**Case Técnico – Automação de Notas**

**Objetivo**

Desenvolva um script em Python que automatize o processo de leitura de chaves de acesso de notas fiscais a partir de arquivos .txt e organize os arquivos resultantes em uma estrutura de pastas segmentadas por filial e data.

**Requisitos Técnicos:**

Implemente as seguintes funcionalidades no script:

1. **Acesse a pasta de entrada:**

* Navegue até a pasta previamente definida que contém os arquivos .txt no caminho:  
  ./CHAVES NOTAS/\*.txt
* Leia os arquivos .txt, que foram previamente renomeados com o **código da filial** (Ex.: CHAVES FILIAL 04.txt, CHAVES FILIAL 05.txt, ...).
* Extraia as chaves de acesso de cada linha do arquivo (formato padrão: **44 dígitos numéricos** por linha).

1. **Baixe os XMLs e DANFEs:**

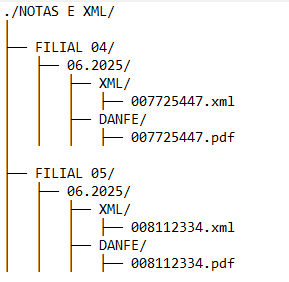
* Para cada chave, acesse o site **https://www.meudanfe.com.br**, realize a consulta e baixe o XML correspondente.
* Em seguida, utilize a API do site no **https://www.meudanfe.com.br/documentacao-api** para gerar a DANFE em PDF.

1. **Organize os arquivos:**

* Salve os arquivos XML e PDF em uma estrutura de pastas com o seguinte formato:  
  ./NOTAS E XML/<filial>/<AAAA-MM-DD>/
* Dentro de cada diretório de data, crie as subpastas XML e DANFE para armazenar os respectivos arquivos.
* Renomeie os arquivos XML e PDF utilizando o número da nota, no seguinte formato:  
  **007725447.xml**, **007725447.pdf**

1. **(Opcional) Implemente exceções de erros:**

* Implemente tratamento de erros para os seguintes casos:
  + Chaves inválidas
  + Falhas de conexão com o site
  + Requisições mal sucedidas
* Gere logs com o resumo do processamento e possíveis falhas.
* Salve os logs de cada execução no diretório:  
  ./LOGS/
* Renomeie o arquivo de log com o seguinte padrão:  
  processamento\_YYYY-MM-DD.txt (Ex.: processamento\_2025-06-23.txt)

**Estrutura de pastas e arquivos que deverá se parecer ao final da execução:**  


**Bibliotecas sugeridas:**

* os, glob, re: para manipulação de arquivos e pastas (sugestão: **os**)
* requests, selenium ou pyppeteer: para interação com o site (sugestão: **selenium**)
* xml.etree.ElementTree ou lxml: para leitura de dados do XML (sugestão: **xml.etree.ElementTree**)
* logging: para geração e armazenamento de logs