📘 Relatório de Experiência de Aprendizado

Durante a realização deste desafio proposto pela DIO, tive a oportunidade de unir teoria e prática em um tema altamente atual: **análise de ameaças de arquitetura de software com Python, FastAPI e Azure OpenAI**.

A jornada começou com a **compreensão da metodologia STRIDE**, um modelo de ameaças amplamente utilizado em segurança de software. Entendi que:

- **S (Spoofing)**: ameaça relacionada à falsificação de identidade.
- **T (Tampering)**: modificação não autorizada de dados ou componentes.
- **R (Repudiation)**: impossibilidade de negar ações realizadas.
- **I (Information Disclosure)**: vazamento de informações confidenciais.
- **D (Denial of Service)**: indisponibilidade de serviços.
- **E (Elevation of Privilege)**: aumento indevido de privilégios de acesso.

Além disso, a parte prática do desafio me mostrou como combinar **Python + FastAPI** para construção de APIs modernas e escaláveis, e como o **Azure OpenAI** pode ser usado para aplicar **prompt engineering** e transformar uma simples imagem (um diagrama de arquitetura) em um relatório técnico de segurança.

Outro ponto de aprendizado foi sobre o **desenho arquitetural** em si: compreendi a importância de registrar visualmente como os componentes se conectam, e como esses diagramas servem de base para análises de segurança.

Experiência Técnica

- Aprendi a estruturar um projeto com **FastAPI**, configurando rotas, upload de arquivos e integração com serviços externos.
- Explorei como o **Azure OpenAI** pode ser integrado a soluções personalizadas, utilizando **chaves de API e variáveis de ambiente** para segurança.
- Reforcei boas práticas de documentação com **README.md** e versionamento no **GitHub**.

Experiência Teórica

- Entendi que **segurança não é um módulo extra**, mas sim parte essencial do ciclo de vida do software.
- Apliquei a **metodologia STRIDE** em exemplos práticos, treinando o raciocínio crítico para identificar ameaças potenciais.
- Reforcei a importância da **engenharia de prompts**: a forma como construímos perguntas e instruções impacta diretamente na qualidade das respostas do modelo de IA.

📸 Exemplos de Uso

1. Upload de imagem de arquitetura via endpoint /upload.
2. API processa a imagem e envia para análise no Azure OpenAI.
3. Retorno: relatório de ameaças baseado na metodologia STRIDE.

🗺️ Mapa de Aprendizado

1. Fundamentos de Segurança e STRIDE

- Microsoft Threat Modeling (STRIDE)

  https://learn.microsoft.com/en-us/security/compass/threat-modeling-tool-threats

- OWASP Threat Modeling

  https://owasp.org/www-community/Threat_Modeling

2. Desenvolvimento com Python + FastAPI

- FastAPI Documentation

  https://fastapi.tiangolo.com/

- Pydantic

  https://docs.pydantic.dev/

- Uvicorn

  https://www.uvicorn.org/

3. Inteligência Artificial com Azure OpenAI

- Azure OpenAI Service

  https://learn.microsoft.com/en-us/azure/cognitive-services/openai/overview

- Prompt Engineering Guide

  https://learn.microsoft.com/en-us/azure/ai-services/openai/concepts/prompt-engineering

4. GitHub & Documentação Técnica

- Markdown Guide

  https://www.markdownguide.org/

- GitHub Docs

  https://docs.github.com/

✅ Conclusão

Este projeto foi uma excelente oportunidade para unir **segurança da informação, IA aplicada e desenvolvimento backend**. A maior lição aprendida foi que **documentar bem o processo é tão importante quanto codar**: só assim conseguimos compartilhar conhecimento e mostrar nossa evolução profissional.