

Sistema de Feedback 360° com Análise de Soft Skills

Guilherme Alves de Oliveira e Oliveira
João Pedro Lima Paulo
Lara Marina de Oliveira
Giulia Nogueira Lopes De Sá
Beatriz de Castilho Ferreira

1. Introdução

Este relatório descreve a implementação de um sistema de feedback 360°, focado na avaliação de soft skills como comunicação, empatia, liderança e proatividade. A aplicação foi desenvolvida com Python e Flask, containerizada com Docker, e permite que os usuários avaliem colegas de equipe anonimamente, com os resultados sendo armazenados, processados e exibidos graficamente.

2. Tecnologias Utilizadas

- Python
- Flask
- Pandas
- Matplotlib
- HTML/CSS
- Docker
- SQLite

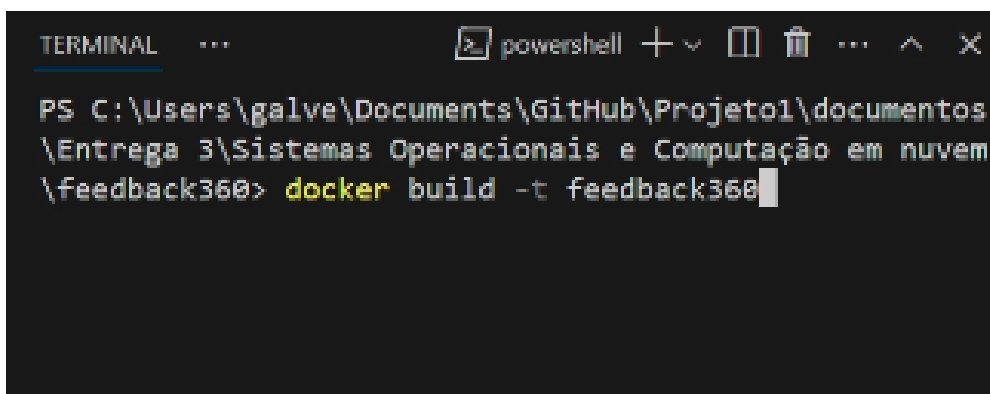
3. Estrutura da Aplicação

A aplicação é dividida em módulos principais:

- Formulário de nomes dos avaliados
- Ciclo individual de avaliação
- Geração de relatório com gráficos
- Interface web com rotas Flask
- Dockerfile para a containerização

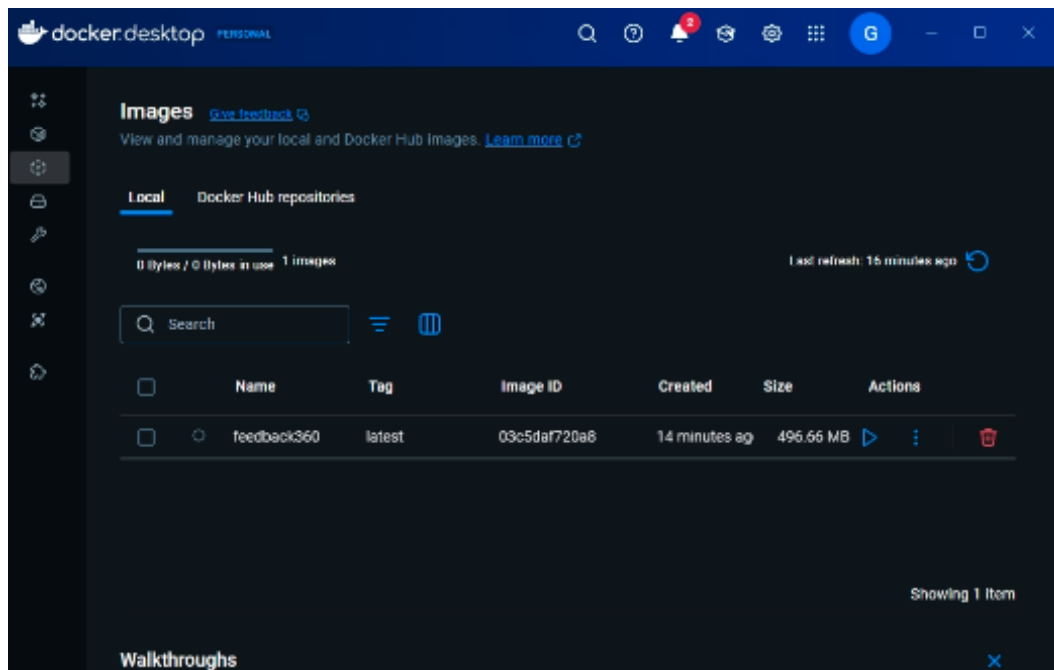
4. Criação da imagem e Push para Docker Hub:

Após a aplicação ser feita, devemos criar a imagem do docker:

A screenshot of a Windows PowerShell terminal window. The title bar at the top reads "TERMINAL" followed by a menu icon and "powershell". The terminal content shows the current directory path as "PS C:\Users\galve\Documents\GitHub\Projeto1\documentos\Entrega 3\Sistemas Operacionais e Computação em nuvem\feedback360". The user has entered the command "docker build -t feedback360" and the cursor is at the end of the line.

```
TERMINAL  ...  powershell + v [icon] [icon] ... ^ X

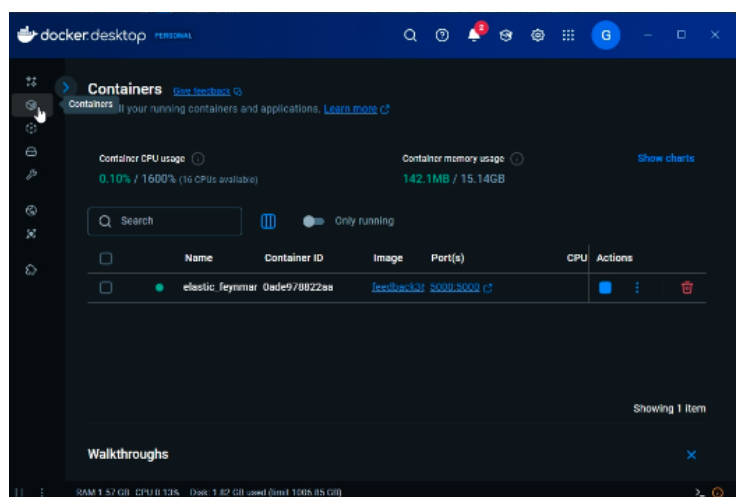
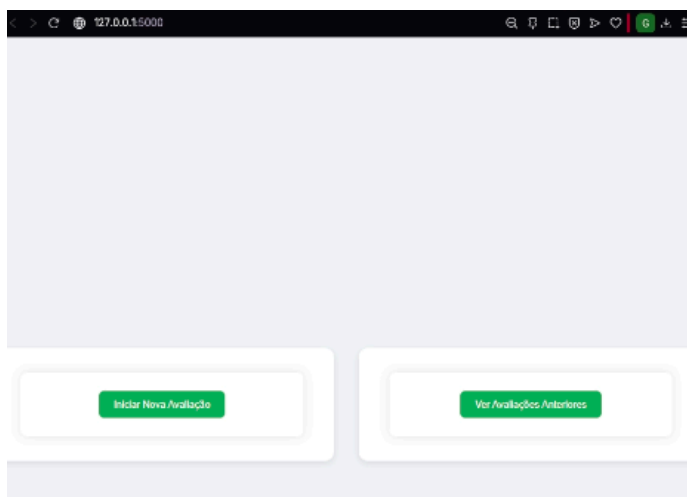
PS C:\Users\galve\Documents\GitHub\Projeto1\documentos
\Entrega 3\Sistemas Operacionais e Computação em nuvem
\feedback360> docker build -t feedback360
```



Devemos conferir se a aplicação está rodando no Docker localmente:

```
PS C:\Users\galve\Documents\GitHub\Projeto1\documentos
\Entrega 3\Sistemas Operacionais e Computação em nuvem
\feedback360> docker run -p 5000:5000 feedback360
```

E está tudo correto:



Para subirmos a Aplicação para o docker hub, primeiro devemos fazer o login pelo terminal:

```
PS C:\Users\galve\Documents\GitHub\Projeto1\documentos
\Entrega 3\Sistemas Operacionais e Computação em nuvem
\feedback360> docker login
```

```
! Info → To login with a different account, run 'docke
r logout' followed by 'docker login'
```

```
Login Succeeded
```

Devemos renomear a aplicação, para que o destino seja a própria conta do DockerHub:

```
Login Succeeded
PS C:\Users\galve\Documents\GitHub\Projeto1\documentos
\Entrega 3\Sistemas Operacionais e Computação em nuvem
\feedback360> docker tag feedback360 guialvesoliveira/
feedback360
```

Após isso, podemos realizar o *push* :

```
PS C:\Users\galve\Documents\GitHub\Projeto1\documentos
\Entrega 3\Sistemas Operacionais e Computação em nuvem
\feedback360> docker push guialvesoliveira/feedback360
```

5. Configurações na AWS para receber a Aplicação

Devemos iniciar com a instalação do docker dentro da aws:

```
aws
```

América do Sul (São Paulo) ▼ Guizaum471 ▼

```
ec2-user@ip-172-31-36-98 ~]$ sudo yum install -y docker
```

```
iptables-libs-1.8.8-3.amzn2023.0.2.x86_64      iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
libcgroup-3.0-1.amzn2023.0.1.x86_64            libnetfilter_conntrack-1.0.8-2.amzn2023.0
.2.x86_64
libnftnl-1.0.1-19.amzn2023.0.2.x86_64          libnftnl-1.2.2-2.amzn2023.0.2.x86_64
pigz-2.5-1.amzn2023.0.3.x86_64                 runc-1.2.4-1.amzn2023.0.1.x86_64
```

```
Complete!
```

Vamos logar novamente no Docker:

```
ec2-user@ip-172-31-36-98 ~]$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If
you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-sco
pe PAT grants better security and is required for organizations using SSO. Learn more a
t https://docs.docker.com/go/access-tokens/

Username: 
Username: guialvesoliveira
Password: 
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-36-98 ~]$
```

Agora iremos fazer o Pull do docker hub:

```
[ec2-user@ip-172-31-36-98 ~]$ sudo docker pull guialvesoliveira/feedback360
Using default tag: latest
latest: Pulling from guialvesoliveira/feedback360
254e724d7786: Pull complete
5e3380732964: Pull complete
5cc9686f2aa9: Pull complete
5d66a27e733a: Pull complete
160899883894: Pull complete
a69aad978faa: Pull complete
68f5018ac3d6: Pull complete
Digest: sha256:03c5daf720a0bb3498cd60c01565ddd01dedea67a705fab5735aaab5590dc04
Status: Downloaded newer image for guialvesoliveira/feedback360:latest
docker.io/guialvesoliveira/feedback360:latest
[ec2-user@ip-172-31-36-98 ~]$
```

E iremos rodar localmente dentro da nossa EC2:

```
[ec2-user@ip-172-31-36-98 ~]$ sudo docker run guialvesoliveira/feedback360
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a
production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 629-467-000
```

Mas temos um porém.

Para podermos acessar essa aplicação rodando, dentro da AWS devemos criar um IP elástico, e liberar as portas que iremos utilizar na aplicação. Nesse caso iremos utilizar as portas 5000 tanto de entrada e saída nos *Security Groups*.

The screenshot shows the AWS Security Groups console. The 'Protocol' dropdown is set to 'TCP personalizado'. The 'Port Range' is set to '5000'. The 'Source' is set to 'Qualquer local-IPv4'. The 'Destination' is set to '0.0.0.0'. The 'Description' field is empty.

Agora podemos rodar nossa aplicação pela porta 5000, e acessar com <IP-ELÁSTICO>:5000.

```
[ec2-user@ip-172-31-36-98 ~]$ sudo docker run -p 5000:5000 guialvesoliveira/feedback360
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a
production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 629 467 888
201.6.227.54 - - [15/May/2025 15:10:07] "GET / HTTP/1.1" 200 -
201.6.227.54 - - [15/May/2025 15:10:07] "GET /static/style.css HTTP/1.1" 200 -
201.6.227.54 - - [15/May/2025 15:10:08] "GET /favicon.ico HTTP/1.1" 404 -
```

Para acessar clique aqui : 52.67.60.209:5000

