

Relatório 2 - Prática: Python: Criando um ReAct Agent do Zero

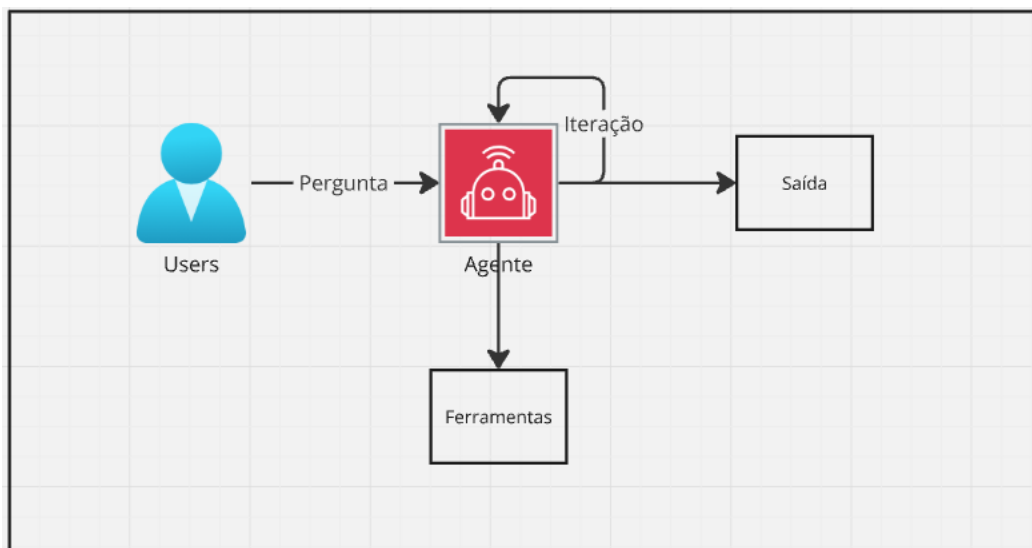
Paulo Victor Dos Santos

O minicurso em questão exibe de forma clara e concisa a criação de um agente auto interativo na linguagem Python. É abordado a arquitetura, os componentes, a criação da chave para utilização de LLM na plataforma Groq e os teste para execução da solução.

O vídeo do minicurso descreve o que é um agente e o processo de interação entre o usuário e o assistente. Através de uma cadeia de regras é demonstrado como o agente pode produzir um resultado e utilizar esse mesmo resultado como entrada ou parâmetro para o próximo prompt que também será executado no mesmo agente.

Em outras palavras, é notável que existe um processo de iteração, de laço de repetição e essa abordagem é implementada e testada com o objetivo de calcular a massa dos planetas do sistema solar. Além do processo iterativo, o minicurso deixa claro a importância da utilização e da personalização das ferramentas que podem interagir com o agente.

Abaixo segue o fluxograma do agente desenvolvido no minicurso.

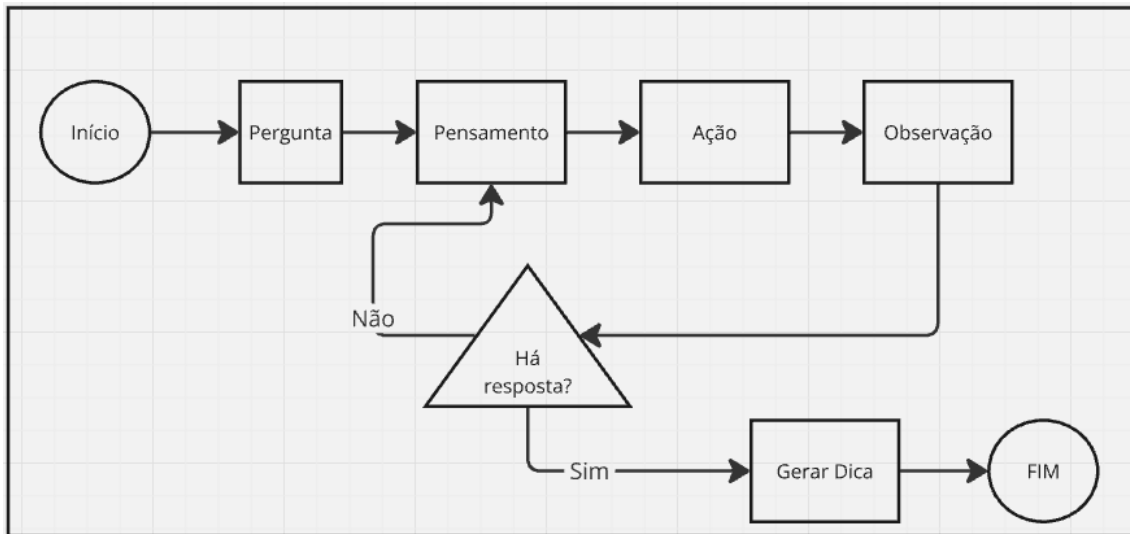


Diante do aprendizado adquirido foi implementado uma solução para cálculo de IMC – Índice de Massa Corpóreo. Com a mesmo tipo de agente exibido no minicurso, que é o agente React, esta solução integra o raciocínio juntamente com a tomada de decisão e execução de tarefas com utilização de ferramentas.

No caso da solução proposta, IMC, a linha de raciocínio se inicia com o usuário solicitando o seu IMC, informando peso e altura.

O agente então segue o raciocínio pré-estabelecido no prompt de sistema, executa as ferramentas de cálculo e classificação do IMC e finalmente responde e decide a dica a ser apresentada.

A solução proposta utiliza o seguinte fluxo:



Onde é proposto uma pergunta: “Qual meu IMC, meu peso é 80 quilos e minha altura é 1.70m”.

Essa pergunta então é processada e o estado atual será pensamento, seguido por ação e observação, onde o agente utiliza ferramentas também desenvolvidas até que o processo iterativo finalize com a resposta desejada e a dica gerada

Abaixo segue os resultados da solução desenvolvida no card 2:

```
iteracao(prompt="Me chamo Paulo, meu peso é 70.5 kilos e minha altura é 1,70 metros. Qual meu IMC?", verbose=False)
✓ [93] 2s 367ms

<Resposta>: O IMC do seu peso e altura está classificado como NORMAL, parabéns! Você está dentro do peso ideal.

<Dica>: Para manter o peso saudável, continue a praticar exercícios regulares e a ter uma alimentação balanceada!
```

Na imagem acima, um teste foi realizado informando os dados obrigatórios de peso e altura. Após a finalização do fluxo de estados do agente, a resposta é exibida juntamente com uma dica de saúde, também personalizada a depender do resultado gerado a partir da tabela de IMC.

O Código desenvolvido segue em anexo ao card.

Dificuldades

Desenvolvimento do prompt inicial e passagem de parâmetros para as ferramentas utilizadas pelo agente.

Conclusões

Esta tarefa é essencial para o entendimento do processo auto iterativo do agente. Fica claro que há uma cadeia de memória em formato de lista e que um único agente consegue interagir consigo mesmo. Fazendo questionamentos, capturando dados de entrada do usuário, invocando ferramentas, formatando os resultados das ferramentas e finalmente exibindo as possíveis respostas e dicas referentes ao IMC.

Referencias

<https://www.youtube.com/watch?v=hKVhRA9kfeM>

<https://console.groq.com/docs/quickstart>

<https://vidasaudavel.einstein.br/calcular-o-seu-imc/>

https://python.langchain.com/v0.1/docs/modules/agents/agent_types/react/

<https://medium.com/google-cloud/building-react-agents-from-scratch-a-hands-on-guide-using-gemini-ffe4621d90ae>