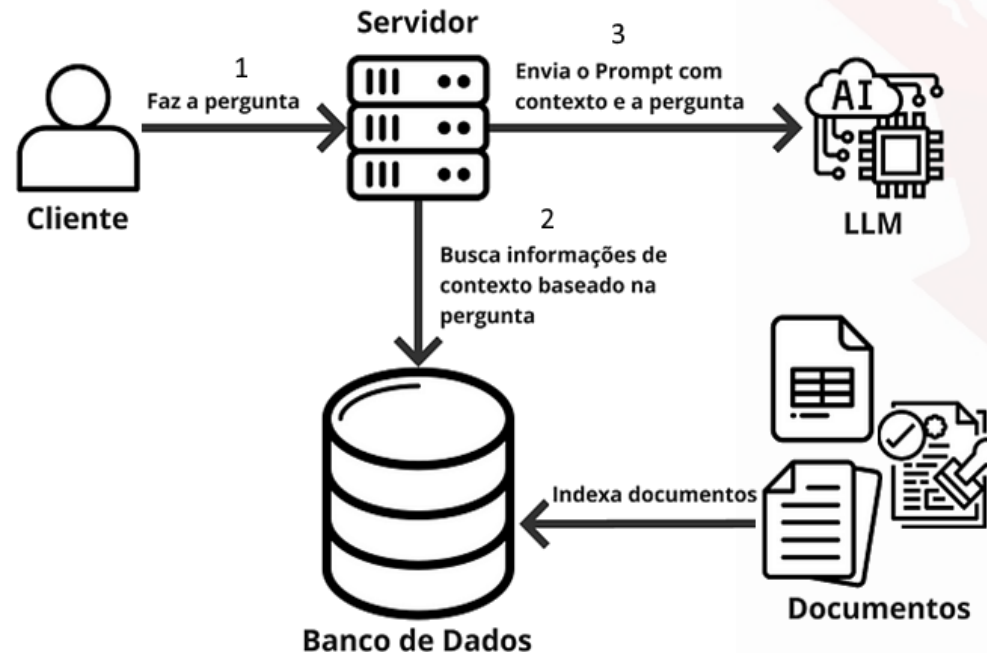


Fonte: https://www.fiddler.ai/blog/four-ways-that-enterprises-deploy-llms?WT.mc_id=academic-105485-koreyst

RAG

Retrieval Augmented Generation (RAG)

- RAG é uma técnica que complementa o prompt com dados externos na forma de blocos de documentos, considerando os limites de tamanho do prompt.





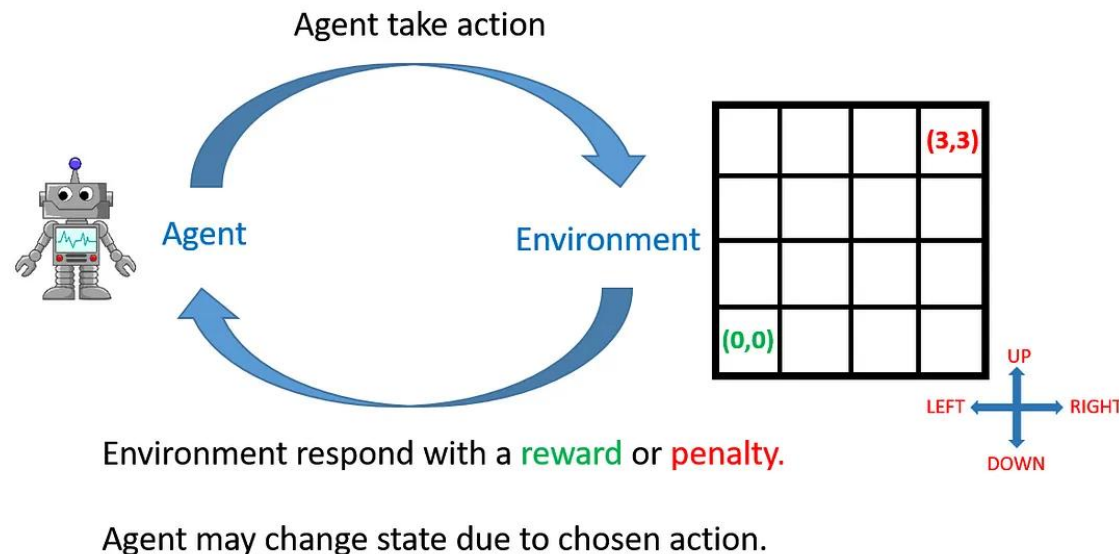
Agentes

<https://openclipart.org/detail/349203/people-in-black>

O que é um agente?

Sistema "inteligente" que interage com algum "ambiente"

- Ambientes físicos: robô, carro autônomo, ...
- Ambientes digitais: Deep Q Network para videogames, Siri, AlphaGo, ...
- Humanos como ambientes: Chatbots



O que é um Agente com LLM?

Nível 1: Agente textual

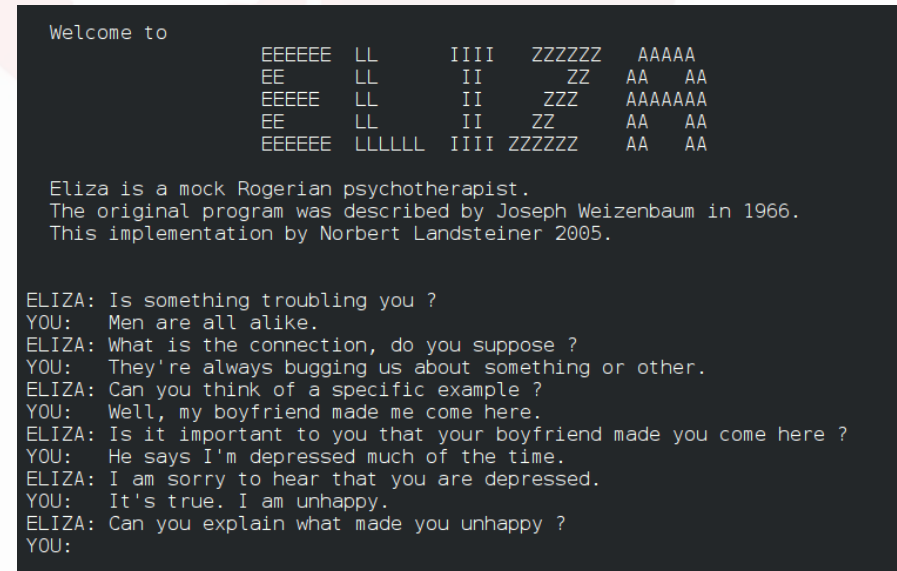
- Realiza ações textuais e coleta resultados
- Específico, criado manualmente
- Exemplos: Eliza, DRQN

Nível 2: LLM Agent

- Usa LLM para agir
- Exemplos: SayCan, LLM-Planner

Nível 3: Reasoning Agent

- Usa a LLM para raciocinar e agir
- Atual estágio onde nos encontramos



```
Welcome to
EEEEEE LL      IIII  ZZZZZZ  AAAAA
EE      LL      II    ZZ    AA  AA
EEEEEE LL      II    ZZ    AAAAAA
EE      LL      II    ZZ    AA  AA
EEEEEE LLLLLL  IIII  ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:
```

Fonte: Domínio Público

Exemplo: Q&A

P: Olá, como vai? R: Olá! Estou muito bem, obrigado por perguntar. →

Requer inferência simples

P: As aves de Maria põem 16 ovos por dia. Todos os dias ela come um no café da manhã e assa bolos, consumindo mais 4 ovos. Ela vende o restante por R\$ 0,50 cada ovo. Quanto ela ganha por semana? →

Requer raciocínio

P: Quem é o Papa atual? → **Requer conhecimento**

P: Qual é o quinquagésimo número da sequência de Fibonacci? →

Requer processamento



Estratégias para Solução de Problemas

Raciocínio “Simples”

P: As aves de Maria põem 16 ovos por dia. Todos os dias ela come um no café da manhã e assa bolos, consumindo mais 4 ovos. Ela vende o restante por R\$ 0,50 cada ovo. Quanto ela ganha por semana?

Etapas de Raciocínio:

The user is asking a word problem about calculating daily earnings. I need to break down the problem into steps:

1. Calculate the total eggs laid per day.
2. Calculate the eggs consumed/used by Maria.
3. Calculate the number of eggs remaining for sale.
4. Calculate the daily earnings from selling the remaining eggs.

This is a simple arithmetic problem, so a direct conversational answer will suffice.

Raciocínio “Simples”

P: As aves de Maria põem 16 ovos por dia. Todos os dias ela come um no café da manhã e assa bolos, consumindo mais 4 ovos. Ela vende o restante por R\$ 0,50 cada ovo. Quanto ela ganha por semana?

Etapas de Resposta:

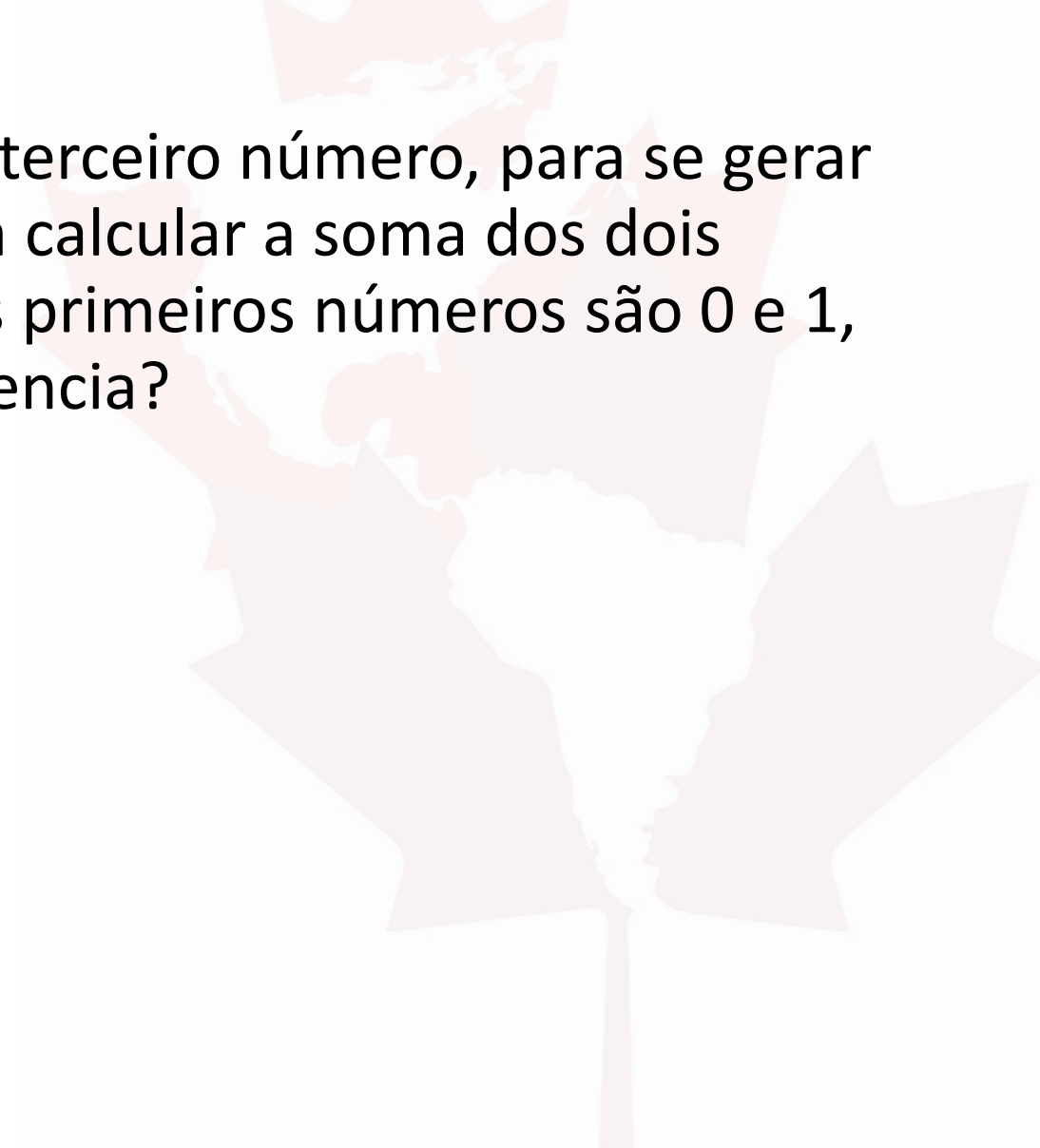
Maria ganha R\$ 5,50 por dia.

Vamos calcular:

1. Ovos postos por dia: 16 ovos
2. Ovos que ela usa/come: 1 (café da manhã) + 4 (bolos) = 5 ovos
3. Ovos restantes para venda: $16 - 5 = 11$ ovos
4. Quanto ela ganha por dia: $11 \text{ ovos} * \text{R\$ } 0,50/\text{ovo} = \text{R\$ } 5,50$

Processamento

P: Na sequência de Fibonacci, a partir do terceiro número, para se gerar um número qualquer da sequência, basta calcular a soma dos dois números anteriores. Sabendo que os dois primeiros números são 0 e 1, qual é o quinquagésimo número da sequência?



Processamento

R: Para encontrar o quinquagésimo número da sequência de Fibonacci (F_{49} , se $F_0 = 0$ for o primeiro termo), precisamos usar a regra de que cada número é a soma dos dois anteriores, começando com 0 e 1.

Calcular isso manualmente seria extremamente trabalhoso e propenso a erros. No entanto, podemos usar um programa simples para fazer isso de forma eficiente.

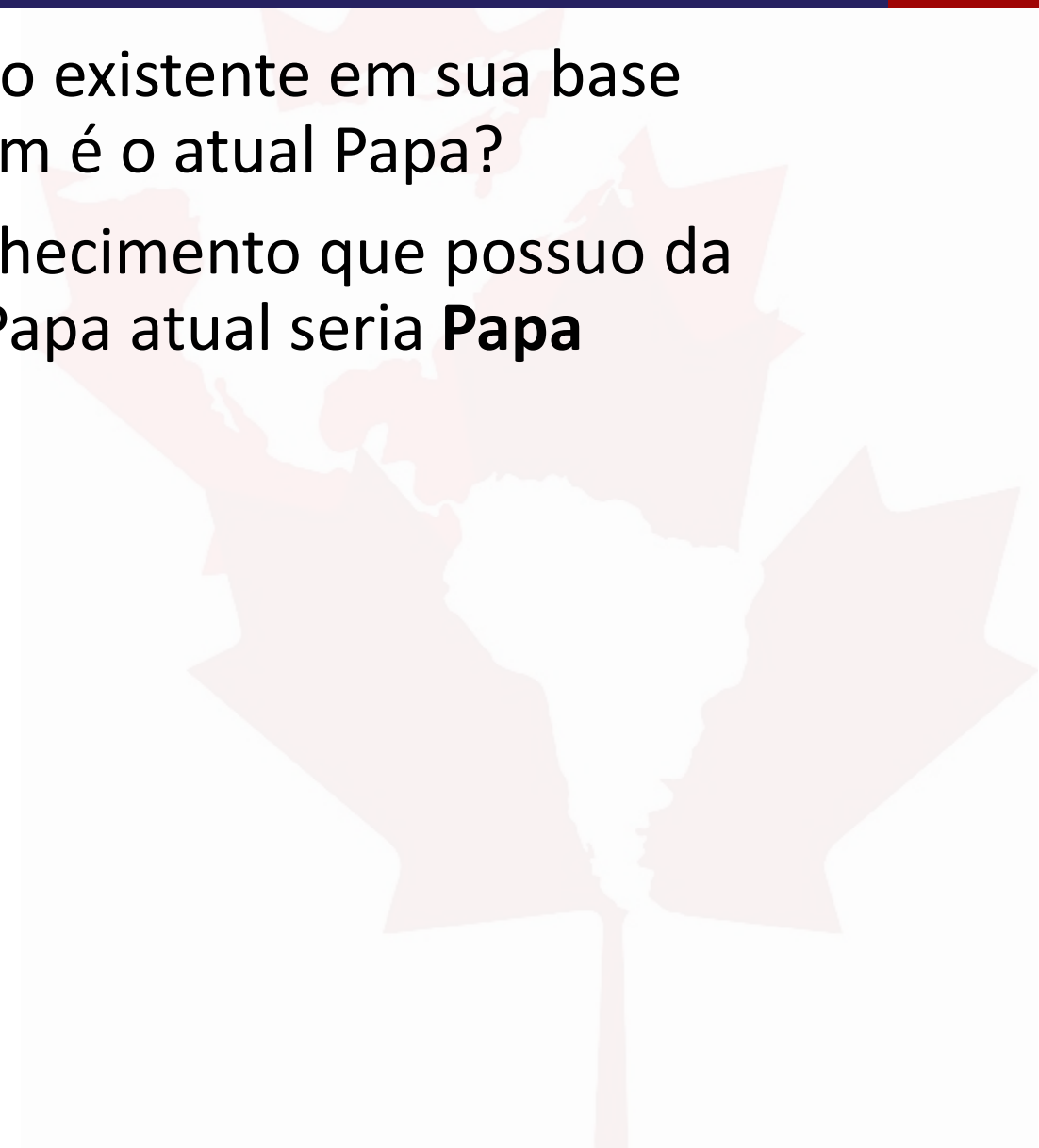
Vamos usar Python para calcular o F_{49} :

Executando o código, o quinquagésimo número da sequência de Fibonacci (F_{49}) é: 7778742049

```
def fibonacci(n):  
    # Condições iniciais  
    if n == 0:  
        return 0  
    elif n == 1:  
        return 1  
    else:  
        # Array para armazenar os números calculados  
        fib_sequence = [0, 1]  
        for i in range(2, n + 1):  
            next_fib = fib_sequence[i-1] + fib_sequence[i-2]  
            fib_sequence.append(next_fib)  
        return fib_sequence[n]  
  
# Queremos o quinquagésimo número.  
# Se  $F_0$  é o 1º,  $F_1$  o 2º, então  $F_{49}$  é o 50º.  
numero_fibonacci = fibonacci(49)  
print(f"O quinquagésimo número da sequência de Fibonacci ( $F_{49}$ ) é: {numero_fibonacci}")
```

Obtenção de Conhecimento Adicional

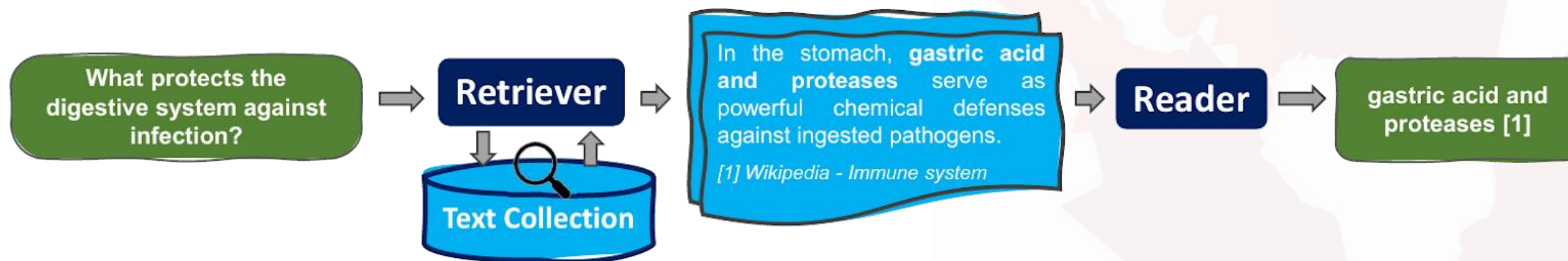
- P: Baseando-se apenas no conhecimento existente em sua base original de treinamento, responda: Quem é o atual Papa?
- R: Baseando-me exclusivamente no conhecimento que possuo da minha base de treinamento original, o Papa atual seria **Papa Francisco** (Jorge Mario Bergoglio).



RAG para Obtenção de Conhecimento Adicional

Respostas para perguntas de caráter intensivo sobre diferentes domínios de conhecimento

- “Corpora” adicional
- Uso de ferramenta de recuperação de dados



Fonte: <http://ai.stanford.edu/blog/retrieval-based-NLP/>

Mas... E se não existirem conjuntos com as informações desejadas?

Ferramentas

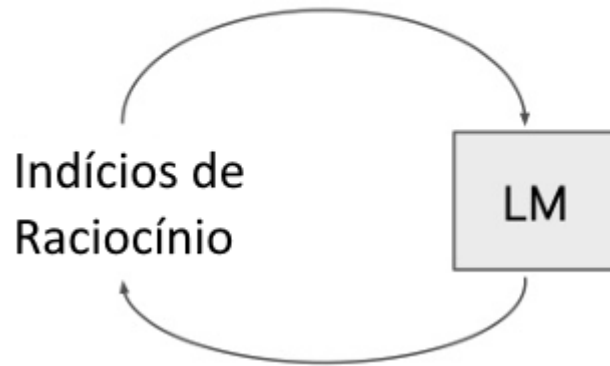
Elementos externos à LLM que permitem que seja executadas ações:

- Ferramentas de busca
- Execução de scripts
- Acesso a banco de dados
- Calculadora
- Uso de modelos específicos
- Uso do sistema de arquivos
- Etc.



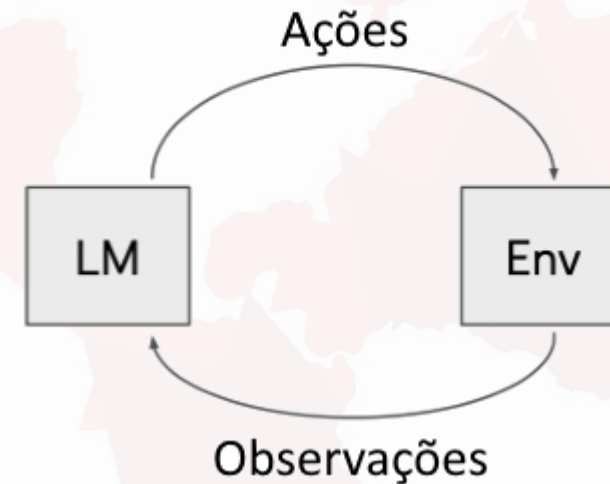
Raciocínio ou Ação?

CoT



- Flexível e generalizável
- Ausência de conhecimento externo e ferramentas

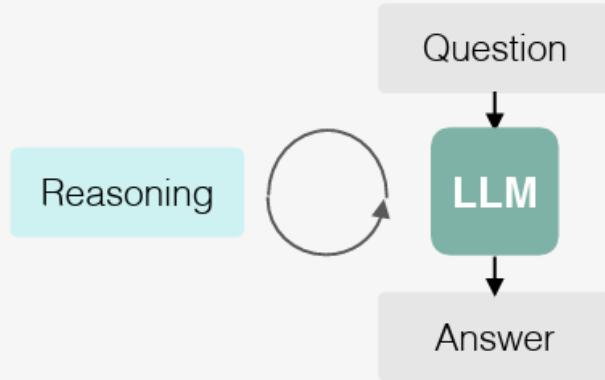
RAG/Retrieval/Code/Tool use



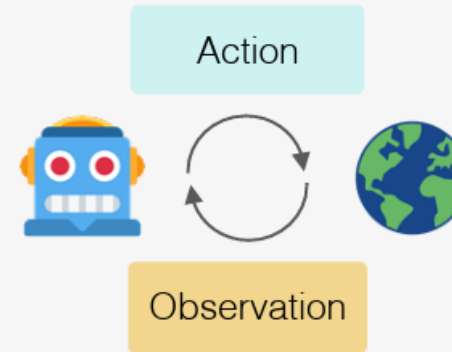
- Ausência de raciocínio
- Flexível e generalizável para conhecimento externo, cálculos, feedback, etc.

Raciocínio e Ação

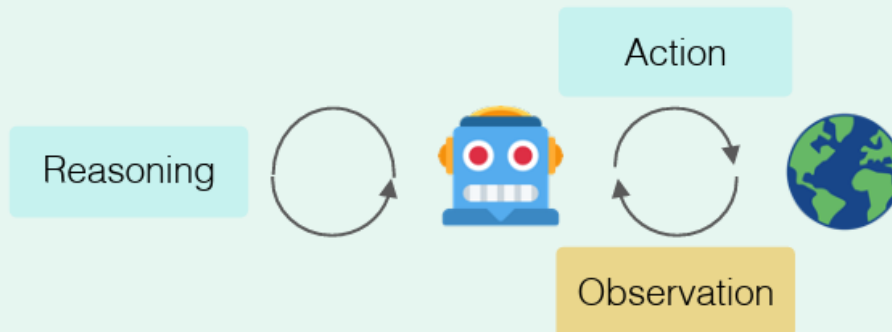
Reasoning (update internal belief)



Acting (obtain external feedback)



ReAct: a new paradigm of agents that **reason and act**



- **Synergy** of reasoning and acting
- **Simple** and intuitive to use
- **General** across domains

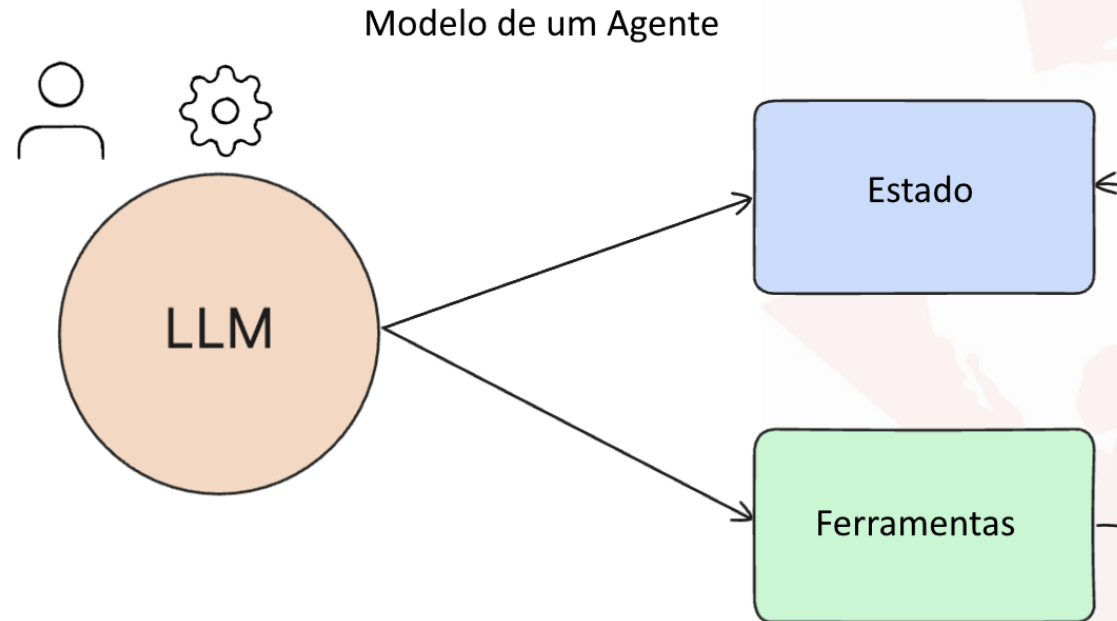
Obtenção de Conhecimento Adicional

- P: Quem é o atual Papa?
- R: Baseado nas informações mais recentes disponíveis, o Papa atual é **Papa Leão XIV**. Ele foi eleito sucessor em 8 de maio de 2025, após o falecimento do Papa Francisco em 21 de abril de 2025. O nome de batismo do Papa Leão XIV é **Robert Francis Prevost**, e ele é o primeiro Papa nascido nos Estados Unidos.
- P: Como você conseguiu esta informação?
- R: A minha capacidade de acessar informações mais recentes, como a eleição de um novo Papa em 2025, **vem da minha capacidade de me conectar e pesquisar informações atualizadas em tempo real na internet.**
- Quando a pergunta é feita de uma forma que requer conhecimento além da minha base de treinamento estática, eu sou capaz de utilizar **ferramentas de busca na web** para encontrar e apresentar a informação mais atualizada disponível. Foi assim que obtive o nome do Papa Leão XIV e os detalhes sobre sua eleição e origem.



Agentic AI: Agentes “Empoderados”

Agentes com Ferramentas




Fonte: https://github.com/microsoft/generative-ai-for-beginners/blob/main/17-ai-agents/images/what-agent.png?WT.mc_id=academic-105485-koreyst

LLM: Raciocínio e identificação da ferramenta que deve ser utilizada.

Estado: Contexto passado e presente. Mantém os resultados.

Ferramentas: Sistemas externos, banco de dados, APIs ou mesmo outro agente

Ferramentas

 **LangChain** [Integrations](#) [API Reference](#) [More](#) v0.3

Providers

- Anthropic
- AWS
- Google
- Hugging Face
- Microsoft
- OpenAI
- More


Components

- Chat models
- Retrievers
- Tools/Toolkits**
- Document loaders
- Vector stores
- Embedding models
- Other

All tools

Name	Description
ADS4GPTs	Integrate AI native advertising into your Agentic application.
AgentQL	AgentQL tools provides web interaction and structured data extraction...
AINetwork Toolkit	AI Network is a layer 1 blockchain designed to accommodate large-scal...
Alpha Vantage	Alpha Vantage Alpha Vantage provides realtime and historical financia...
Amadeus Toolkit	This notebook walks you through connecting LangChain to the Amadeus t...
Apify Actor	Apify Actors are cloud programs designed for a wide range of web scra...

Fonte: <https://python.langchain.com/docs/integrations/tools/>


 **n8n** [Product](#) [Use cases](#) [Docs](#) [Community](#) [Enterprise](#) [Pricing](#)

Connect anything to everything


[All Types](#) [Regular](#) [Trigger](#) [Core Nodes](#)

Categories **1054 integrations** [Sort: Popularity](#)

- ☒ All categories
- ☐ AI
- ☐ Analytics
- ☐ Communication
- ☐ Cybersecurity
- ☐ Data & Storage
- ☐ Developer Tools
- ☐ Development
- ☐ Evaluation
- ☐ Finance & Accounting
- ☐ HITL
- ☐ Marketing
- ☐ Miscellaneous
- ☐ Productivity
- ☐ Sales
- ☐ Utility




HTTP Request




Webhook

Webhooks are automatic notifications that apps send when something...




Google Sheets

Google Sheets is a web-based spreadsheet program offered by...




Telegram

Telegram is one of the fastest and most secured messaging apps on th...



Send Email



MySQL

The relational database management system MySQL is free and open...

Fonte: <https://n8n.io/integrations/>

Agentes com Ferramentas

The screenshot displays the Microsoft Azure Logic Apps Designer interface for a workflow named "Weather-Report". The workflow starts with a trigger "When a HTTP request is received", which connects to a "Weather agent" component. This agent is configured with two parallel actions: "Send email" and "Get weather". The "Send email" action is further configured with "Send an email (V2)", and the "Get weather" action is configured with "Get forecast for today". The right-hand pane shows the configuration for the "Weather agent", including the "Deployment Model Name" set to "gpt-4o-test1 (gpt-4o)", the "Current connection" as "fabrikam-azureopenai-connection", and "Instructions For Agent" and "System Instructions" text areas.

Microsoft Azure

Search resources, services, and docs (G+)

Copilot

Home > Agent-logic-app | Workflows >

Weather-Report | Designer

Workflow

Search

Run Save Discard Parameters Code view Errors AI Info

Tools

Designer

Code

Run history

Configuration

When a HTTP request is received

Weather agent

Send email

Get weather

Send an email (V2)

Get forecast for today

Weather agent

Parameters Channels Settings Code view About

Deployment Model Name *

gpt-4o-test1 (gpt-4o)

Current connection: fabrikam-azureopenai-connection

Instructions For Agent *

Add instructions so the agent understands its role and tasks. Include helpful information about workflow structure, restrictions, tools, and interactions in specific scenarios. [Tips for writing agent instructions](#)

System Instructions *

You're an AI agent that answers questions about the weather. You can also send weather information in email to a provided email address. If you're asked to send the weather in email, ask for an email address, if not provided.

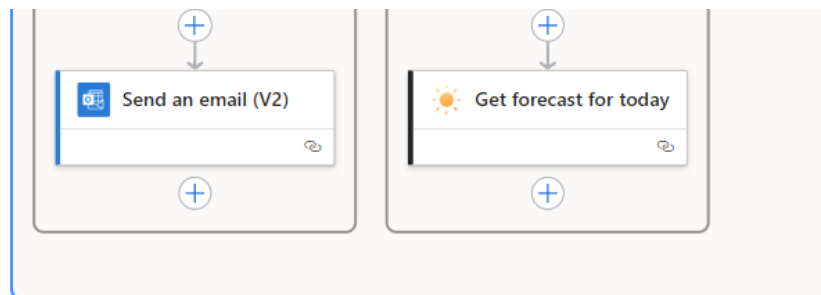
Format your response with bullet lists when appropriate. Make your responses concise and useful, but use a conversational and friendly tone. You can include suggestions like "Carry an umbrella" or "Dress in layers".

Agentes com Ferramentas

System Instructions *

You're an AI agent that answers questions about the weather. You can also send weather information in email to a provided email address. If you're asked to send the weather in email, ask for an email address, if not provided.

Format your response with bullet lists when appropriate. Make your responses concise and useful, but use a conversational and friendly tone. You can include suggestions like "Carry an umbrella" or "Dress in layers".



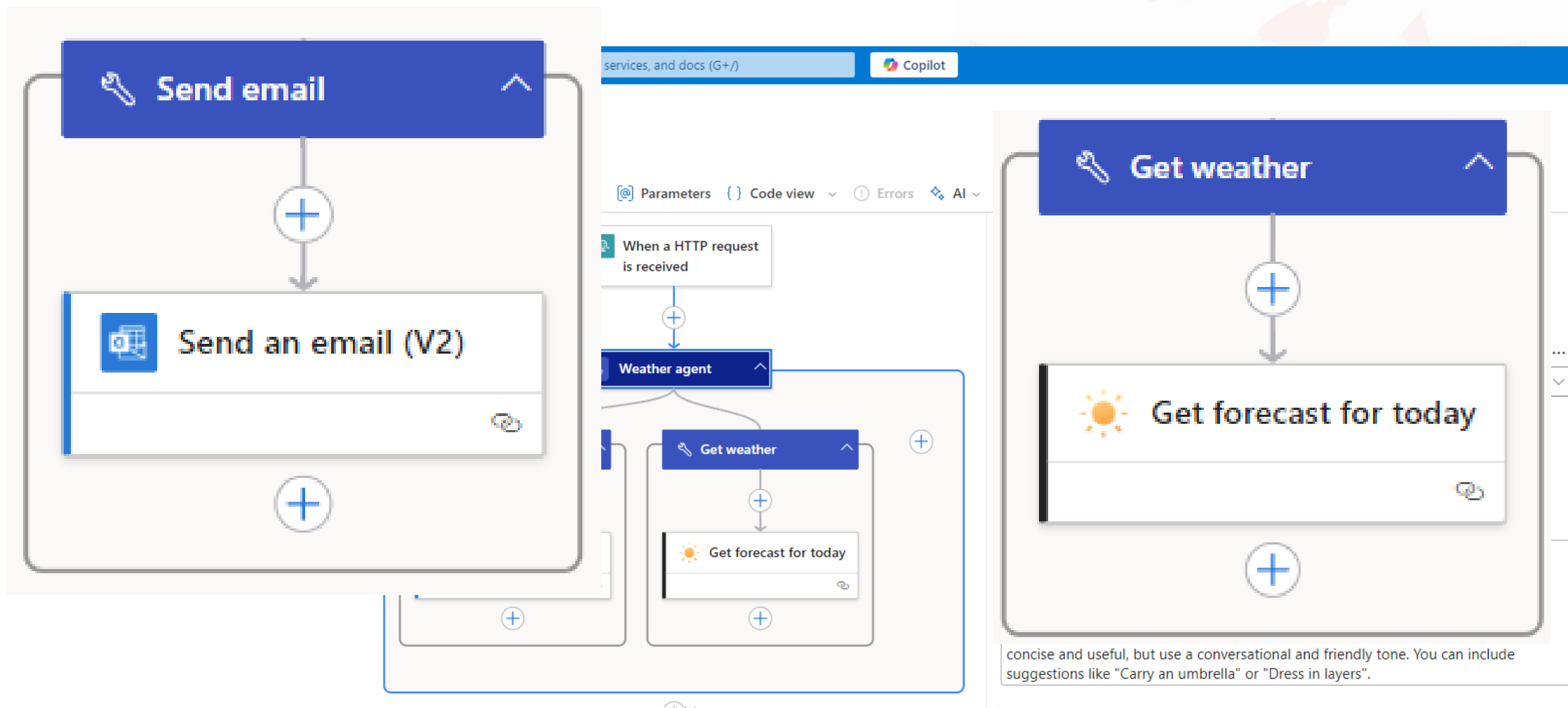
structure, restrictions, tools, and interactions in specific scenarios. [Tips for writing agent instructions](#)

System Instructions *

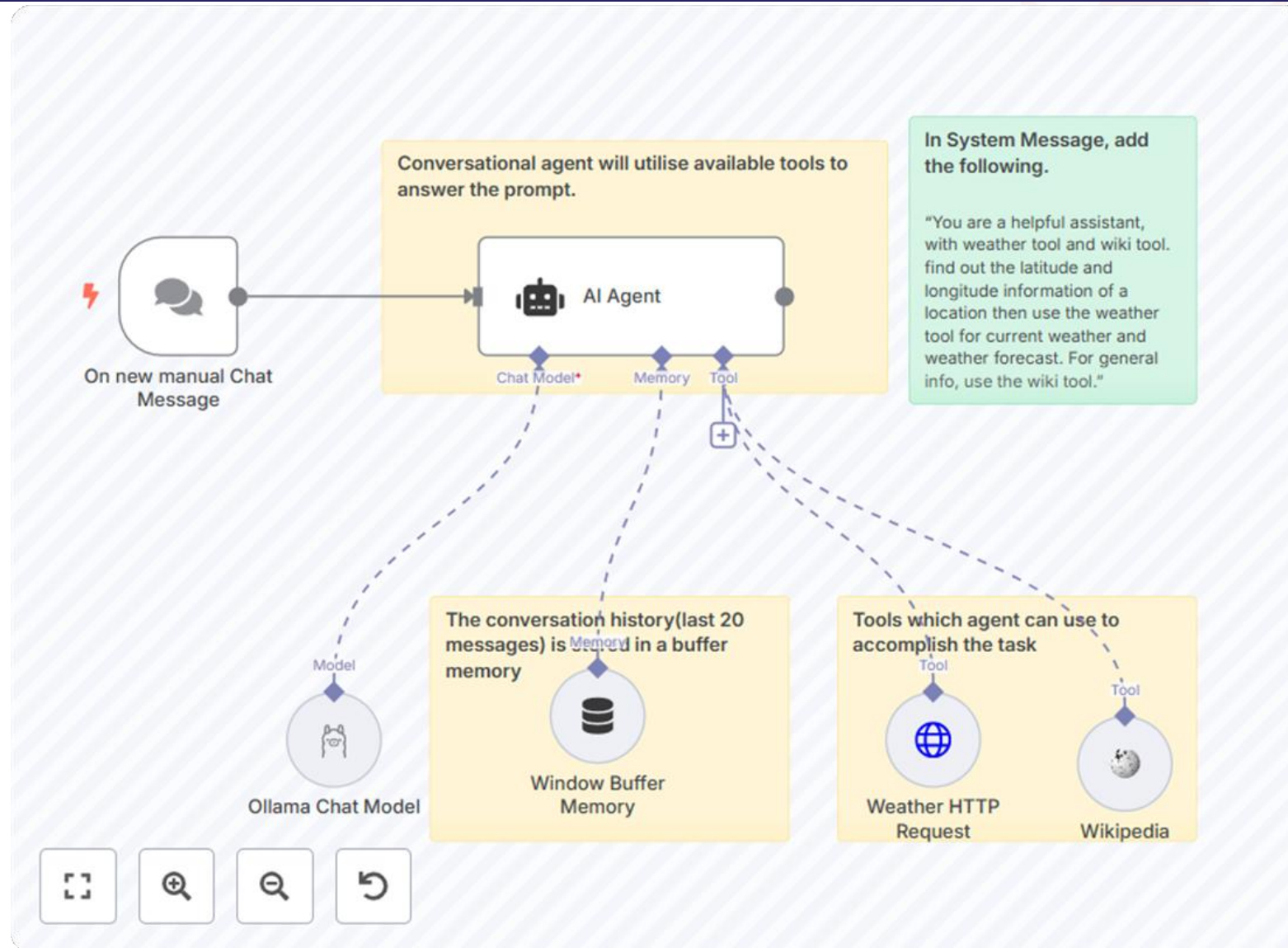
You're an AI agent that answers questions about the weather. You can also send weather information in email to a provided email address. If you're asked to send the weather in email, ask for an email address, if not provided.

Format your response with bullet lists when appropriate. Make your responses concise and useful, but use a conversational and friendly tone. You can include suggestions like "Carry an umbrella" or "Dress in layers".

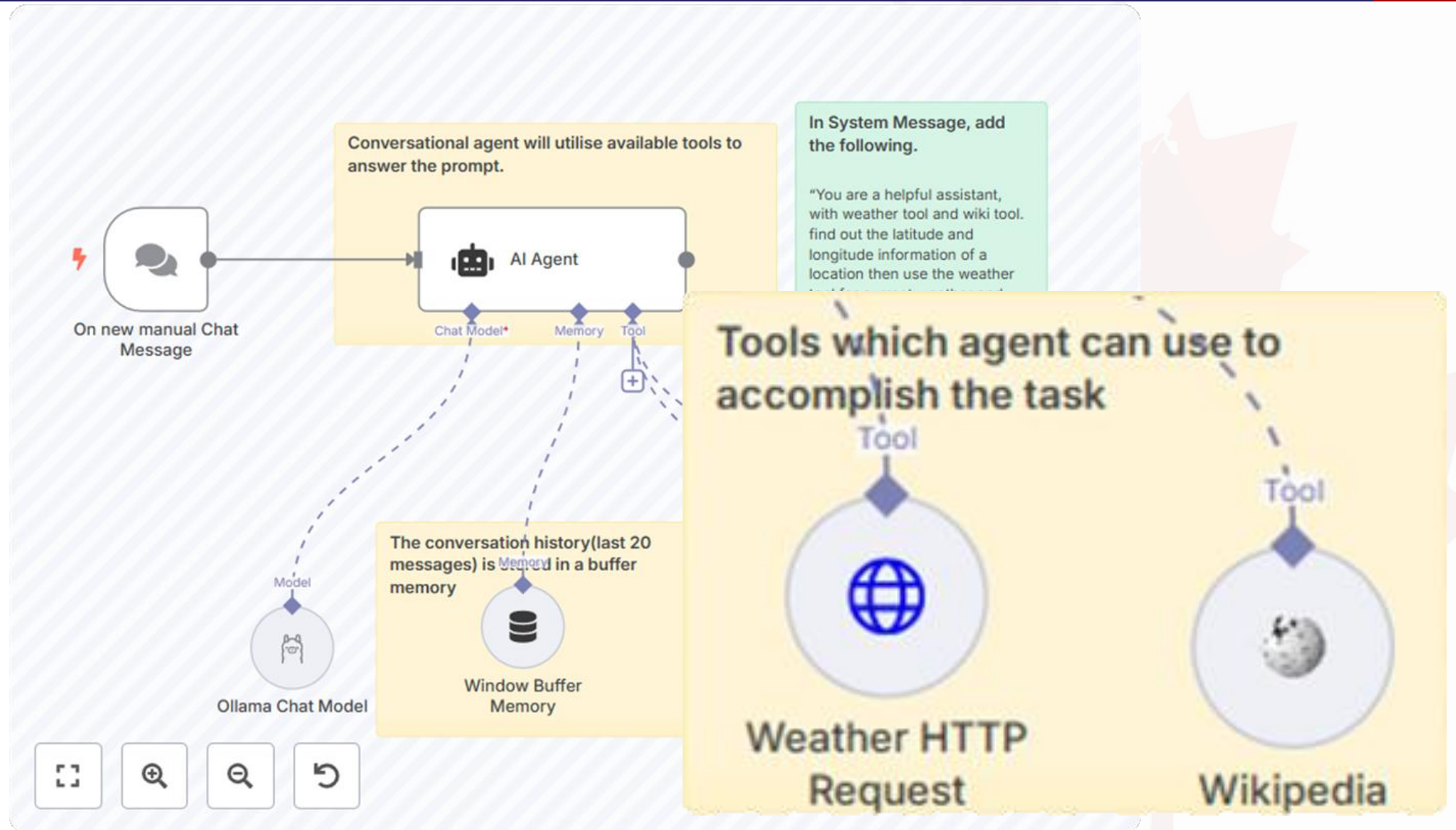
Agentes com Ferramentas



Agentes com Ferramentas

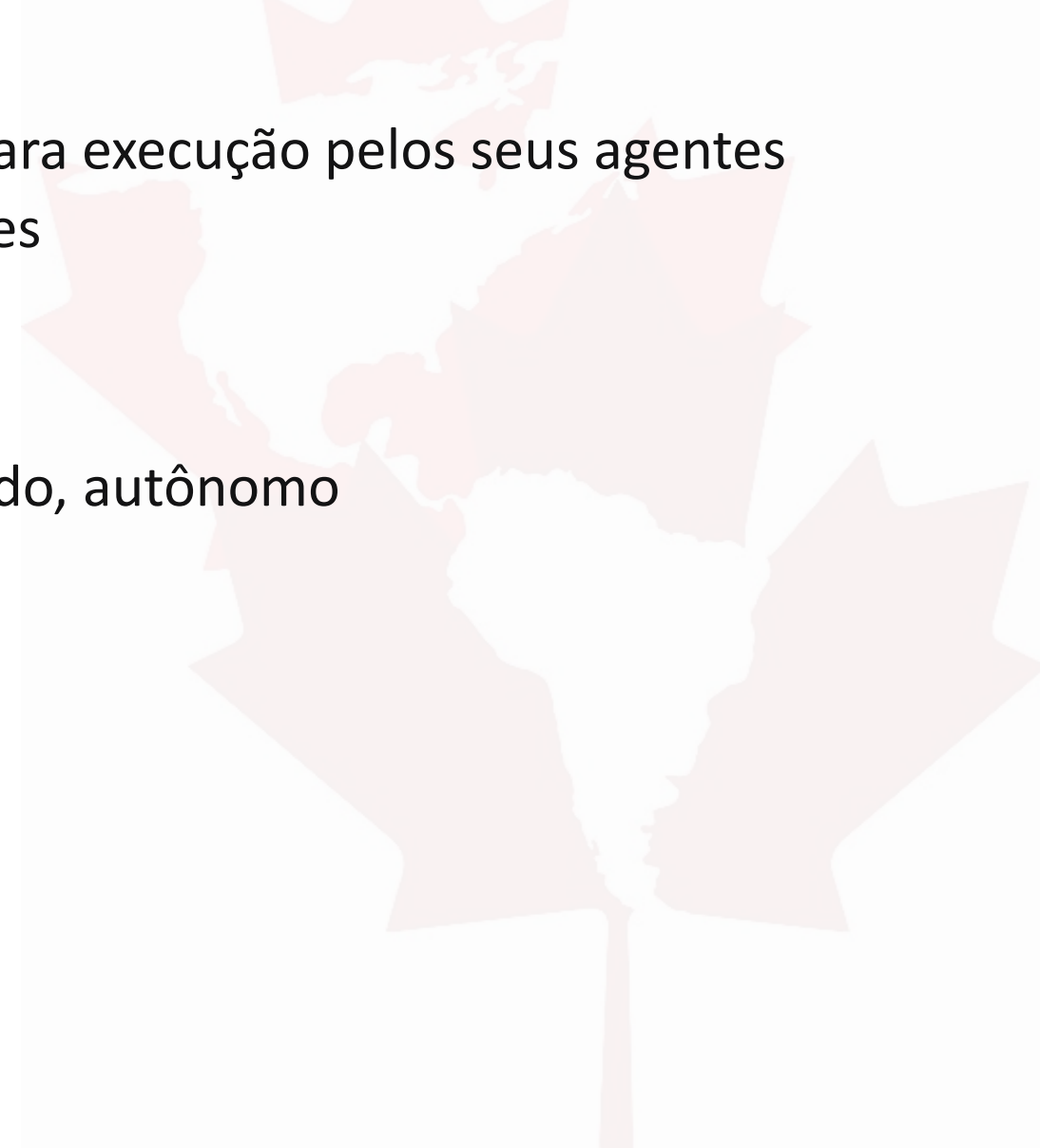


Agentes com Ferramentas



Planejando seus agentes

- Decomposição de Funções e Tarefas
 - Quebrar em tarefas menores adequadas para execução pelos seus agentes
 - Garantir que esteja alinhada com os agentes
 - Permitir supervisão humana
- Determinar nível de autonomia
 - Desde copiloto, colaborativo, supervisionado, autônomo
- Projetar as capacidades do(s) agente(s)
 - LLM + ferramentas
 - Quais Frameworks
 - Quais ferramentas
 - Como serão acessadas
 - MCP



Planejando seus agentes

- Planejamento da Interação
 - Sistemas multiagentes?
 - A2A?
- Incorporação da Capacidade de Adaptação
 - Seu agente poderá se adaptar?
 - Adaptação contextual: modificar o comportamento com base no contexto, como interações passadas ou condições inferidas
 - Adaptação Paramétrica: atualizando a relação funcional entre estados e ações
 - Adaptação Reflexiva: Raciocínio/reflexão mais profundos sobre como atualizar a relação funcional
- Planejamento para Monitoramento e Tratamento de Exceções
 - Desde copiloto, colaborativo, supervisionado, autônomo

Frameworks e Ferramentas

Principais Frameworks

- <https://autogenhub.github.io/autogen/>
- <https://ai.pydantic.dev/>
- <https://www.langchain.com/>
- <https://www.llamaindex.ai/>
- <https://www.crewai.com/>

E Ferramentas:

- <https://n8n.io/>
- <https://www.langflow.org/>



Autogen

O que é?

- Framework da Microsoft para orquestrar **múltiplos agentes de IA conversacionais** que colaboram entre si para resolver tarefas complexas.
- Permite a criação de sistemas de IA que se comunicam e se autoajustam, definindo papéis e capacidades para cada agente.

Vantagens:

- **Colaboração entre Agentes:** Habilidade de múltiplos LLMs (e outras IAs) interagirem para resolver problemas complexos.
- **Flexibilidade:** Permite definir diversos tipos de agentes (humano-em-loop, codificadores, etc.).
- **Automação de Fluxos de Trabalho:** Excelente para automatizar tarefas que exigem múltiplas etapas de raciocínio e execução.
- **Resolução Complexa:** Pode abordar problemas que um único LLM não conseguiria de forma eficaz.

Desvantagens:

- **Complexidade:** Projetar e depurar a interação entre múltiplos agentes pode ser desafiador.
- **Uso de Recursos:** Pode ser intensivo em recursos computacionais (múltiplas chamadas a LLMs).
- **Curva de Aprendizagem:** Requer uma compreensão do design de Agentic Systems.

Pydantic

O que é?

- Biblioteca Python para **validação e parsing de dados** usando *type hints* (anotações de tipo).
- Essencial para definir **estruturas de dados claras** e garantir que os dados recebidos (especialmente de LLMs) estejam no formato esperado.

Vantagens:

- **Validação Automática:** Garante que os dados estejam corretos e no tipo certo, reduzindo erros.
- **Seriação/Desserialização:** Facilita a conversão de objetos Python para JSON/dicionários e vice-versa.
- **Integração com LLMs:** Crucial para forçar LLMs a **gerar saídas estruturadas** (JSON, etc.), melhorando a confiabilidade das automações.
- **Documentação Implícita:** As anotações de tipo servem como documentação clara do modelo de dados.
- **Desempenho:** É otimizado para velocidade.

Desvantagens:

- **Python-centric:** Exclusivo para projetos em Python.
- **Sobrecarga para Casos Simples:** Pode ser um pouco excessivo para validações muito triviais.
- **Curva de Aprendizagem:** Para validações mais complexas ou customizadas.

LangChain

O que é?

- Framework abrangente para desenvolver aplicações alimentadas por LLMs.
- Oferece módulos para criar *chains* (sequências de chamadas a LLMs), *agentes* (LLMs com capacidade de usar ferramentas), *memória*, *prompts* e *ferramentas*.

Vantagens:

- **Modularidade:** Permite combinar componentes (LLMs, prompts, ferramentas, memória) de forma flexível.
- **Ecossistema Rico:** Grande variedade de integrações com LLMs, bancos de dados vetoriais, APIs e ferramentas.
- **Rápida Prototipagem:** Ajuda a construir rapidamente aplicações complexas de LLMs.
- **Comunidade Ativa:** Amplo suporte e muitos exemplos de uso.

Desvantagens:

- **Complexidade:** Pode ser excessivo para tarefas simples, e a quantidade de abstrações pode ser confusa inicialmente.
- **Overhead de Performance:** Às vezes, as abstrações podem introduzir uma pequena latência.
- **Evolução Rápida:** A API e a estrutura podem mudar frequentemente, exigindo atualizações constantes.

LlamaIndex

O que é?

- Framework focado em estruturar, indexar e recuperar dados externos para LLMs (principalmente para Geração Aumentada por Recuperação - RAG).
- Atua como uma camada de dados entre seus LLMs e seus dados privados ou em tempo real.

Vantagens:

- **Otimizado para RAG:** Excelente para construir sistemas de perguntas e respostas ou chatbots baseados em conhecimento específico.
- **Conectores de Dados:** Ampla gama de *loaders* para ingestão de dados de diferentes fontes (PDFs, APIs, bancos de dados, sites).
- **Estratégias de Indexação:** Suporta várias maneiras de indexar e armazenar embeddings para recuperação eficiente.
- **Abstração Complexa:** Facilita o tratamento de etapas complexas de processamento de dados para RAG.

Desvantagens:

- **Foco Principal em RAG:** Embora suporte agentes, seu ponto forte é a recuperação de dados, menos a orquestração agêntica geral comparado ao LangChain.
- **Intensivo em Memória/Recursos:** Para bases de conhecimento muito grandes, o processamento e a indexação podem exigir muitos recursos.
- **Curva de Aprendizagem:** Requer entendimento dos conceitos de embeddings, bancos de dados vetoriais e estratégias de recuperação.

Crewai

O que é?

- Um framework agêntico Python que permite criar "**equipes**" de **agentes de IA colaborativos** com papéis, objetivos e tarefas definidos.
- Inspirado no conceito de "agentes cognitivos" e focado em um fluxo de trabalho mais "humano-símile" para resolver problemas.

Vantagens:

- **Colaboração Intuitiva:** Designado para que os agentes trabalhem juntos de forma natural e estruturada.
- **Definição de Papéis/Tarefas/Processos:** Facilita a atribuição de responsabilidades e a definição do fluxo de trabalho.
- **Foco em Automação de Fluxos:** Ideal para tarefas que se beneficiam da divisão de trabalho e da iteração colaborativa.
- **Human-like Workflow:** Simula a interação entre membros de uma equipe humana.

Desvantagens:

- **Mais Recente:** Ainda em estágio de desenvolvimento mais inicial que LangChain/LlamaIndex, com uma comunidade menor.
- **Pode ser "opinionated":** Sua estrutura de "equipe" pode ser menos flexível para casos muito atípicos.
- **Desempenho:** O modelo de comunicação pode introduzir latência.

O que é?

- Ferramenta de **automação de fluxo de trabalho de código aberto** (self-hostable) que conecta APIs, serviços web e bancos de dados.
- Permite construir **automações visuais** (low-code/no-code) complexas, incluindo integrações com LLMs.

Vantagens:

- **Interface Visual:** Drag-and-drop para construir fluxos de trabalho, tornando-o acessível a usuários não-desenvolvedores.
- **Conectividade Extensa:** Centenas de integrações prontas para serviços e APIs populares.
- **Self-Hostable:** Permite rodar a ferramenta em sua própria infraestrutura para maior controle e privacidade.
- **Flexibilidade para LLMs:** Pode ser usado para orquestrar chamadas a LLMs e integrar suas saídas com sistemas externos (enviar emails, atualizar CRMs, etc.).
- **Customização:** Permite criar nós personalizados para necessidades específicas.

Desvantagens:

- **Menos Flexível que Código Puro:** Para lógicas extremamente complexas ou altamente otimizadas, o código puro ainda é superior.
- **Escalabilidade (Self-hosted):** Gerenciar a escala de uma instância self-hosted pode exigir conhecimento técnico.
- **Dependência de Nós Prontos:** Embora haja muitos, pode haver limitações se não houver um nó para uma integração muito específica.

LangFlow

O que é?

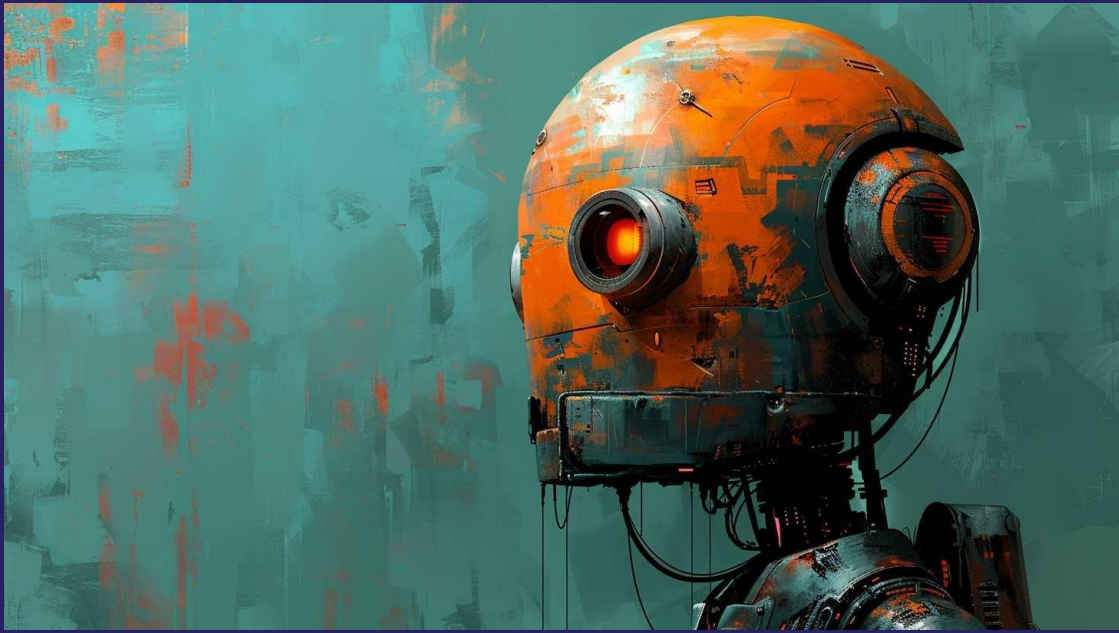
- **Interface de usuário visual (UI) para LangChain**, permitindo construir, testar e implantar cadeias e agentes de LLMs usando uma abordagem de drag-and-drop.
- Transforma o código do LangChain em um **fluxo de trabalho gráfico**, facilitando a visualização e prototipagem.

Vantagens:

- **Desenvolvimento Visual (Low-Code)**: Facilita a prototipagem e a construção de aplicações de LLMs sem muito código.
- **Depuração Visual**: Ajuda a entender o fluxo de dados e a depurar problemas em cadeias complexas.
- **Compartilhamento de Componentes**: Permite salvar e reutilizar fluxos e componentes.
- **Exploração Rápida**: Ótimo para experimentar diferentes prompts, modelos e ferramentas.
- **Acessibilidade**: Torna o desenvolvimento de aplicações LLM mais acessível para usuários com menos experiência em programação.

Desvantagens:

- **Dependência do LangChain**: Estritamente ligado ao ecossistema do LangChain; se você não usa LangChain, LangFlow não é relevante.
- **Abstração Excessiva**: Para desenvolvedores experientes, pode esconder detalhes importantes da implementação.
- **Performance**: Pode introduzir uma pequena sobrecarga em comparação com o código puro.
- **Foco na UI**: Não é um framework de IA por si só, mas uma ferramenta para construir com um framework existente.



Construindo Agentes

Mais um passo rumo ao seu projeto

Desafio 2 - Primeira entrega

A proposta do Projeto

- Cada grupo deverá gerar um relatório em formato PDF
- O arquivo do relatório deverá ter o nome:
<Nome_do_Grupo>_Proposta_de_Projeto.pdf
- Este relatório deverá conter:
 - Nome do Grupo
 - Integrantes do Grupo
 - Descrição do Tema Escolhido (o que se pretende fazer)
 - Público alvo (quem irá utilizar a solução proposta)
 - Justificativa do Tema Escolhido (por que isto é importante, o que agrega de valor ao público alvo)
 - Proposta preliminar de como o projeto será desenvolvido
 - Elementos adicionais: tabelas, gráficos, diagramas

Desafio 2 - Primeira entrega

- O grupo também deverá gerar uma apresentação de slides em formato de *pitch deck* para apresentação a uma banca examinadora com até 5 minutos de duração.
- O arquivo PPTX da apresentação deverá ter o nome: <Nome_do_Grupo>_Pitch_Deck_do_Projeto.pptx.
- Detalhes sobre pitches em <https://sebrae.com.br/sites/PortalSebrae/ufs/pe/artigos/pitch-deck-o-que-e-e-como-estruturar-o-seu,4787c413c7fd4710VgnVCM1000004c00210aRCRD>

Desafio 2 - Primeira Entrega

- **A entrega será por e-mail**
 - O responsável pelo grupo deverá enviar uma mensagem para o endereço challenges@i2a2.academy com os arquivos anexados (relatório em formato PDF e apresentação em formato PPTX), com cópia para si mesmo.
 - O título do e-mail deverá ser “**Agentes Autônomos – Projeto – Entrega 1**”.
 - Opcionalmente, o responsável pelo grupo poderá copiar os demais membros do grupo e desta forma gerar um “protocolo de entrega” adicional.
- O limite para entrega é **11/06/2025 às 23h59**.
- Esta atividade **É ELIMINATÓRIA**. A ausência da entrega elimina todos os participantes do **GRUPO**
- No dia 12/06/2025 teremos a apresentação de alguns grupos.
- Aguardamos que voluntários se manifestem.
- Quem desejar fazer sua apresentação, informe no corpo do e-mail.



Dall.e

<https://labs.openai.com/e/xYvsdZh7P3kY7OTbeD4RqdD6>

Aprofundando...

“Aumentando a experiência”

Em que vamos trabalhar?

- **Objetivo:**

- Criar um ou mais agentes que tornem possível a um usuário realizar perguntas sobre os arquivos CSV disponibilizados.
- Exemplo: Qual é o fornecedor que teve maior montante recebido? Qual item teve maior volume entregue (em quantidade)? E assim por diante.

- **Modo de operação:**

- Interface onde o usuário informa sua pergunta. O agente gerar a resposta.
- O(s) agente(s) deverá:
 - descompactar os arquivos,
 - selecionar o arquivo desejado
 - carregar os dados
 - fazer as queries
 - gerar a resposta para o usuário.

Como faremos?

- **Recursos:**

- Utilizar os arquivos csv disponibilizados no site/drive compartilhado

- **Tecnologias:**

- Scripts Python ou ferramentas LowCode/NoCode.
- Utilizar pelo menos 1 das frameworks/ferramentas a seguir:

- <https://autogenhub.github.io/autogen/>
- <https://ai.pydantic.dev/>
- <https://www.langchain.com/>
- <https://www.llamaindex.ai/>
- <https://www.crewai.com/>
- <https://n8n.io/>
- <https://www.langflow.org/>

- Este trabalho deve ser desenvolvido em **GRUPO**

Como será a entrega?

- **Gerar relatório em formato PDF descrevendo:**
 - 1. A framework escolhida
 - 2. Como a solução foi estruturada
 - 3. Pelo menos 4 perguntas com as respectivas respostas.
 - 4. Link para a pasta do Github contendo os códigos fonte ou um link para acessar seu agente.
 - 5. Não se esqueçam de ocultar chaves utilizadas nos softwares.
 - **Importante!** Não queremos que vocês obtenham as respostas de forma manual utilizando alguma LLM como o ChatGPT. Queremos que o(s) seu(s) agente(s) executem esta tarefa.
- **A entrega será por e-mail**
 - O responsável pelo grupo deverá enviar uma mensagem para o endereço challenges@i2a2.academy, com cópia para si mesmo, contendo o arquivo PDF anexado.
 - O título do e-mail deverá ser “**Agentes Autônomos – Análise de CSV**”.
 - Opcionalmente, o responsável pelo grupo poderá copiar os demais membros do grupo e desta forma gerar um “protocolo de entrega” adicional.
- O limite para entrega é **18/06/2025 às 23h59**.
- Esta atividade **É ELIMINATÓRIA. A ausência de entrega elimina o GRUPO.**

Perguntas?





<https://i2a2.academy>



Celso Azevedo
COO – I2A2

“May the force be with you”.



+55 16 99213-2650



celso@i2a2.academy



[/in/celso-augusto-morato-azevedo](https://www.linkedin.com/in/celso-augusto-morato-azevedo)