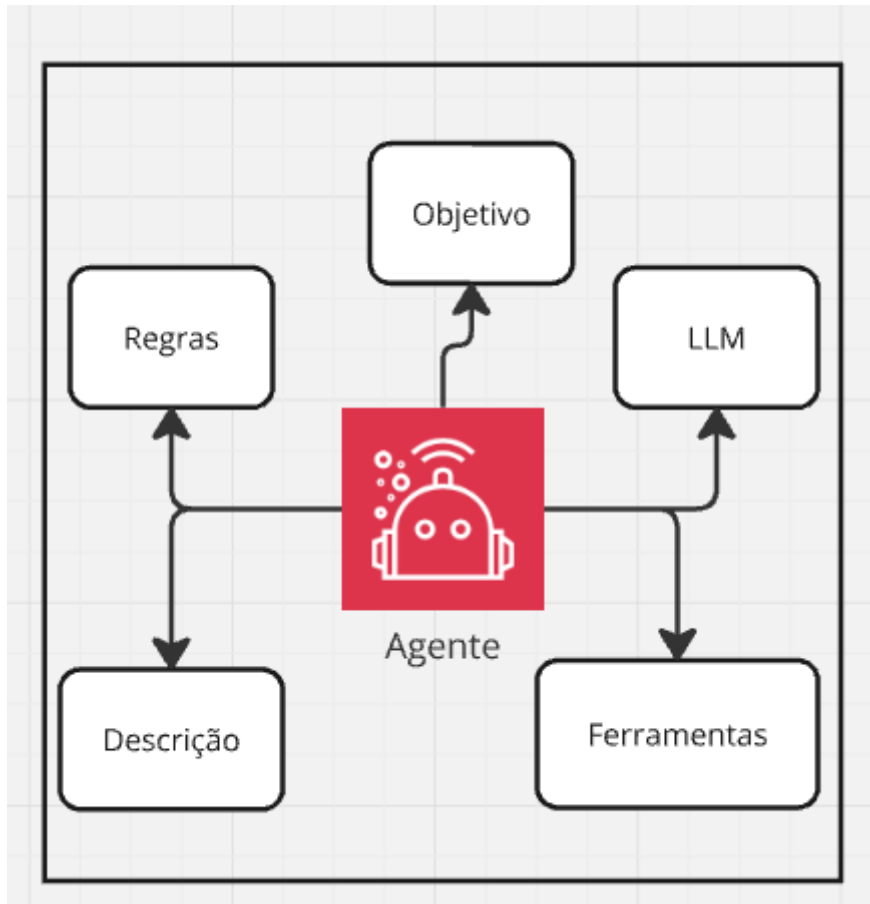


Relatório 3 – Mini curso de Agentes com Crew e Validação com Pydantic

Paulo Victor Dos Santos

O minicurso de agentes inteligentes da ferramenta Crew AI da plataforma DeepLearning.AI demonstra o uso da biblioteca de código aberto para automatizar a criação de agentes para facilitar com tarefas repetitivas. Destaca-se no curso que é necessário ter visão de gerente para definir o que cada agente deverá fazer, ou seja o objetivo, onde cada agente terá uma função clara, uma tarefa clara, ferramentas que irão auxiliar na execução daquelas tarefas e acesso a algum modelo de LLM.



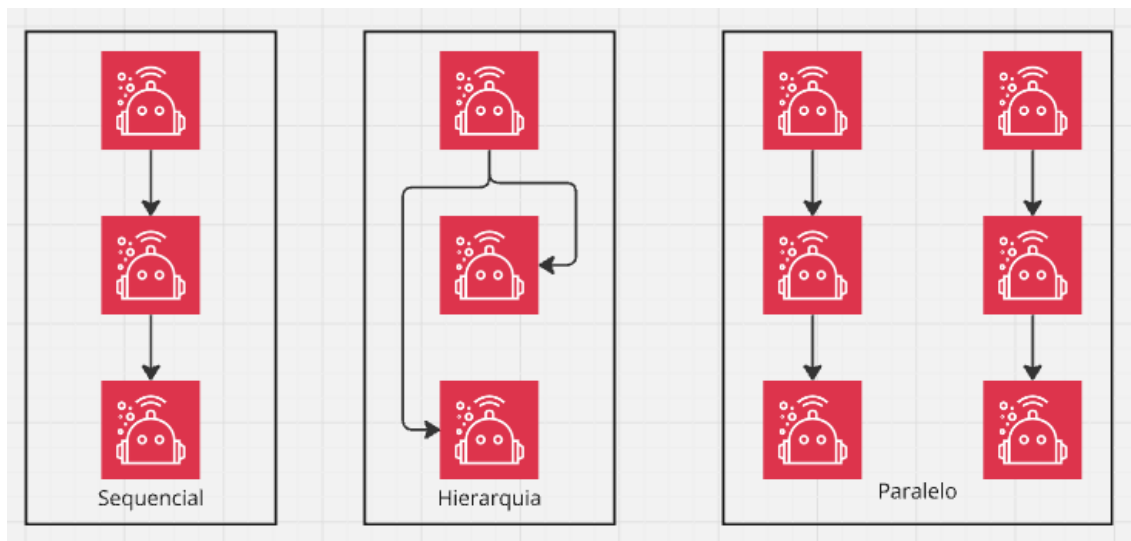
É demonstrado, além da fase de atribuir papéis especializados para cada agente, como manter memória nos agentes, essas memórias são úteis para que os agentes se lembrem do passado e possam manter uma certa coerência nas tomadas de decisão.

Em relação as ferramentas que cada agente pode utilizar, foi demonstrado as ferramentas já prontas que fazem parte do código aberto do CrewAI, além da demonstração de como personalizar essas ferramentas para cada tarefa em específico. Assim como há delegação ou não de agentes e tarefas, cada agente poderá executar uma ou mais ferramentas designadas para cada tarefa ou agente.

No curso também foi detalhado a necessidade de criação de guardrails, tanto para tratar possíveis falhas ou desvios que podem acontecer durante a execução de alguma tarefa. Esses erros podem ser falhas não previstas ou simplesmente respostas não esperadas. É importante ter a implementação de guardrails para poder ter resguardo de assuntos éticos e morais.

Outra importante funcionalidade do CrewAI é a opção da divisão de tarefas complexas, assumindo a postura de um gerente e tendo mão de obra para execução de certas tarefas, o

papel do gerente seria pensar em quem vai fazer o que e como deverá ser feito, para que no fim uma grande atividade esteja concluída com sucesso no tempo esperado. Essa mesma visão deve ser implementada nos agentes, onde há a opção de execução de tarefas sem série, em paralelo e até agentes cooperando entre si, os possíveis fluxos são:



Validação com Pydantic

O minicurso da ferramenta Pydantic é exibido como utilizar a ferramenta para fazer validação de dados básicos, como por exemplo, campos obrigatórios.

A biblioteca é desenvolvida em python e permite a personalização de objetos de negócio com validação inclusa, ou seja, qualquer classe de negócio a ser desenvolvida pode herdar as funcionalidades de uma base de modelo do pydantic e utilizar as validações necessárias para cada campo da classe de negócio.

A ferramenta já traz grande parte de validações comuns prontas, bastando utilizar, como por exemplo, validação de propriedades de senhas, de e-mail, de campos numéricos ou não. Além da possibilidade de usar decoradores em funções que podem ser chamadas antes de um evento ou após a execução de um determinado evento. Possibilitando a criação de fluxos dinâmicos onde uma determinada ação pode ocorrer ou não em detrimento de uma validação.

Na aula é exibido um caso de teste de validação de um objeto “usuário” e suas propriedades, é exibido detalhadamente quais propriedades podem ser validadas dentro do modelo ou separadas. Por fim, na aula é exibido como fazer a integração do pydantic com uma ferramenta de criação de API para integração de sistemas.

A integração do pydantic com FastAPI é feita de modo a simular a integração entre dois ou mais sistemas, onde no cenário hipotético um dos sistemas cria um “usuário” em outro sistema através da API. Diante disso é necessário que todas os dados do objeto usuário estejam de acordo com as regras do sistema de destino, assim sendo, o pydantic permite o lançamento de exceções para o sistema de origem para que as falhas de preenchimento sejam corrigidas.

Uma prova de conceito foi criada, simulando a interação entre dois sistemas da área médica. Um protocolo foi criado, a fim de garantir que um sistema converse com outro e possa disponibilizar arquivos de diferentes formatos e que essa mensagem possa ser enviada e recebida de forma rápida e validada.

Assim, dentro da prova de conceito, o objeto protocolo representa um “baseModel” do pydantic, pois herda as características necessárias para executar as validações já fornecidas pela solução e para a criação de validação personalizada.

Todo código da prova de conceito pode ser verificado no github disponibilizado no card 3.

Dificuldades

Não houve dificuldades com o card. Apenas paciência para entendimento do curso de CrewAI.

Conclusões

Concluo o Card 3 ressaltando a relevância do tema, principalmente da ferramenta pydantic e da forma como foi exibido no vídeo. Forma clara, objetiva e com exemplo real.

Esta ferramenta é essencial para criação de sistemas mais complexos e que precisam ter dados de qualidade, uma vez que dados com ruídos ou dados preenchidos de forma errada podem comprometer a qualidade das informações que possam geradas pelos agentes.

Diante disso, é essencial implementar validação nas pipelines de desenvolvimento, garantindo além da qualidade dos dados, a qualidade do sistema que foi desenvolvido para atender possíveis tarefas e objetivos específicos.

Referencias

<https://www.youtube.com/watch?v=502XOB0u8OY&t=46s>

<https://learn.deeplearning.ai/courses/multi-ai-agent-systems-with-crewai/lesson/ddys8/overview>

<https://docs.pydantic.dev/latest/>