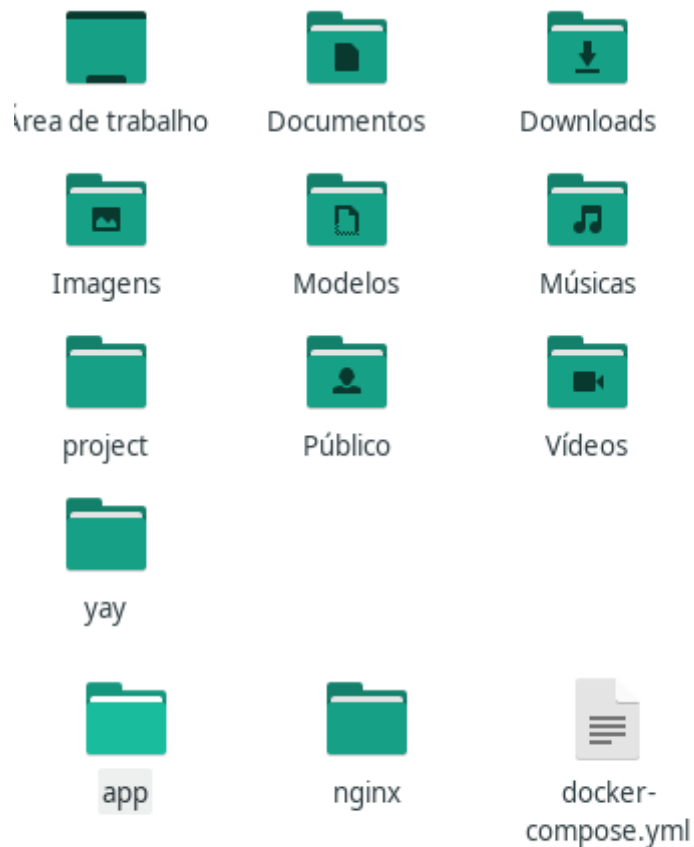
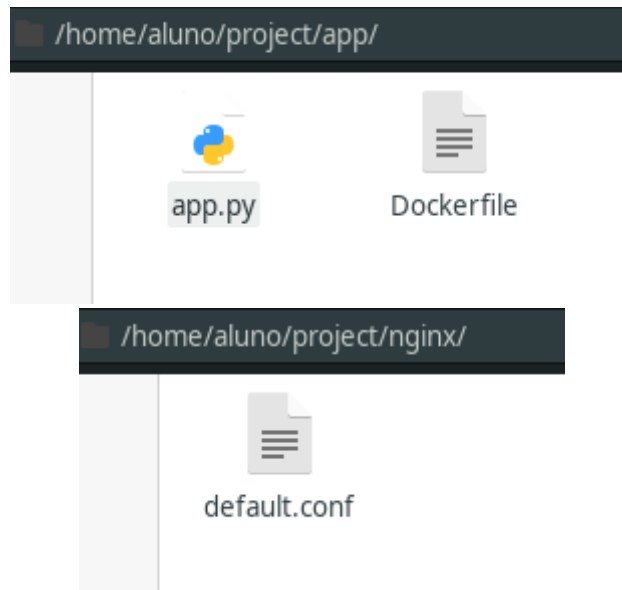


# ATIVIDADE 8

Aluno: Lucas Muniz Alves

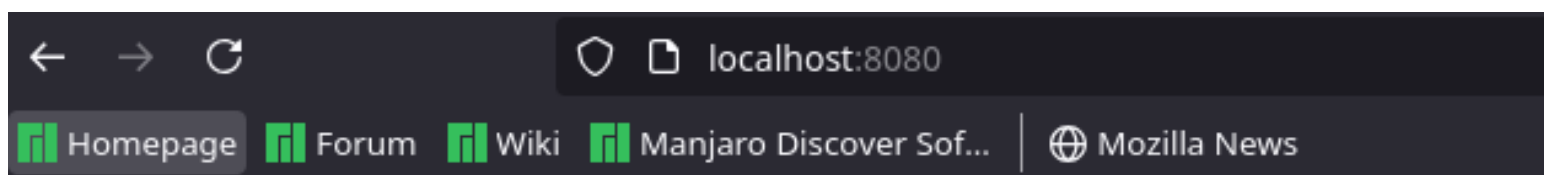
```
project/  
|— docker-compose.yml  
|— nginx/  
|   └─ default.conf  
|— app/  
|   └─ app.py  
|   └─ Dockerfile
```





```
Arquivo Editar Ver Terminal Abas Ajuda
[redes-1 project]# docker-compose up -d
WARN[0001] /home/aluno/project/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Building 21.2s (10/10) FINISHED
    docker:default
=> [backend internal] load build definition from Dockerfile 0.1s
=> -- transferring dockerfile: 127B 0.1s
=> [backend internal] load metadata for docker.io/library/python:3.9 1.4s
=> [backend internal] load .dockerignore 0.0s
=> -- transferring context: 2B 0.0s
=> [backend 1/4] FROM docker.io/library/python:3.9@sha256:c17c71e1f5f258 0.0s
=> [backend internal] load build context 0.0s
=> -- transferring context: 28B 0.0s
=> CACHED [backend 2/4] WORKDIR /app 0.0s
=> CACHED [backend 3/4] COPY app.py . 0.0s
=> [backend 4/4] RUN pip install flask 14.1s
=> [backend] exporting to image 5.3s
=> -- exporting layers 5.2s
=> -- writing image sha256:8d8ce73ef3140d41ee531ee5760b05a2c08389a116e 0.0s
=> -- naming to docker.io/library/project-backend 0.0s
=> [backend] resolving provenance for metadata file 0.0s
[+] Running 4/4
✔ backend Built 0.0s
✔ Network project_default Created 0.1s
✔ Container backend_service Started 1.2s
✔ Container proxy Started 1.1s
```

Acessando <http://localhost:8080>



Olá, este é um serviço intermediado pelo Proxy Reverso!

**Implementar um Proxy Reverso usando NGINX. A implementação deve usar containers usando Docker-Compose de forma a usar serviços cujo acesso deve feito**

através do Proxy Reverso. Evidenciar a configuração do Docker Compose, a configuração do NGINX e o acesso a um serviço intermediado pelo Proxy Reverso.

### Estrutura do Projeto:

```
project/
├── docker-compose.yml
├── nginx/
│   └── default.conf
├── app/
│   ├── app.py
│   └── Dockerfile
```

#### 1 Criar o Serviço Backend (Flask)

Crie um diretório **app/** e adicione um arquivo **app.py**:

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def home():
```

```
    return "Olá, este é um serviço intermediado pelo Proxy Reverso!"
```

```
if __name__ == "__main__":
```

```
    app.run(host="0.0.0.0", port=5000)
```

Adicione um **Dockerfile** dentro da pasta **app/**:

```
FROM python:3.9
```

```
WORKDIR /app
```

```
COPY app.py .
```

```
RUN pip install flask
```

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

## 2 Configurar o NGINX como Proxy Reverso

Crie um diretório **nginx/** e um arquivo **default.conf** dentro dele:

```
server {  
    listen 80;  
  
    location / {  
        proxy_pass http://backend:5000;  
        proxy_set_header Host $host;  
        proxy_set_header X-Real-IP $remote_addr;  
    }  
}
```

## 3 Criar o Arquivo **docker-compose.yml**

Na raiz do projeto, crie o arquivo **docker-compose.yml**:

```
version: '3.8'
```

```
services:
```

```
  nginx:
```

```
    image: nginx:latest
```

```
    container_name: proxy
```

```
    ports:
```

```
      - "8080:80"
```

```
    volumes:
```

```
      - ./nginx/default.conf:/etc/nginx/conf.d/default.conf
```

```
    depends_on:
```

```
      - backend
```

```
  backend:
```

```
    build: ./app
```

```
    container_name: backend_service
```

```
    expose:
```

```
      - "5000"
```

#### **4 Rodar o Projeto**

No terminal, dentro da pasta **project/**, execute:

```
docker-compose up -d
```

Acesse **<http://localhost:8080/>** no navegador e você verá a resposta do backend sendo servida pelo Proxy Reverso!