

## Tarea 1

### Laboratorio de Sistemas Operativos 1

Erick Javier Bernal Orellana

201480017

#### 1. Python y docker

- DockerFile

```
apiFlas > Dockerfile > ...
1  # syntax=docker/dockerfile:1
2
3  FROM python:3.8-slim-buster
4
5  WORKDIR /python-docker
6
7  COPY requirements.txt requirements.txt
8  RUN pip3 install -r requirements.txt
9
10 COPY . .
11 EXPOSE 5000
12 CMD ["flask", "run", "--host=0.0.0.0"]
```

- Comandos para correr el contenedor.

```
#construi la imagen
docker build -t hellow_python_flask .

#correr la imagen
docker run -p 5000:5000 hellow_python_flask
```

## 2. Golang y docker

- DockerFile

```
apiGolan > Dockerfile > ...
1  # syntax=docker/dockerfile:1
2
3  FROM golang:1.16-alpine
4
5  WORKDIR /app
6
7  COPY go.mod ./
8  COPY go.sum ./
9  RUN go mod download
10
11 COPY *.go ./
12
13 RUN go build -o /main
14
15 EXPOSE 8080
16
17 CMD [ "/main" ]
```

- Comandos para correr el contenedor.

```
#construir la imagen de docker
docker build --tag hellow_go .

#ejecutar la imagen de docker
docker run -p 8080:8080 hellow_go
```

## 3. Consulta de endpoint, desde navegador.

The screenshot displays a terminal window at the top showing the output of the command `docker ps -a`. It lists two containers: `exciting_ardinghelli` (image `hellow_python_flask`, command `"flask run --host=0.0.0.0"`) and `wonderful_feistel` (image `hellow_go`, command `"/main"`). Below the terminal, two web browser windows are shown. The left browser window is at `localhost:5000` and displays the text "Hellow World, flask python!". The right browser window is at `localhost:8080` and displays the text "Hello Worl Go, Docker!".