Relatório 3 - Prática: Validação de dados com Pydantic (I)

Suele Sousa

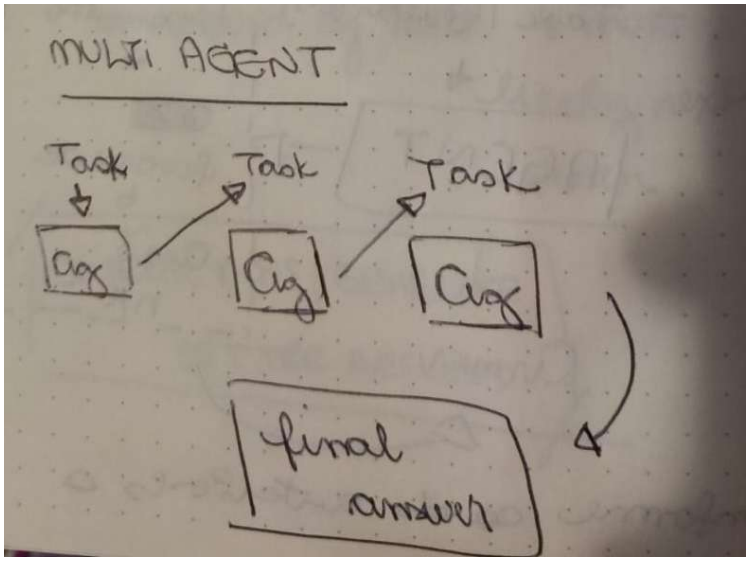
Descrição da atividade

Devemos assistir dois vídeos para entender qual o funcionamento do multi-agentes e também do tratamento de dados com o Pydantic.

Multiagentes basicamente são agentes trabalhando em cadeia, cada agente trabalha com o resultado do anterior até chegar ao resultado final esperado.

A vantagem de fazer isso é que ter agentes dedicados a resolver apenas a sua tarefa específica, ou seja, temos um “especialista”, mas diversos agentes trabalhando por um único objetivo, fazendo com que cada um performe muito bem no seu próprio objetivo e com isso o resultado final seja melhor.

Também é possível utilizar um modelo a cada agente, dessa forma, podemos colocar o melhor modelo para cada atividade e obter uma resposta mais completa.



Desenho esquemático de um multi agente

Pydantic

Pydantic é uma biblioteca que auxilia na validação de dados. No decorrer do vídeo de exemplo, pudemos acompanhar algumas formas de utilizar as bibliotecas. Os exemplos apresentados foram:

Exemplo 1: Criação de usuários e posteriormente utilizando o pydantic para validação dos dados inseridos.

Exemplo2: É uma melhoria do código anterior, incluindo a validação de campos específicos como a senha e user, utilizando regex.

Exemplo3: É uma melhoria do código anterior, incluindo a serialização de dados, que é uma forma de transformar os dados, facilitando sua transferência.

Exemplo4: Uma variação do código anterior, demonstrando a utilização da biblioteca Pydantic em conjunto com uma API, demonstrando a versatilidade da mesma.

Para consolidar os conhecimentos adquiridos fiz uma validação de CPF, para isso, foram necessárias realizar as seguintes checagens:

1. Mínimo e máximo de dígitos;
2. Caracteres numéricos;
3. Dígitos verificadores válidos (conforme exemplo encontrado no site https://dicasdeprogramacao.com.br/algoritmo-para-validar-cpf/)

Todas as validações, assim como a função e implementação podem ser encontrados no código prático, no github: https://github.com/SueleSousa/FastCampAgents/tree/main/card3

Dificuldades

Tive dificuldade para entender a lógica do último exemplo, acredito por se tratar de uma lógica diferente para a construção com a API, entendi um pouco melhor a criação das classes, mas continuo com um pouco de dificuldade, também por estar aprendendo novos elementos a cada nova tarefa.

Conclusões

Os vídeos do DeepLearning AI são muito fáceis de acompanhar e de compreender, o segundo vídeo do pydantic foi mais difícil porque não houve o desenvolvimento em conjunto conosco, apenas a implementação. Mas em geral ambos os vídeos cumpriram o propósito e a biblioteca pydantic é realmente útil para quem trabalha com dados.

Referências

Multi AI Agent Systems with crewAI. (n.d.). Retrieved from https://learn.deeplearning.ai/courses/multi-ai-agent-systems-with-crewai/lesson/eusjw/create-agents-to-research-and-write-an-article-(code). Acesso em 12 mar 2025.

Alves, G. F. de O. (n.d.). Algoritmo para Validar CPF. Retrieved from https://dicasdeprogramacao.com.br/algoritmo-para-validar-cpf/. Acesso em 12 mar 2025.
ArjanCodes. (n.d.). examples/2024/pydantic_refresh at main · ArjanCodes/examples. Retrieved from https://github.com/ArjanCodes/examples/tree/main/2024/pydantic_refresh. Acesso em 12 mar 2025.
Why You Should Use Pydantic in 2024 | Tutorial. ArjanCodes. Retrieved from https://www.youtube.com/watch?v=502XOB0u8OY&t=46s. Acesso em 12 mar 2025.