

## INTRODUCCIÓN DOCKER

- FROM nginx:latest rightarrow necesario para archivos HTML.
- COPY /sitio /usr/share/nginx/html rightarrow va a correr con un servidor web.

## BORRAR IMÁGENES:

```
docker rmi -f 5bed...  
docker rmi sitioweb:latest.
```

## CONSTRUIR UNA IMAGEN:

```
docker build -t sitioweb:latest .
```

## CORRER CONTENEDOR:

```
docker run -it --rm -d -p 8080:80 --name web sitioweb
```

- -it: Interactivo
- -rm: Borra anteriores, segundo plano
- -d: Detached (ejecuta en segundo plano)
- -p: Puerto (externo:interno)
- --name: Nombre del contenedor
- sitioweb: Imagen

## BUSCAR IMAGEN:

```
docker images sitioweb  
docker images --filter="reference='*:1.0'"
```

## ACTUALIZAR TAG:

```
docker image tag sitioweb:latest gmin/sitioweb:latest .
```

## TAMAÑO DE CONTENEDOR:

```
docker ps --size
```

## ESTADÍSTICAS:

```
docker stats
```

## PARAR CONTENEDOR:

```
docker stop 5bed...
```

COMPRIMIR IMAGEN:

```
docker save g.longel/linktree > linktree.rar
```

CARGAR RAR IMAGEN:

```
docker load --input linktree.rar
```

GESTIÓN DE CONTENEDORES DOCKER COMPOSE

docker-compose.yml:

YAML

```
version: '3.7'
services:
  backend:
    image: backend
    build:
      context: ./backend
      dockerfile: Dockerfile
    ports:
      - 5000:5000
  frontend:
    image: frontend
    build:
      context: ./frontend
      dockerfile: Dockerfile
    ports:
      - 8080:80
```

DESPLEGAR API DE PYTHON FLASK:

CORRER EN LOCAL:

```
python -m flask run
python --version
```

## DOCKERFILE:

### Dockerfile

```
FROM python:3.12-alpine3.17.  
WORKDIR /app  
COPY requirements.txt requirements.txt  
RUN pip install -r requirements.txt.  
COPY . .  
CMD ["python", "-m", "flask", "run", "--host=0.0.0.0"]
```

## VOLÚMENES:

```
docker run -it --rm -d -p 8080:80 -v /sitio:/urs/share/nginx/html --name web nginx
```

¡Alerta! puedes interactuar con tu máquina en tiempo real

## REDES:

```
docker network ls
```

## CREAR RED:

```
docker network create platzinet
```

## PUBLICAR IMAGEN: (DOCKER HUB)

```
docker login docker build -t g10n092/linktree:latest docker push g10n092/linktree:latest
```

## COMANDOS PARA HACER UN BUILD CON DOCKER COMPOSE:

### Bash

```
docker compose build
```

Si deseas construir las imágenes y levantar los servicios (es decir, iniciar los contenedores), puedes usar:

### Bash

```
docker compose up --build
```

El `--build` asegura que las imágenes se construyan (o reconstruyan si hay cambios) antes de que los servicios se inicien.