

Ejercicios de Repaso-Docker

1. Instala docker en una máquina y configúralo para que se pueda usar con un usuario sinprivilegios.

```
usermod -aG docker usuario
```

□

2. Ejecuta un contenedor a partir de la imagen hello-word. Comprueba que nos devuelve lasalida adecuada. Comprueba que no se está ejecutando. Lista los contenedores que estánparado. Borra el contenedor.

- Ejecutar contendor con imagen hello-world

```
docker run hello-world
```

□

- Comprobacion de que no se está ejecutando, se ejecuta una vez y cuando esta deja de estar activa, se apaga automaticamente, comprobacion:

```
docker ps
```

□

- Lista de los contenedores

```
docker ps -a
```

□

- Eliminar el contenedor , en este caso elegi por su ID, aunque tambien se puede por su nombre

```
docker rm a8b2baeae079
```

□

3. Crea un contenedor interactivo desde una imagen debian. Instala un paquete (por ejemplonano). Sal de la terminal, ¿sigue el contenedor corriendo? ¿Por qué?. Vuelve a iniciar elcontenedor y accede de nuevo a él de forma interactiva. ¿Sigue instalado el nano?. Sal delcontenedor, y bórralo. Crea un nuevo contenedor interactivo desde la misma imagen. ¿Tiene el nano instalado?

- Creacion del contenedor

```
docker run -it --name contenedorInteractivo debian bash
```

□ -Instalación nano

```
docker exec contenedorInteractivo apt-get install nano
```

□

DockerDoc1_Repaso_AinhoaSalas

- Borramos el contenedor, y comprobamos en la lista que ya no aparece

```
docker rm -f contenedorInteractivo
docker ps -a
```

□

Preguntas:

- ¿sigue el contenedor corriendo?, ¿por que ? → No, una vez fuera de la terminal el contenedor se apaga
 - ¿Sigue instalado el nano? → Si, se mantienen las descargas ejecutadas
4. Crea un contenedor demonio con un servidor nginx, usando la imagen oficial de nginx. Alcrear el contenedor, ¿has tenido que indicar algún comando para que lo ejecute? Accede al navegador web y comprueba que el servidor esta funcionando. Muestra los logs del contenedor.
- Creacion del contendor demonio Ejecutamos el contenedor a partir de la imagen con docker container run

```
docker container run --name my-nginx -p 80:80 nginx
```

□

```
https://127.0.0.1
```

□

- Mostrar los logs del contenedor

```
docker logs my-nginx
```

□ □

5. Crea un contenedor con la aplicación Nextcloud, mirando la documentación en docker Hub, para personalizar el nombre de la base de datos sqlite que va a utilizar.

- Descargamos nextcloud

```
docker pull nextcloud
```

□

```
docker run -d -p 8080:80 nextcloud
```

```
docker ps -a
```

□

