

Relatório de Experiência de Aprendizado

Durante este desafio da DIO, tive a oportunidade de explorar **automação de testes unitários** usando **IA aplicada ao desenvolvimento Python**, consolidando conceitos de **LangChain, Azure OpenAI e pytest**.

O objetivo do projeto era criar um **agente inteligente** capaz de receber um trecho de código Python e gerar automaticamente um arquivo de testes unitários, seguindo padrões de boas práticas de teste.

Experiência Técnica

- **LangChain + Azure OpenAI:** Aprendi a criar fluxos de agentes que recebem entradas específicas e retornam saídas estruturadas.
- **Automação de testes:** Entendi como a IA pode acelerar a criação de testes unitários, gerando funções `test_*` com casos de sucesso e falha automaticamente.
- **Integração com pytest:** Testei os arquivos gerados e validei que funcionam corretamente com pytest, reforçando o conceito de **Test Driven Development (TDD)**.
- **Boas práticas de projeto:** Organizei o código em módulos claros e documentei o passo a passo de configuração, incluindo variáveis de ambiente e chaves de API do Azure OpenAI.

Experiência Teórica

- Refleti sobre como **IA pode apoiar desenvolvedores**, reduzindo tarefas repetitivas como escrita de testes unitários.
- Reforcei a importância da **qualidade de código e cobertura de testes**, mesmo em pequenos trechos de código.
- Compreendi melhor o conceito de **prompt engineering**, criando instruções precisas para que o agente gerasse testes corretos e robustos.

Mapa de Aprendizado

1. Testes Unitários e pytest

- Documentação oficial do pytest
<https://docs.pytest.org/en/stable/>
- Boas práticas em testes unitários Python
<https://realpython.com/python-testing/>

2. LangChain e Agentes de IA

- LangChain Docs
<https://www.langchain.com/docs/>
- Exemplo de agentes com LLM
<https://www.langchain.com/docs/use-cases/agents/>

3. Azure OpenAI

- Azure OpenAI Overview
<https://learn.microsoft.com/en-us/azure/cognitive-services/openai/overview>
- Prompt Engineering no Azure
<https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/prompt-engineering>

4. Boas práticas de projeto e GitHub

- Markdown Guide
<https://www.markdownguide.org/>
- GitHub Docs
<https://docs.github.com/>

Conclusão

Este projeto reforçou que **IA não substitui o desenvolvedor**, mas **potencializa sua produtividade**, especialmente em tarefas repetitivas como escrita de testes unitários. Além disso, documentar cada etapa e estruturar o repositório adequadamente foi essencial para facilitar o uso e entendimento do projeto.