Visão Geral

Este projeto realiza a automação da extração de dados de uma tabela em um site e a análise de imagens de faturas via OCR. Etapas:

- Acessar dinamicamente uma tabela em uma página web utilizando **Selenium**;
- Baixar imagens vinculadas à tabela;
- Utilizar **Tesseract OCR** para extrair informações relevantes das imagens;
- Aplicar expressões regulares para identificar datas e códigos de fatura;
- Gerar um arquivo CSV com os dados tratados.

Estrutura do Projeto

Funções principais:

- **apagar_arquivos**(): Remove a pasta faturas/ e o arquivo resultado.csv para garantir um ambiente limpo a cada execução.
- **baixar_imagem**(url): Baixa imagens das faturas a partir das URLs extraídas da tabela e armazena localmente.
- **ocr_imagem**(caminho_imagem): Realiza OCR utilizando Tesseract para extrair texto da imagem. Em seguida, aplica regex para capturar o número da fatura e a data presente na imagem.
- **regex_data**(texto): Extrai datas presentes no texto em dois formatos - “yyyy-mm-dd” e formato por extenso (e.g. Jun 1, 2019)
- **regex_fatura**(texto): Extrai códigos de fatura presentes no texto com base em um padrão pré-definido (# CODIGO).
- **obter_valores_tabela**(): Utiliza Selenium para abrir a página e extrair dados da tabela HTML usando JavaScript injetado no navegador.
- **criar_csv**(lista): Gera o arquivo resultado.csv com os dados processados.
- **main**(): Coordena toda a execução, desde a preparação do ambiente, extração e download de imagens, até o processamento e geração do CSV final.

Decisões Técnicas e Otimizações

Extração via JavaScript + Selenium:

Ao invés de depender apenas do dom do HTML tradicional, foi utilizado JavaScript para forçar o carregamento completo da tabela com 12 linhas por página, garantindo uma extração mais eficiente, precisa e única dos dados.

Paralelismo com ThreadPoolExecutor:

A etapa de download de imagens foi otimizada com o uso de “ThreadPoolExecutor”, que permite baixar múltiplas imagens simultaneamente, reduzindo significativamente o tempo total de execução.

OCR com Pytesseract:

O Tesseract foi escolhido por sua precisão, flexibilidade e facilidade de integração com Python, possibilitando a conversão de imagem para texto com ótima performance.

Uso de Regex para dados específicos:

A extração de datas e códigos foi feita com expressões regulares para tornar o reconhecimento mais confiável mesmo com variações no conteúdo das imagens.

Como Executar

Pré-requisitos (Se necessário, altere conforme o seu ambiente.):

- Python 3.x
- Google Chrome instalado (e compatível com o ChromeDriver)
- Tesseract OCR instalado: digi.bib.uni-mannheim.de/tesseract/

Passos de Execução

1. **Clone o repositório:**
    - “git clone https://github.com/ThIaGoOLuiZz/Teste_Tecnico”
    - “cd Teste_Tecnico”

2. **Instale as dependências:**
    - “pip install -r requirements.txt”

3. **Execute o script principal:**

    - “python main.py”

Resultado Esperado

Ao final da execução:

- A pasta faturas/ será criada contendo todas as imagens baixadas.
- Um arquivo resultado.csv será gerado na raiz do projeto com as colunas:
    - id
    - Data_Vencimento
    - Data_Fatura (extraída da imagem)
    - Fatura (número da fatura extraído da imagem)