Relatório: Consulta de Temperatura por Localização

Nome: Mariana dos Santos Nicola

RM: 561698

Turma: 1TDCPG

Disciplina: Coding for Security

Data: 13/08/2025

Estrutura do projeto

O projeto foi desenvolvido com foco na modularização e clareza, utilizando uma estrutura baseada em funções reutilizáveis. A ideia principal era obter a localização aproximada do usuário com base em seu IP, utilizando APIs externas, e proteger informações sensíveis através do uso de variáveis de ambiente, armazenadas com segurança em um arquivo .env.

A biblioteca python-dotenv foi adotada para lidar com a leitura dessas variáveis, assegurando que dados como chaves de API não ficassem expostos no código-fonte. Além disso, o projeto foi executado dentro de um ambiente virtual Python, o que contribuiu para o isolamento e organização das dependências.

Durante o desenvolvimento, o código foi comentado linha por linha e aprimorado com mensagens informativas para o usuário em caso de erros ou falhas de execução.

Dificuldades encontradas

Durante o desenvolvimento deste projeto, enfrentei algumas dificuldades que contribuíram para o meu aprendizado prático em Python. Uma das principais situações ocorreu com o uso do arquivo .env: inicialmente, o posicionei dentro da pasta do ambiente virtual (venv), o que impediu que as variáveis de ambiente fossem carregadas corretamente. Esse erro causava falhas na execução da aplicação, especialmente ao tentar acessar a API de geolocalização, e foi resolvido após investigar o motivo e entender que o arquivo .env deve estar na raiz do projeto, fora da pasta do ambiente virtual.

Outro desafio importante foi o tratamento de exceções. Embora já tivesse familiaridade teórica com try e except, identificar exatamente onde implementar os blocos de tratamento para oferecer mensagens informativas ao usuário exigiu atenção e prática. Além disso, foi necessário revisar a documentação das bibliotecas utilizadas (como requests e dotenv) para entender seu comportamento em diferentes cenários de erro.

Também tive dificuldades iniciais ao configurar corretamente o ambiente virtual e instalar os pacotes necessários com o pip, principalmente ao diferenciar comandos como *pip install* dentro e fora do ambiente virtual. Por fim, integrar múltiplas APIs, testar diferentes comportamentos e garantir que o fluxo do programa fosse compreensível para o usuário, com mensagens claras e funcionais, exigiu paciência, testes e aprimoramento constante.

Apesar dessas barreiras, todos esses obstáculos me ajudaram a consolidar habilidades práticas importantes, como depuração, organização de código e leitura de documentação técnica, essenciais para o desenvolvimento de soluções mais robustas em Python.

Funcionalidades Implementadas

- Obtenção automática da geolocalização do usuário por meio do IP.
- Tratamento de exceções com mensagens explicativas no terminal.
- Leitura de variáveis de ambiente sensíveis com segurança via `.env`.
- Organização do projeto com comentários explicativos para cada linha.
- Execução em ambiente virtual para evitar conflitos de dependências.
- Utilização de IA para refatoração, explicações e apoio técnico.
- Leitura e interpretação das documentações oficiais das bibliotecas utilizadas.

Observações finais

Este projeto foi bastante desafiador e exigiu um comprometimento maior do que eu inicialmente esperava. A complexidade do código e a necessidade de integrar diversas funcionalidades, algumas delas avançadas, tornaram o desenvolvimento um processo que demandaria mais do que as poucas horas disponíveis em sala de aula para ser concluído com qualidade.

A realização deste projeto em casa foi fundamental para que eu pudesse dedicar o tempo necessário para planejar, testar e implementar as funcionalidades com calma

e atenção aos detalhes. Trabalhar em um ambiente tranquilo e sem pressa me permitiu compreender melhor o funcionamento das APIs externas, assim como aprofundar o entendimento sobre o uso correto de variáveis de ambiente e tratamento de erros em Python.

Essa flexibilidade possibilitou identificar e corrigir erros importantes, como a má localização do arquivo .env, que impactava diretamente no funcionamento do programa, além de aprimorar as mensagens de feedback para o usuário, tornando o sistema mais amigável e robusto.

Além disso, o processo de desenvolvimento estimulou a pesquisa cuidadosa das documentações oficiais das APIs Ipstack e OpenWeatherMap, bem como a adoção de boas práticas no código, como o uso de ambiente virtual e modularização.

Por fim, a integração do uso da inteligência artificial para revisão e otimização do código mostrou-se uma ferramenta valiosa, acelerando a resolução de dúvidas e a melhoria da qualidade do programa. Essa experiência reforçou a importância da organização, paciência e curiosidade para resolver desafios técnicos com eficiência.

Abaixo, estão algumas fontes de consulta que auxiliaram a construção do trabalho:

- https://pypi.org/project/python-dotenv/
- https://docs.python-requests.org/en/latest/
- https://docs.python.org/3/tutorial/errors.html
- https://ipstack.com/documentation
- https://openweathermap.org/api/one-call-3
- https://www.hashtagtreinamentos.com/previsao-do-tempo-com-python