Kauan Felipe Dutra de Almeida – 2403004

1. Verifique a Instalação:
    - Verifique a versão do Docker.

```
Microsoft Windows [versão 10.0.19045.6332]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\2403004>docker --version
Docker version 28.1.1, build 4eba377
```

2. Baixe uma Imagem:
    - Baixe a imagem do servidor Nginx.

```
C:\Users\2403004\julia>docker pull nginx
Using default tag: latest
Error response from daemon: error from registry: failed to resolve reference "docker.io/library/nginx:latest": failed to
 authorize: failed to fetch oauth token: unexpected status from GET request to https://auth.docker.io/token?scope=reposi
tory%3Alibrary%2Fnginx%3Apull&service=registry.docker.io: 401 Unauthorized

C:\Users\2403004\julia>docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
4f4e50e20765: Pull complete
08cfef42fd24: Pull complete
5c32499ab806: Pull complete
375a694db734: Pull complete
3cc5fdd1317a: Pull complete
16d05858bb8d: Pull complete
5f825f15e2e0: Pull complete
Digest: sha256:8adbdcb969e2676478ee2c7ad333956f0c8e0e4c5a7463f4611d7a2e7a7ff5dc
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

3. Rode o Primeiro Contêiner:
    - Baixe e rode o contêiner "Hello World"

```
C:\Users\2403004\julia>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:54e66cc1dd1fcb1c3c58bd8017914dbed8701e2d8c74d9262e26bd9cc1642d31
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

4. Inicie um Contêiner:
    - Inicie o Nginx em um contêiner.
    - Acesse o Nginx rodando em um navegador.

```
C:\Users\2403004\julia>docker run -d -p 8181:80 --name meu-nginx nginx
98bfbb37ad60ca6a0b8631602270ad0eaa55abe3b24187f3d85817d2188204a5
```

5. Liste os Contêineres:
    - Liste os contêineres em execução.

```
C:\Users\2403004\julia>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                  NAMES
98bfbb37ad60   nginx     "/docker-entrypoint.…"   4 minutes ago   Up 4 minutes   0.0.0.0:8181->80/tcp   meu-nginx
```

6. Pare e Remova o Contêiner 'Hello World':
    - Pare o contêiner "Hello World".
    - Remova o contêiner "Hello World".

```
:\Users\2403004\julia>docker ps -a
ONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                  NAMES
98bfbb37ad60   nginx     "/docker-entrypoint.…"   10 minutes ago   Up 10 minutes   0.0.0.0:8181->80/tcp   meu-nginx
```

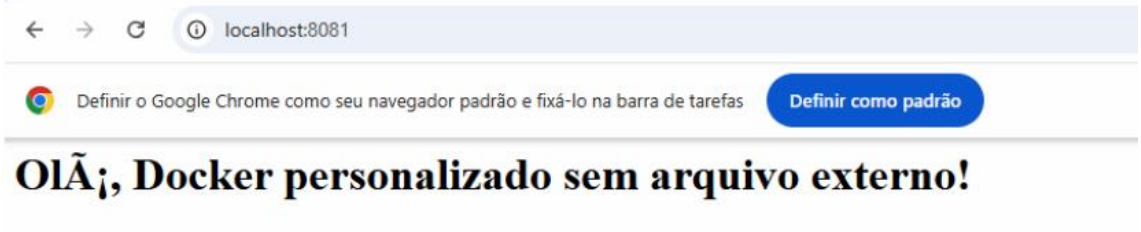
7. Remova Imagens:
    - Remova uma imagem baixada.

```
C:\Users\2403004\julia>docker rmi hello-world
Untagged: hello-world:latest
Deleted: sha256:54e66cc1dd1fcb1c3c58bd8017914dbed8701e2d8c74d9262e26bd9cc1642d31
```

8. Crie um Dockerfile:
    - Crie um arquivo Dockerfile.
    - Construa uma imagem a partir do Dockerfile.
    - Execute a imagem criada.

```
C:\Users\2403004\julia>
C:\Users\2403004\julia>RUN echo '<h1>Olá, Docker personalizado sem arquivo externo!</h1>' > /usr/share/nginx/html/index.html
O sistema não pode encontrar o arquivo especificado.
[+] Building 3.3s (7/7) FINISHED                                                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                                0.0s
 => => transferring dockerfile: 157B                                                                                                0.0s
 => [internal] load metadata for docker.io/library/nginx:latest                                                                     0.1s
 => [internal] load .dockerignore                                                                                                   0.0s
 => => transferring context: 2B                                                                                                     0.0s
 => [1/2] FROM docker.io/library/nginx:latest@sha256:8adbdcb969e2676478ee2c7ad333956f0c8e0e4c5a7463f4611d7a2e7a7ff5dc               1.7s
 => => resolve docker.io/library/nginx:latest@sha256:8adbdcb969e2676478ee2c7ad333956f0c8e0e4c5a7463f4611d7a2e7a7ff5dc               1.3s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                                        0.0s
 => [2/2] RUN echo '<h1>Olá, Docker personalizado sem arquivo externo!</h1>' > /usr/share/nginx/html/index.html                     0.6s
 => exporting to image                                                                                                              0.5s
 => => exporting layers                                                                                                             0.2s
 => => exporting manifest sha256:d1fb9b63fcfdeb2f10fa3cec5ce0d06e5720e046361d6a862d9a01b0010de46c                                   0.0s
 => => exporting config sha256:cdfb12401b49a32c7df4c12b8dd123f5e7bd51c5b669bdcc347afb5fc6d9bf9e                                     0.0s
 => => exporting attestation manifest sha256:dbaad5d3b2be040a6742368f2b87894af7bdceacd8e0449d0a4b601dea6713b8                       0.0s
 => => exporting manifest list sha256:a469b08144a97ad8b3120933670b1534f9774312d2e33ce6420683b1bdf66b46                              0.0s
 => => naming to docker.io/library/meu-nginx-custom:latest                                                                          0.0s
 => => unpacking to docker.io/library/meu-nginx-custom:latest                                                                       0.1s

C:\Users\2403004\julia>docker run -d -p 8081:80 --name nginx-custom meu-nginx-custom
5806e188b60120efe518ec62981f6ddcb96de460b125604ff1df4717f582ef4f
```



9. Utilize Volumes:
    - Crie um volume.
    - Execute um contêiner com um volume.
    - Verifique o conteúdo do volume dentro do contêiner.

```
C:\Users\2403004\julia>docker volume create meu-volume
meu-volume

C:\Users\2403004\julia>docker run -d -p 8082:80 -v meu-volume:/usr/share/nginx/html --name nginx-volume nginx
63c3f0f9976bb0bcf1ffa4483f8e762b8f48df2b42a70d63fcc74fdfa249c084

C:\Users\2403004\julia>docker exec -it nginx-volume bash
ls /usr/share/nginx/html
root@63c3f0f9976b:/# ls /usr/share/nginx/html
50x.html  index.html
root@63c3f0f9976b:/#
```

10. Comunique Contêineres via TCP/IP:
    - Crie uma rede Docker.
    - Rode dois contêineres na mesma rede.
    - Comunique-se entre os contêineres

```
C:\Users\2403004\julia>docker network create minha-rede
6f69ee101d3114f20ee5f175608de25260449650fe6afe210b18c83bbfc237fa

C:\Users\2403004\julia>docker run -dit --name web1 --network minha-rede nginx
17926968f1acc7d6e911f5b14a40b8e6d35bf51e3846537d68bc3c41b44826f4

C:\Users\2403004\julia>docker run -dit --name web2 --network minha-rede nginx
22b9bca04ee9cfc11a07ed3b7d0d96e8d5c4c9e5bac8f80489808b003b3220b2

C:\Users\2403004\julia>docker exec -it web1 ping web2
DCI runtime exec failed: exec failed: unable to start container process: exec: "ping": executable file not found in $PATH: unknown
```