

**FACULTAD DE PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN**



Internet de las Cosas

ALUMNOS:

Quiñonez Lopez Efrain
Saravia Velasquez Paul Diego
Torres Mamani Fernando Miguel

AREQUIPA, PERÚ

2021

1. Docker-compose

Para ejecutar los ejercicios del capítulo 7 del libro de “Build Your Own IoT Platform” era necesario levantar los servicios de:

- Node-red
- Mosquitto
- MySQL
- phpMyAdmin

En la ruta ./ raíz del repositorio podrá encontrar el archivo docker-compose.yml, este archivo contendrá los 4 servicios ya antes mencionados, podrá encontrar etiquetas por cada servicio como:

- Image: Nombre de la imagen a descargar desde dockerhub.
- Container_name: Nombre para identificar al contenedor con el servicio
- Networks: Esta configuración permitirá tener a todos los contenedores que lo posean conexión y visibilidad entre ellos.
- Ports: Esta etiqueta mapea tanto el puerto del contenedor como el de nuestra máquina local, para poder acceder sin problemas.
- Configuración de base de datos: Aquí existen varias variables de configuración como las contraseñas y usuarios para la base de datos.

Como primer paso, luego de analizar nuestro yml procedemos a ejecutar con el siguiente comando este archivo.

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

Captura de resultado:

```
efra@efra-X540UA:~/Documentos/uns-codex/2021-B/iot/iot_continua$ docker-compose up -d
Creating network "iot_continua_von" with the default driver
Recreating db ... done
Recreating mosquitto ... done
Recreating pma ... done
Recreating node-red ... done
```

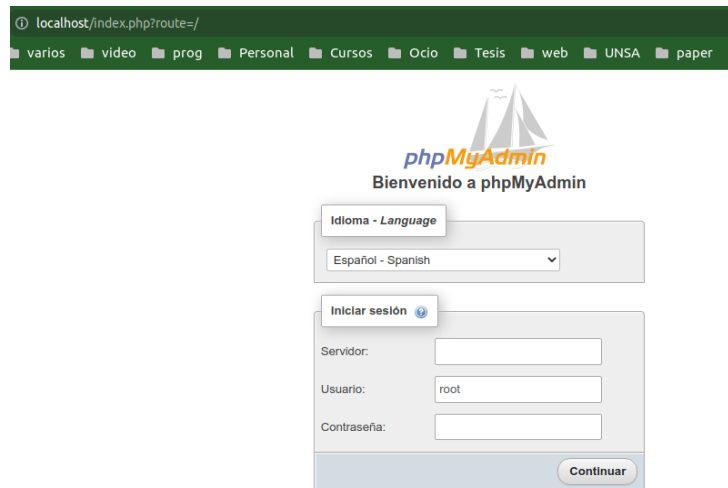
Luego podremos observar los 4 servicios funcionando si todo va bien, se usa el siguiente comando:

```
docker ps
```

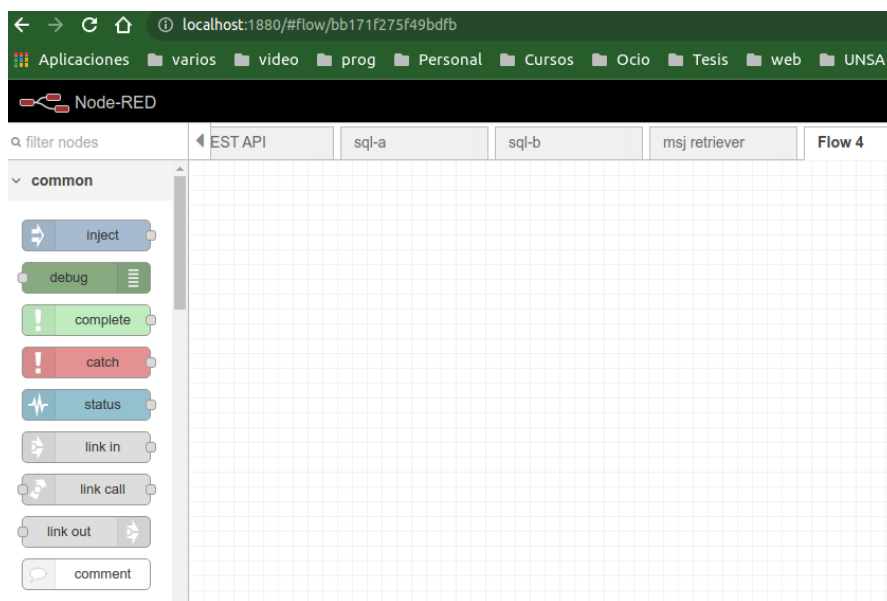
| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS |
|--------------|-----------------------|--------------------------|-------------------|----------------------------|--|
| 27a3ddf9bc49 | phpmyadmin/phpmyadmin | "/docker-entrypoint..." | About an hour ago | Up About an hour | 0.0.0.0:80->80/tcp, :::80->80/tcp |
| f3a5568dd527 | mysql:5.7 | "docker-entrypoint.s..." | About an hour ago | Up About an hour | 0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp |
| 95503f717b48 | node-red:latest | "npm --no-update-not..." | About an hour ago | Up About an hour (healthy) | 0.0.0.0:1880->1880/tcp, :::1880->1880/tcp |
| c0248e68c372 | eclipse-mosquitto:1.6 | "/docker-entrypoint..." | About an hour ago | Up About an hour | 0.0.0.0:1883->1883/tcp, :::1883->1883/tcp, 0.0.0.0:9001->9001/tcp, :::9001->9001/tcp |

2. Verificación de los diferentes servicios

a. MySQL y phpMyAdmin



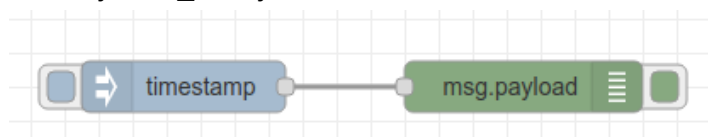
b. Node-Red



3. Desarrollo de actividades del capítulo 7

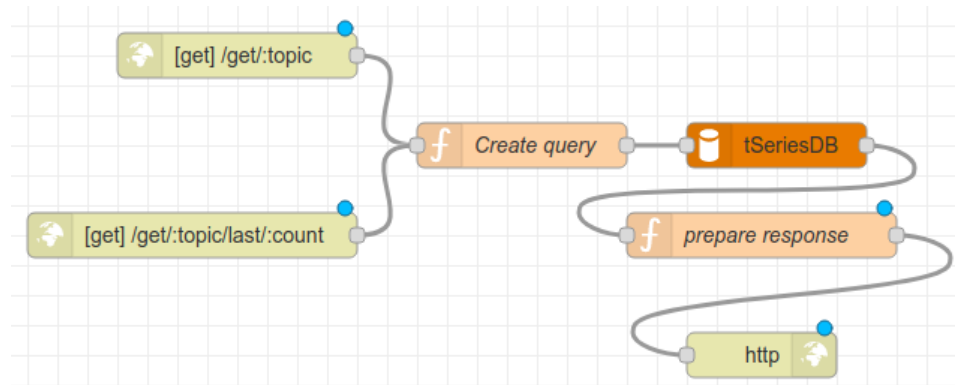
A continuación se presentan las actividades con su respectiva ruta en nuestro repositorio donde encontrará los archivos JSON donde contienen la mayoría de configuraciones para poder importarlo en el servicio de Node-Red para replicarlo con mayor facilidad.

- a. Creación de base de datos para próximos ejercicios
Ruta: ./database/tSeriesDB.sql
- b. Creando el primer flujo para nuestra plataforma
Ruta: ./json/0_intro.json



- c. Adición de la capacidad de publicación de MQTT
Ruta: ./json/1_TimeStamp.json

Ruta: ./json/5_retriever.json



h. Ejecución de Node-RED en segundo plano continuamente

Para este ejercicio debíamos ingresar al contenedor para poder instalar una instancia de una librería “forever”, se logro listar el servicio de mosquito:

```
efra@efra-X540UA:~/Documentos/unsas-codex/2021-B/iot/iot_continua$ docker exec -it mosquito /bin/sh
/ # npm install forever -g
/bin/sh: npm: not found
/ # lsof -I :1880
44 /bin/busybox /dev/pts/0
44 /bin/busybox /dev/pts/0
44 /bin/busybox /dev/pts/0
44 /bin/busybox /dev/tty
/ # exit
```

4. Link de Github

- <https://github.com/PaulSV123/Internet-de-las-Cosas>