**Relatório 3 - Validação de dados com Pydantic (I)**

Guilherme Gueiros Ferreira

**1 Descrição da atividade**

Nesta atividade o foco foi compreender e aplicar a validação de dados utilizando Pydantic dentro de uma aplicação construída em FastAPI. Os notebooks de exemplo demonstraram as habilidades adquiridas a partir dos vídeos Multi AI Agent Systems with CrewAI (DeepLearning.AI, 2024) e Why You Should Use Pydantic in 2024 | Tutorial (YouTube, 2024), que apresentaram conceitos fundamentais sobre boas práticas de modelagem e validação. Já o notebook prático consolidou esse aprendizado com a implementação de uma API de produtos e a execução de testes automatizados. A validação de dados se mostrou essencial não apenas para garantir consistência e segurança nas requisições, mas também como um pilar para arquiteturas multi-agent, onde múltiplos agentes precisam trocar informações de forma confiável e sem ambiguidade.

**1.1 Conceitos abordados**

* Definição de modelos com BaseModel para estruturar dados de entrada.
* Uso de tipagem explícita (str, float, int, UUID4, datetime) para consistência.
* Aplicação de restrições com Field e validadores (field\_validator) para impor regras como tamanho mínimo de string ou formato de identificadores.
* Integração natural com FastAPI, onde a validação ocorre automaticamente nas requisições HTTP, retornando mensagens de erro padronizadas em caso de falha.

### 1.2 Realização

Com base nesses conceitos, desenvolvi uma API básica para cadastro e consulta de produtos, a API foi validada com testes simulados com get e post, contemplando um caso de sucesso e um de falha para cada cenário, organizados da seguinte forma:

* Cadastro e duplicação: inclusão de produto válido e tentativa de cadastrar o mesmo produto novamente (erro 409).
* Nome válido e inválido: verificação de cadastro correto e erro ao enviar nome vazio (erro 422).
* Dados corretos e tipo errado: aceitação de preço numérico válido e rejeição de valor textual (erro 422).
* Busca existente e inexistente: retorno bem-sucedido de produto válido e erro ao consultar UUID inválido (erro 422).

Essa prática possibilitou validar, na prática, a importância de Pydantic no tratamento de entradas para a API..

**2 Dificuldades**

As principais dificuldades enfrentadas foram compreender como funcionava a estrutura necessária para criação da API, desde a organização inicial até a forma de integrar os modelos do Pydantic.

Algo que não considero exatamente uma dificuldade, mas sim um ponto que encarei como essencial, foi entender como utilizar corretamente os parâmetros de validação no Pydantic, explorando a criação de métodos e personalização das regras para atender aos requisitos de cada campo.

**3 Conclusões**

A realização desta atividade permitiu consolidar o entendimento de como o Pydantic trabalha no desenvolvimento de aplicações em API, fornecendo mecanismos robustos de validação de dados. A prática demonstrou que a definição de modelos claros e a utilização de validadores personalizados são fundamentais para garantir a consistência das entradas e a clareza das respostas retornadas pela API.

Durante os testes implementados percebi a importância de verificar os mais diversos cenários de sucesso e falha, garantindo ao máximo que a aplicação esteja preparada para lidar com diferentes tipos de entrada de forma previsível e confiável.

Outro ponto relevante foi conseguir entender a importância dessas técnicas, quando bem aplicadas em arquiteturas multi-agent, nas quais múltiplos agentes trocam informações constantemente, nesse contexto, a validação de dados não apenas evita erros de execução, mas também se torna essencial para a comunicação entre agentes, reduzindo ambiguidades e garantindo integridade nas interações.

**4 Referencias**

DeepLearning.AI. *Multi AI Agent Systems with CrewAI* [online]. DeepLearning.AI, 2024. Disponível em:<https://learn.deeplearning.ai/courses/multi-ai-agent-systems-with-crewai/lesson/ddys8/overview>

YouTube. *Why You Should Use Pydantic in 2024 | Tutorial* [vídeo]. YouTube, 2024. Disponível em:<https://www.youtube.com/watch?v=502XOB0u8OY>

Documentação oficial do Pydantic. *Pydantic Documentation* [online]. 2024. Disponível em: <https://docs.pydantic.dev>

Documentação oficial do FastAPI. *FastAPI Documentation* [online]. 2024. Disponível em: <https://fastapi.tiangolo.com>