Relatório de Implementação -Sistema de Feedback 360° com Análise de Soft Skills

O objetivo deste projeto foi desenvolver um sistema simples de coleta e análise de feedback 360° para equipes, com foco em soft skills (comunicação, empatia, liderança). O sistema permite que os usuários insiram notas e comentários para cada integrante da equipe, gera um relatório estruturado e salva os dados em arquivo. Toda a solução foi containerizada utilizando Docker.

Estrutura do Projeto

```
feedback360/

— app.py # Código principal do sistema (CLI interativo)
— requirements.txt # Dependências Python
— Dockerfile # Dockerfile para containerização
— relatorio.csv # Relatório gerado (após execução)
— README.md # Instruções e informações do projeto
```

Passo a Passo da Implementação

- Desenvolvimento do Programa
 - O programa foi desenvolvido em Python, utilizando a biblioteca pandas para manipulação de dados.
 - O arquivo principal, app.py, implementa uma interface de linha de comando (CLI) que:
 - o Pergunta o número de pessoas na equipe.
 - Para cada pessoa, solicita notas de 1 a 10 para soft skills (comunicação, empatia, liderança) e um comentário.
 - o Ao final, gera um relatório em formato CSV com todas as informações coletadas.

Criação do requirements.txt

O arquivo requirements.txt foi criado com o seguinte conteúdo:

→ pandas

Isso garante que todas as dependências necessárias sejam instaladas no ambiente do container.

Criação do Dockerfile

O Dockerfile foi criado para automatizar a construção do ambiente e execução do programa:

```
FROM python:3.10-slim

WORKDIR /app

COPY . /app

RUN pip install --no-cache-dir -r requirements.txt

CMD ["python", "app.py"]
```

Build da Imagem Docker

No terminal, dentro da pasta do projeto, foi executado:

docker build -t feedback360 .

```
:\Users\Yasmin\Documents\feedback360>docker build -t feedback360 .
+] Building 211.1s (9/9) FINISHED
                                                                                                        docker:desktop-linux
  [internal] load build definition from Dockerfile
                                                                                                                         0.4s
=> => transferring dockerfile: 173B
=> [internal] load metadata for docker.io/library/python:3.10-slim
=> [internal] load .dockerignore
                                                                                                                         5.5s
=> [1/4] FROM docker.io/library/python:3.10-slim@sha256:e1013c40c02a7875ae30c78c69b68ea7bee31713e8ac1c0f5469c120
=> => resolve docker.io/library/python:3.10-slim@sha256:e1013c40c02a7875ae30c78c69b68ea7bee31713e8ac1c0f5469c120
                                                                                                                        0.3s
                                                                                                                         0.2s
                                                                                                                         0.1s
                                                                                                                         0.0s
=> [3/4] COPY . /app
=> [4/4] RUN pip install --no-cache-dir -r requirements.txt
                                                                                                                        0.2s
                                                                                                                       113.6s
=> exporting to image
                                                                                                                        89.2s
=> => exporting layers
=> => exporting manifest sha256:7bc827fcf3ed013bc1d7b4b6a15de962056df5d9edd044b0724281c89bbf5f80
=> => exporting config sha256:14908d99dfdfb51515cf9f6c4bffb18dfe94779bf305bb638cbe7751b73ff0db
                                                                                                                         0.1s
=> => exporting attestation manifest sha256:fd00a408f89cacde81a2cfbaf38231fbd3ba1bf675fd52c5e35f8053ce3d6ace
=> => exporting manifest list sha256:4ea65b4217151a9ba30016ff311d6baf250d5d6<u>164e</u>8f39c3036c6b287f93a79
                                                                                                                         0.1s
=> => naming to docker.io/library/feedback360:latest
                                                                                                                         0.0s
=> => unpacking to docker.io/library/feedback360:latest
                                                                                                                        32.2s
```

Execução do Container

Para rodar o programa e salvar o relatório na máquina local, foi utilizado:

→ docker run -it -v C:\Users\Yasmin\Documents\feedback360:/app feedback360

O parâmetro -it permite interação no terminal.

O parâmetro -v faz o bind da pasta local para que o arquivo relatorio.csv gerado dentro do container fique disponível na máquina.

```
:\Users\Yasmin\Documents\feedback360>docker run -it -v C:\Users\Yasmin\Documents\feedback360:/app feedback360
 == Sistema de Feedback 360° ===
Quantas pessoas na equipe? 2
Nome da pessoa 1: Yasmin
Nome da pessoa 2: Rafael
Feedback para Yasmin:
Nota para comunicacao (1-10): 5
Nota para empatia (1-10): 8
Nota para lideranca (1-10): 8
Comentário: faz boa parte do projeto
Feedback para Rafael:
Nota para comunicacao (1-10): 8
Nota para empatia (1-10): 7
Nota para lideranca (1-10): 7
omentário: fez boa parte do projeto
 == Relatório Gerado ===
    nome comunicacao empatia lideranca
                                                            comentario
  Yasmin
                              8
                                          8 faz boa parte do projeto
                                          7 fez boa parte do projeto
  Rafael
elatório salvo como relatorio.csv
```

O programa foi testado inserindo diferentes quantidades de pessoas, notas e comentários.

O arquivo relatorio.csv foi gerado corretamente na pasta do projeto.

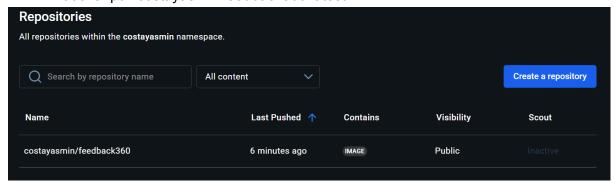
Publicação da Imagem no Docker Hub

Após a validação local, a imagem Docker foi publicada no Docker Hub para facilitar o compartilhamento e a execução em outros ambientes. Os comandos utilizados foram

- → docker login
- → docker tag feedback360cli costayasmin/feedback360i:latest
- → docker push costayasmin/feedback360:latest

Com isso, a imagem ficou disponível publicamente no Docker Hub, podendo ser baixada e executada em qualquer máquina com Docker instalado

→ docker pull costayasmin/feedback360:latest



Considerações Finais

- O sistema pode ser facilmente adaptado para incluir mais soft skills ou outros tipos de análise.
- O uso do Docker garante portabilidade e facilidade de execução em qualquer ambiente.
- O relatório gerado pode ser utilizado para análise de desempenho e desenvolvimento de equipes.