

10 - Prática: Multi Agentes com CrewAI (III)

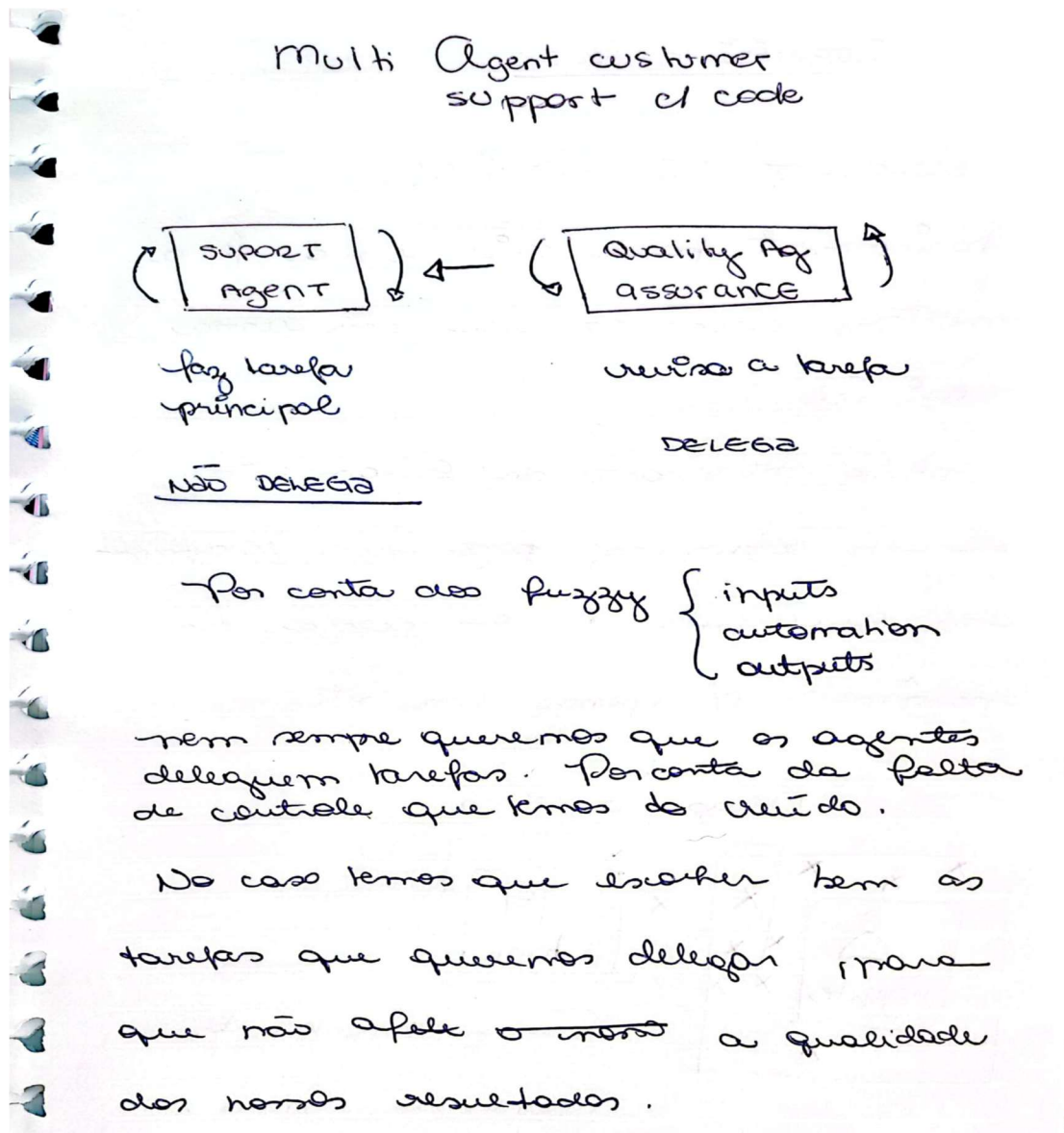
Suele Sousa

Descrição da atividade

O objetivo dessa atividade era continuar o aprendizado na plataforma do CrewAI focando no funcionamento do e construção de ferramentas.

Durante os vídeos apresentados o João Moura explica como construir uma equipe de agentes de suporte ao cliente, diferente do card anterior, nesse ele foca em explicar a delegação de tarefas, o uso da memória e também aprofunda nas ferramentas.

Na imagem abaixo podemos ver a explicação e como seria o funcionamento de um agente que tem permissão de delegar:

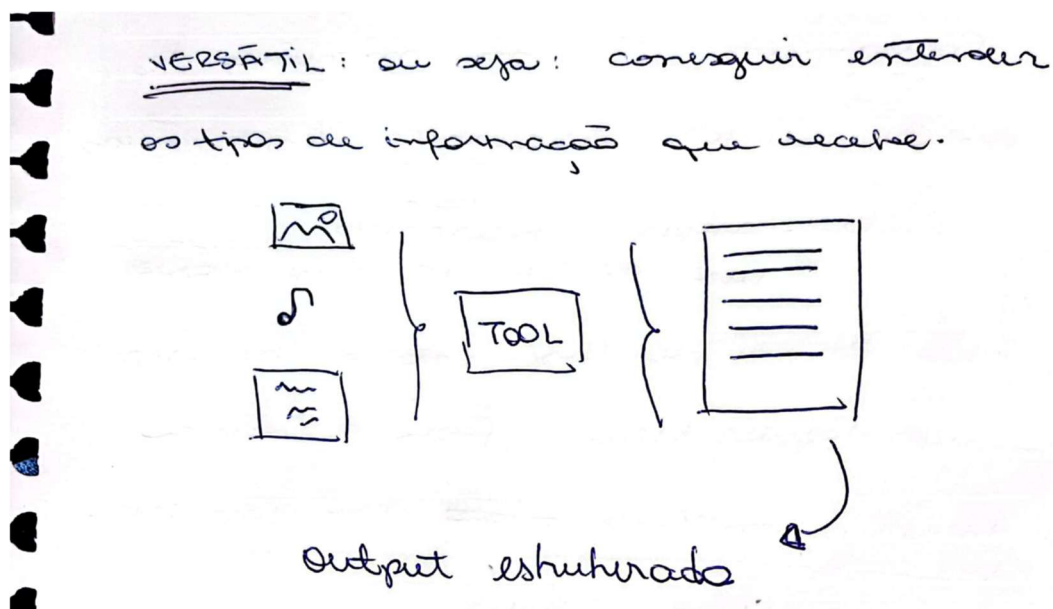


No caso acima, o agente de qualidade pode delegar para o agente de suporte, com isso, após checar a qualidade da resposta deste, pode pedir que ele reveja o conteúdo passado ou se aprofunde mais em algum tópico, por exemplo.

Além disso é explicado que as ferramentas têm uma certa priorização: o agente pode ter n ferramentas atribuídas, assim como a tarefa. Mesmo que o agente possua muitas ferramentas, durante a execução da tarefa, só poderá usar àquelas atribuídas na tarefa e não a ele próprio, por exemplo: O agente têm disponível 10 ferramentas, mas na tarefa, foi atribuído três ferramentas, então o agente só usará três ferramentas.

Assim como foi feito com os agentes, é abordado as melhores práticas na criação das tools, nas imagens abaixo há um resumo do vídeo sobre esses aspectos:

- **Versatilidade:**



- **Fault-Tolerance**

FAULT TOLERANT: Basicamente a ~~agente~~ tool deve ser capaz de ~~se~~ aprender e resolver quando a tool falha.

Basicamente: ~~dever~~ tentar novamente a tarefa, entender as exceções e o erro, aprender e se auto ajustar.

- Caching

- Caching

Basicamente é uma maneira de ~~fazer~~ evitar solicitações desnecessárias.

Evita: tarefas desnecessárias
uso de recursos desnecessários

A forma de fazer isso é, por exemplo,
dois agentes tentando fazer a mesma
tarefa, um deles ~~será~~ será impedido.
cross caching

No caso das ferramentas do Crewai, é utilizado esse cross-caching.

Pontos abordados sobre criação de equipes:

Memória:

Facilidade de utilizar memória quando estamos usando o Crewai, basicamente é ativar um parâmetro na função `crew`.

Memory

- Setting `memory=True` when putting the crew together enables Memory.

```
In [ ]: crew = Crew(  
    agents=[support_agent, support_quality_assurance_agent],  
    tasks=[inquiry_resolution, quality_assurance_review],  
    verbose=2,  
    memory=True  
)
```

Guardrails:

Através de código, é exemplificado como podemos criar um *guardrail* simples na hora fazer o input da solicitação, definindo quais as respostas esperadas, isso evita que o agente traga respostas alucinatórias ou confusas.

Guardrails

- By running the execution below, you can see that the agents and the responses are within the scope of what we expect from them.

```
In [ ]: inputs = {
    "customer": "DeepLearningAI",
    "person": "Andrew Ng",
    "inquiry": "I need help with setting up a Crew "
               "and kicking it off, specifically "
               "how can I add memory to my crew? "
               "Can you provide guidance?"
}
result = crew.kickoff(inputs=inputs)
```

No final do vídeo é apresentado ainda o funcionamento completo do agente criado, isso pode ser visto dentro do arquivo `aula_10_multi_agentes_crewai.ipynb` incluído na pasta do *github* do projeto, segue link:

Dica para a criação de agentes:

João Moura ainda destaca pontos importantes para a criação de agentes de forma efetiva: Pense em pessoas que você gostaria de contratar para a sua equipe, pessoas com experiência, background e um atuação profissional bem definidos.

Ainda há um pequeno resumo dos conceitos apresentados até o momento no curso:



- Agents "self improve" using memory
- Guardrails prevents agents from going into "rabbit holes" (unproductive, repetitive loops)
- Agents always attempt to get to an answer (avoid iterating indefinitely)
- Focus
 - narrowly defined task
 - specific agent roles and objectives
 - limited set of tools assigned to one agent

Dificuldades

A minha dificuldade principal foi que não consegui fazer o agente professor trazer todo o conteúdo que ele retirou do site dos estados.

Conclusões

A plataforma do CrewAI é uma plataforma que auxilia no desenvolvimento e criação de agentes de forma escalável, versátil, reutilizável e com a integração de memória, recurso que adiciona muito na criação de agentes. Além disso, a ferramenta de delegação de tarefas faz com que seja possível criar equipes inteiras que podem colaborar entre si.

Referências

Multi AI Agent Systems with crewAI. (n.d.). Retrieved from [https://learn.deeplearning.ai/courses/multi-ai-agent-systems-with-crewai/lesson/nk13s/multi-agent-customer-support-automation-\(code\)](https://learn.deeplearning.ai/courses/multi-ai-agent-systems-with-crewai/lesson/nk13s/multi-agent-customer-support-automation-(code)). Acesso em 27 abr 2025.

Multi AI Agent Systems with crewAI. (n.d.). Retrieved from <https://learn.deeplearning.ai/courses/multi-ai-agent-systems-with-crewai/lesson/fkfya/mental-framework-for-agent-creation>. Acesso em 27 abr 2025.

Multi AI Agent Systems with crewAI. (n.d.). Retrieved from <https://learn.deeplearning.ai/courses/multi-ai-agent-systems-with-crewai/lesson/c4j19/key-elements-of-agent-tools>. Acesso em 27 abr 2025.

Sivan, V. (2024). Building a multi agent system using CrewAI. Retrieved from <https://medium.com/pythoneers/building-a-multi-agent-system-using-crewai-a7305450253e>. Acesso em 27 abr 2025.