

EPD 1

**Sistemas De Alto Rendimiento
Escalables**

Docker

Índice

Ejercicio 1 3

Ejercicio 2 4

Problema 1 6

Problema 2 8

Ejercicio 1

¿Cómo sabemos cuál es la versión de Docker que tenemos instalada?

```
C:\Users\agarg>docker --version
Docker version 20.10.21, build baeda1f
```

¿Cuántos contenedores hay funcionando en su PC?

```
C:\Users\agarg>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
```

Pon a funcionar un contenedor usando la imagen de debian:

```
C:\Users\agarg>docker run -i -t debian
root@f5cf68596b2e:/#
```

Para el contenedor que has creado:

```
C:\Users\agarg>docker stop clever_colden
clever_colden
```

Elimina el contenedor que has creado:

```
C:\Users\agarg>docker rm clever_colden
clever_colden
```

Ejercicio 2

Crear contenedor debian:

```
C:\Users\agarg>docker run -it --name test1 debian  
root@89c3a9e7ae97:/#
```

Acceder al contenedor desde otra terminal diferente:

```
C:\Users\agarg>docker exec -it test1 /bin/bash  
root@89c3a9e7ae97:/#
```

Probando unos comandos dentro del contenedor:

```
root@89c3a9e7ae97:/# cd opt/  
root@89c3a9e7ae97:/opt# mkdir carpetaPrueba  
root@89c3a9e7ae97:/opt# ls  
carpetaPrueba  
root@89c3a9e7ae97:/opt#
```

Comprobación de la creación de la carpeta en la otra terminal:

```
root@89c3a9e7ae97:/# ls /opt  
carpetaPrueba  
root@89c3a9e7ae97:/#
```

Para subir la imagen, debemos logarnos en Docker desde la terminal:

```
C:\Users\agarg>docker login  
Login with your Docker ID to push an image to Docker Hub or to a self-hosted registry.  
ub.docker.com to create one.  
Username: algargon99  
Password:  
Login Succeeded  
  
Logging in with your password grants you access to all repositories owned by your user.  
For better security, log in with a personal access token.  
See https://docs.docker.com/docker-hub/access-tokens/  
C:\Users\agarg>
```

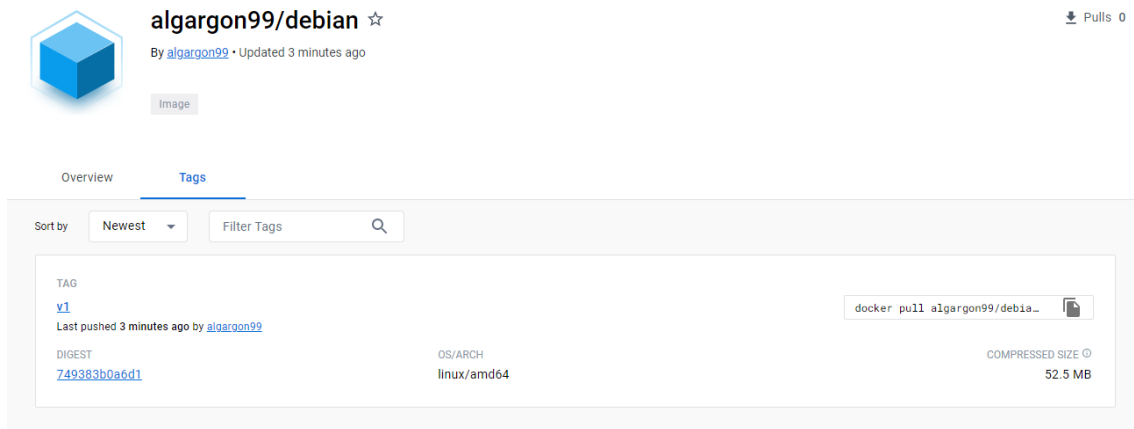
Ya logados, establecemos una etiqueta a la imagen del contenedor:

```
C:\Users\agarg>docker tag debian algargon99/debian:v1  
C:\Users\agarg>
```

Y la subimos a nuestro usuario de DockerHub con el nombre del tag:

```
C:\Users\agarg>docker push algargon99/debian:v1
The push refers to repository [docker.io/algargon99/debian]
8e396a1aad50: Mounted from library/debian
v1: digest: sha256:749383b0a6d17fb745d397b108d2ea38b5832226586b25c9f5cf7fcde24458ab size: 529
```

En DockerHub queda subida la imagen:



algargon99/debian ☆ Pulls 0

By algargon99 · Updated 3 minutes ago

Image

Overview Tags

Sort by Newest Filter Tags

TAG		
v1	Last pushed 3 minutes ago by algargon99	docker pull algargon99/debian:v1
DIGEST	OS/ARCH	COMPRESSED SIZE
749383b0a6d1	linux/amd64	52.5 MB

La url de acceso a DockerHub para obtener la imagen es:

<https://hub.docker.com/r/algargon99/debian/tags>

Parada del contenedor:

```
C:\Users\agarg>docker stop test1
test1
```

Borrado del contenedor:

```
C:\Users\agarg>docker rm test1
test1
```

Problema 1

Este problema se solucionará por completo en la carpeta P1.

Para su resolución, se utilizará un archivo docker-compose.yml, el cual contendrá la configuración correspondiente a los contenedores solicitados.

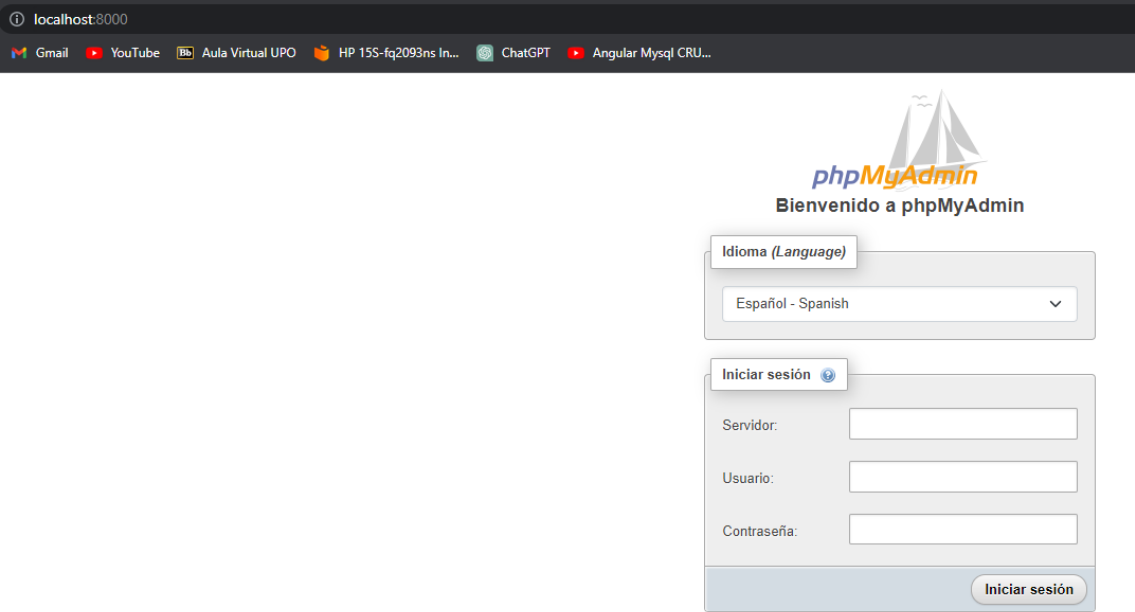
El contenido del archivo es el siguiente:

```
1  version: '3.1'
2
3  services:
4    mysqldb:
5      image: mysql
6      container_name: db
7      environment:
8        MYSQL_ROOT_PASSWORD: eps
9        MYSQL_DATABASE: usuarios
10       MYSQL_USER: eps
11       MYSQL_PASSWORD: eps
12     ports:
13       - "6033:3306"
14     volumes:
15       - dbdata:/var/lib/mysql
16   phpmyadmin:
17     image: phpmyadmin/phpmyadmin
18     container_name: pma
19     links:
20       - mysqldb
21     environment:
22       PMA_HOST: db
23       PMA_PORT: 3306
24       PMA_ARBITRARY: 1
25     restart: always
26     ports:
27       - 8000:80
28   volumes:
29     dbdata:
```

Como imágenes se utilizan mysql como servidor de base de datos, y phpmyadmin/phpmyadmin como dashboard para controlar dicho servidor.

Se establecen las variables de entorno de los servidores, de tal manera que phpmyadmin integrará el servidor mysql anteriormente creado, enlazando puerto y nombre del contenedor.

Al desplegar dicho docker-compose, y tratar de mirar la dashboard, usando localhost:8080 (puerto descrito en el docker-compose), el resultado es el siguiente:



Aquí se ven los contenedores funcionando:

```
PS C:\Users\agang> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a2cfd6045172	phpmyadmin/phpmyadmin	"/docker-entrypoint..."	About a minute ago	Up 59 seconds	0.0.0.0:8000->80/tcp	pma
Administrar 0592	mysql	"docker-entrypoint.s..."	About a minute ago	Up 59 seconds	33060/tcp, 0.0.0.0:6033->3306/tcp	db

Problema 2

Para crear este Docker orquestado, se precisan, aparte de un docker-compose con tres contenedores (apache con php, mysql, y phpmyadmin)

Todos los archivos se encuentran dentro de la carpeta P2