

docker

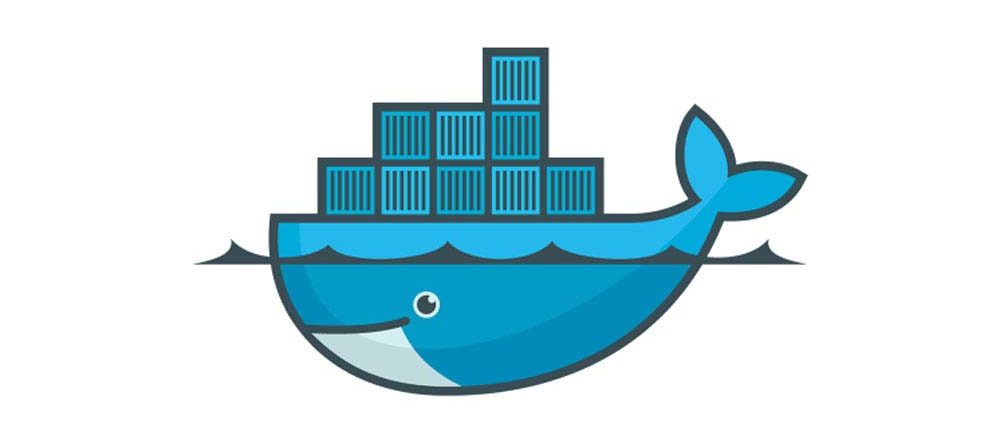
Sistemas Informáticos (1º DAM)



[Fecha]

[Nombre de la compañía]

[Dirección de la compañía]



Índice

Contenido

[1. Actividad 1 2](#_Toc186726797)

[1.1. Descargar imágenes de Docker Hub. 2](#_Toc186726798)

[1.2. Mostrar imágenes 4](#_Toc186726799)

[1.3. Contenedor con Ubuntu 18.04, nombre “ubuntu” 5](#_Toc186726800)

[1.3.1. Parar contenedor 5](#_Toc186726801)

[1.3.2. Rearrancar contenedor 5](#_Toc186726802)

[1.3.3. Mostrar fichero sin entrar al contenedor 6](#_Toc186726803)

[2. Actividad 2 7](#_Toc186726804)

[2.1. Contenedor con PHP, nombre “web” y puerto 8181 7](#_Toc186726805)

[2.2. Colocar ficheros index.html e index.php en el directorio web del contenedor 7](#_Toc186726806)

[2.3. Contenedor con mariadb con diferentes parámetros 8](#_Toc186726807)

[2.4. Conectar con un cliente de base de datos 8](#_Toc186726808)

[3. Gestión de imágenes 10](#_Toc186726809)

[3.1. Descargar imagen Ubuntu:20.04 10](#_Toc186726810)

[3.2. Volcar la información de la nueva imagen en un archivo 10](#_Toc186726811)

[3.3. Crear contenedor con Ubuntu:20.04 llamada “modulo3” 11](#_Toc186726812)

[3.4. Borrar contenedor con Ubuntu:20.04 11](#_Toc186726813)

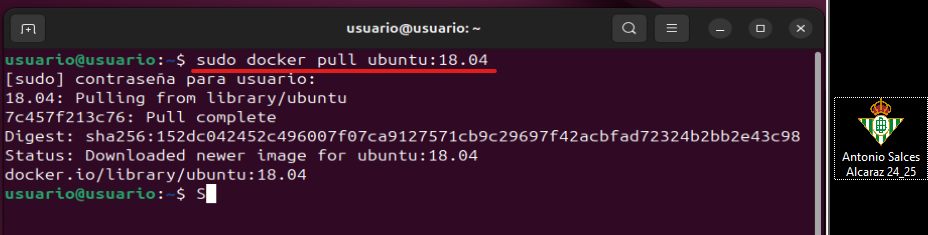
[3.5. Hacer acciones necesarias para borrar el contenedor Ubuntu:20.04 11](#_Toc186726814)

# Actividad 1

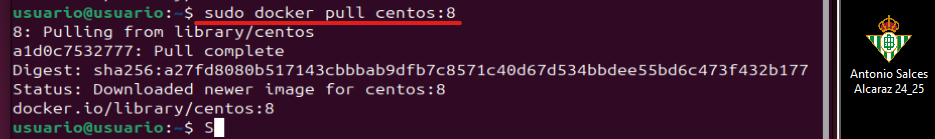
## Descargar imágenes de Docker Hub.

Para descargar los dockers, debemos de utilizar el comando “***docker pull <imagen[:versión]>***.Por defecto, cogerá la versión “latest” si no ponemos versión.

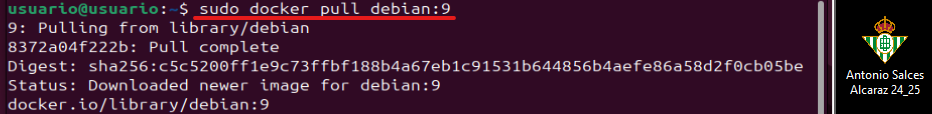
Ubuntu:18.04



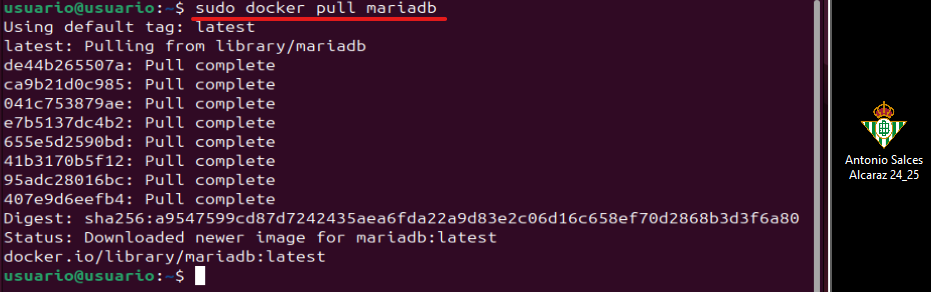
Centos:8



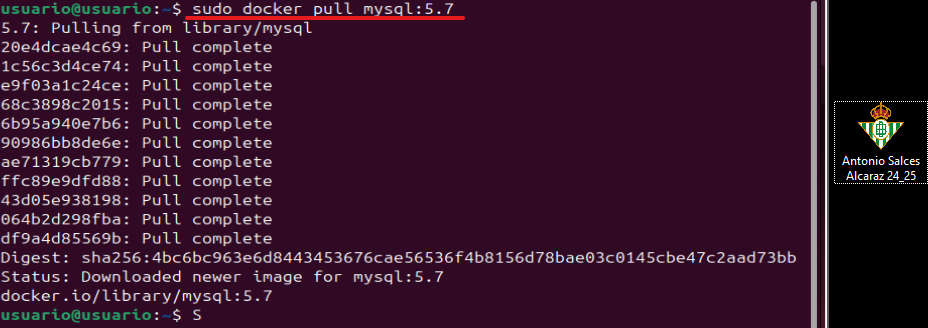
Debian: 9



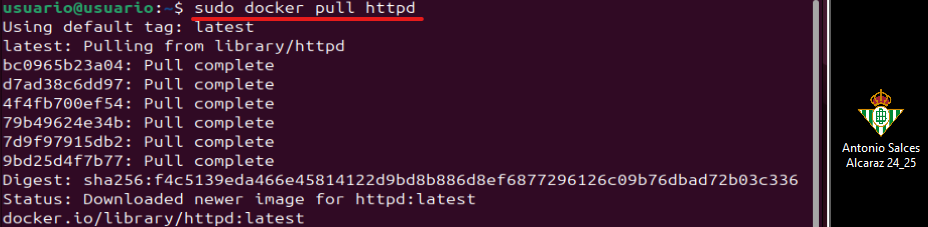
MariaDB:latest



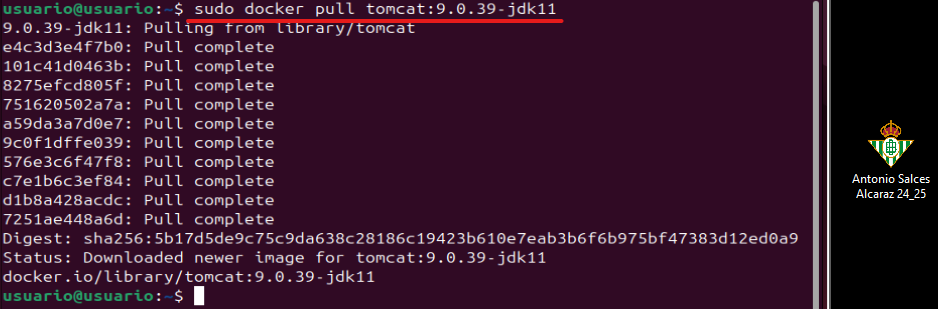
MySQL:5.7



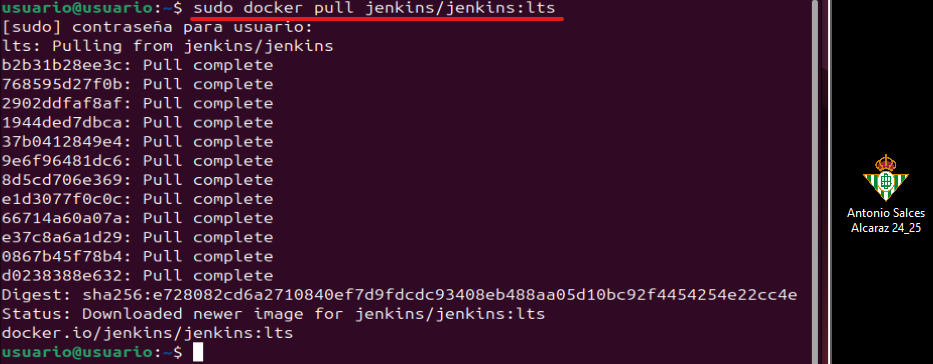
httpd



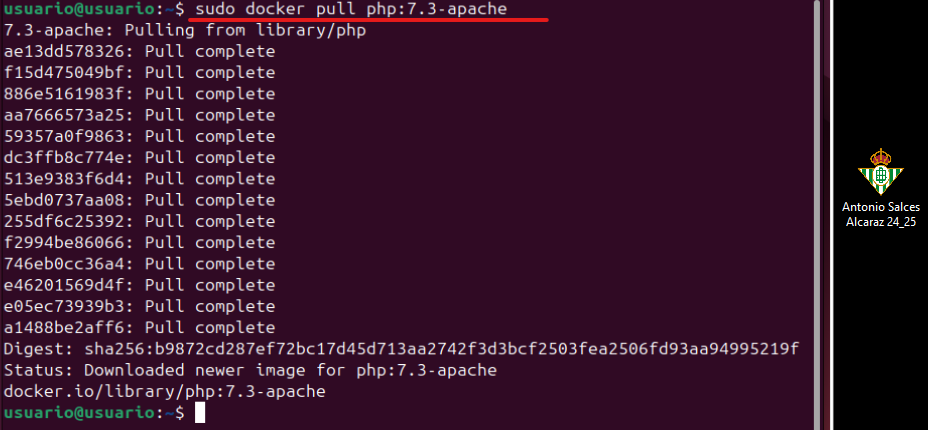
tomcat:9.0.39-jdk11



jenkins/jenkins:lts



php:7.3-apache



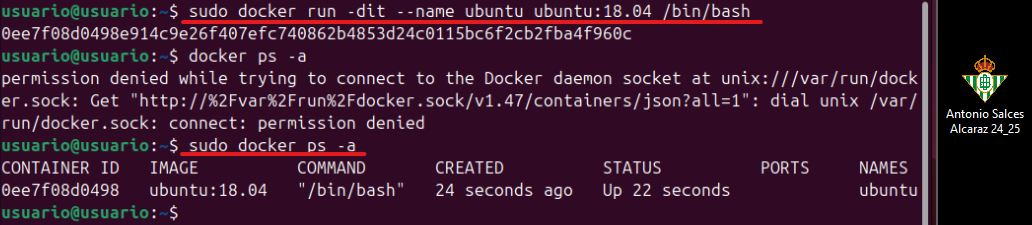
## Mostrar imágenes

Para hacer esto, debemos de hacer uso del comando “***docker images***”



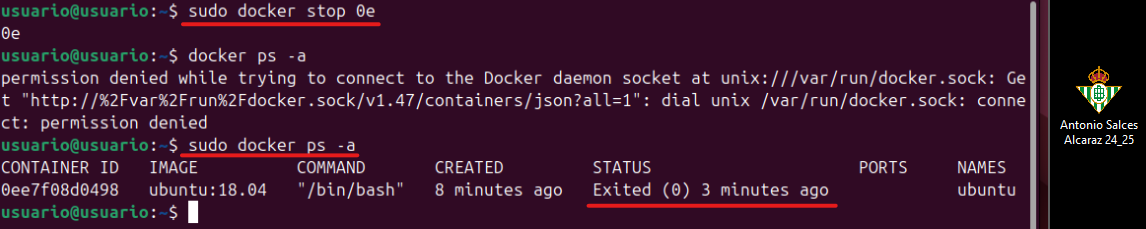
## Contenedor con Ubuntu 18.04, nombre “ubuntu”

Para arrancar el contenedor y darle un nombre, utilizaremos el comando “***docker run [-dit] [--name <nombre>] <imagen:versión> [comando]***”. También utilizaremos el comando “***docker ps -a***” para ver todos los contenedores del sistema y asi poder saber el ID del docker de ubuntu.



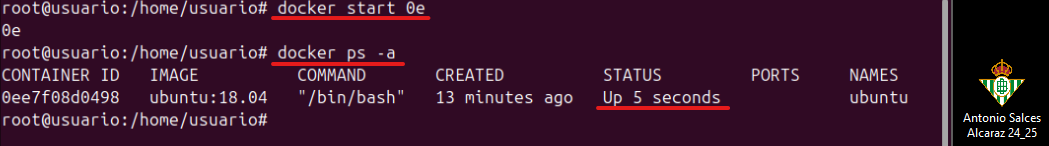
### Parar contenedor

Para parar el contenedor, ejecutaremos el comando “***docker stop <ID>***”. Basta con poner el inicio del ID, no es necesario ponerlo entero. Para comprobar que está parado, basta con poner “***docker ps -a***” y comprobar el estado, donde se ve que está parado.



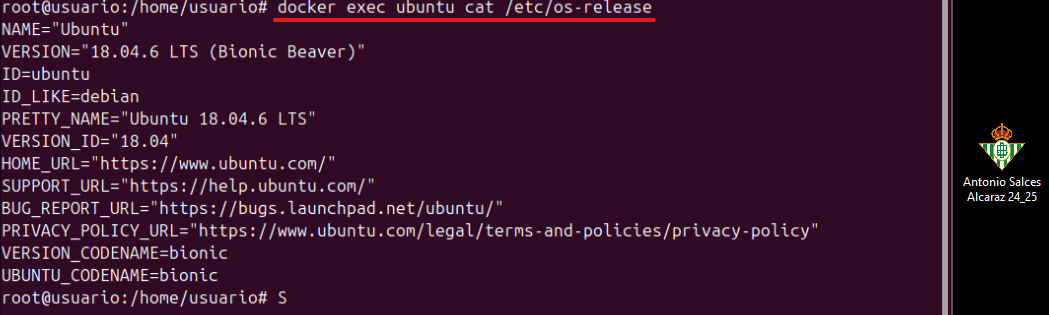
### Rearrancar contenedor

Para rearrancar el contenedor, utilizaremos “***docker start <ID>***”, y para comprobar si se ha iniciado, “***docker ps -a***”.



### Mostrar fichero sin entrar al contenedor

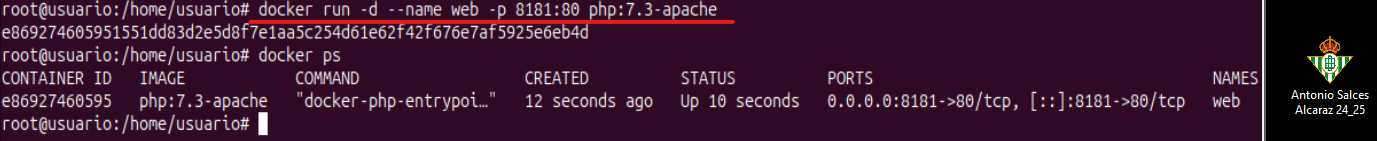
Para ello, debemos utilizar el comando “***docker exec <nombre> cat /etc/os-release***”. Docker exec nos permite ejecutar comandos del contenedor desde fuera del mismo, mientras que “cat” se utiliza para visualizar archivos. En lugar del nombre, también podemos poner el ID.



# Actividad 2

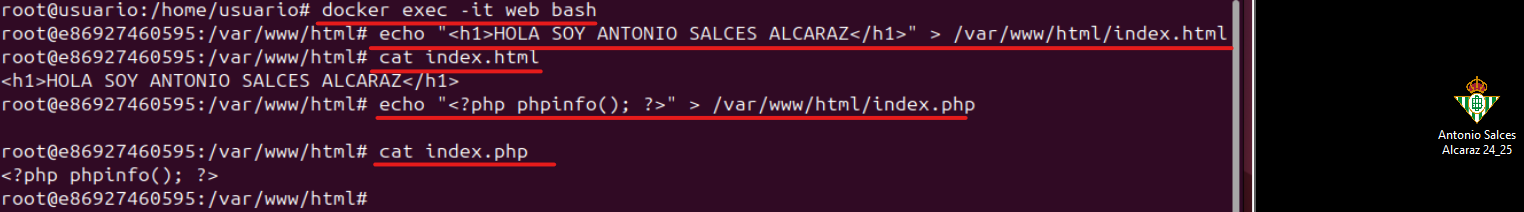
## Contenedor con PHP, nombre “web” y puerto 8181

Para iniciar el contenedor, utilizaremos el comando “***docker run [-d] [--name <nombre>] [-p <puerto>] <imagen>***”. Con “-d” ejecutaremos el docker en segundo plano, y con “-p” podremos elegir el puerto para el docker, en este caso el 8181.



## Colocar ficheros index.html e index.php en el directorio web del contenedor

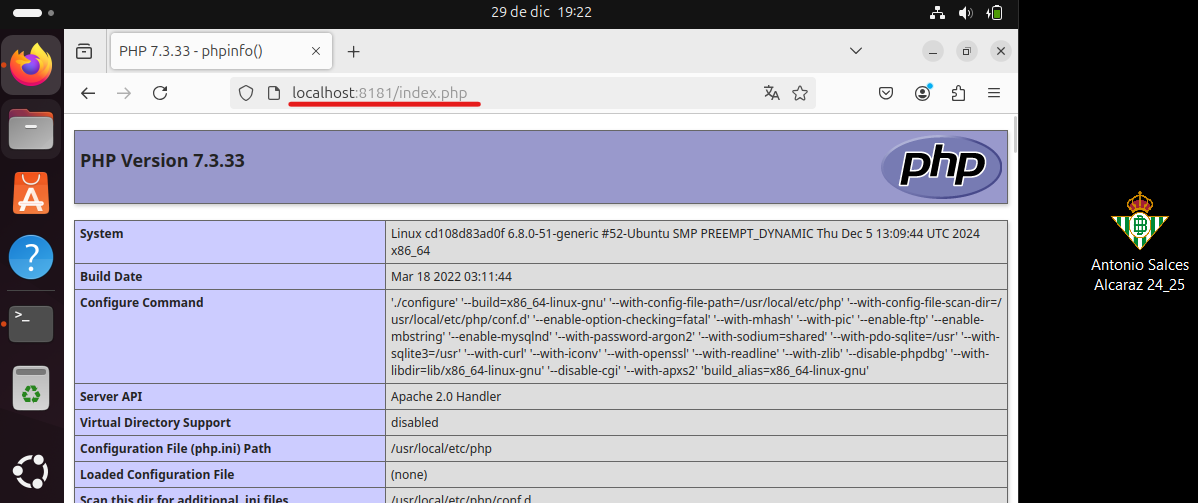
En primer lugar, accederemos al directorio web del contenedor con el primer comando que se ve en la imagen (“***docker exec -it [nombre o ID] bash***”). Una vez estemos en el contenedor, utilizaremos “***echo <contenido del archivo> > <ubicación del archivo>***”. Para comprobar el contenido del archivo, podemos utilizar “***cat <nombre>***”. Haremos lo mismo con el fichero “index.php”.



Si buscamos la siguiente dirección en la maquina virtual, podremos ver el fichero index.html que hemos creado.



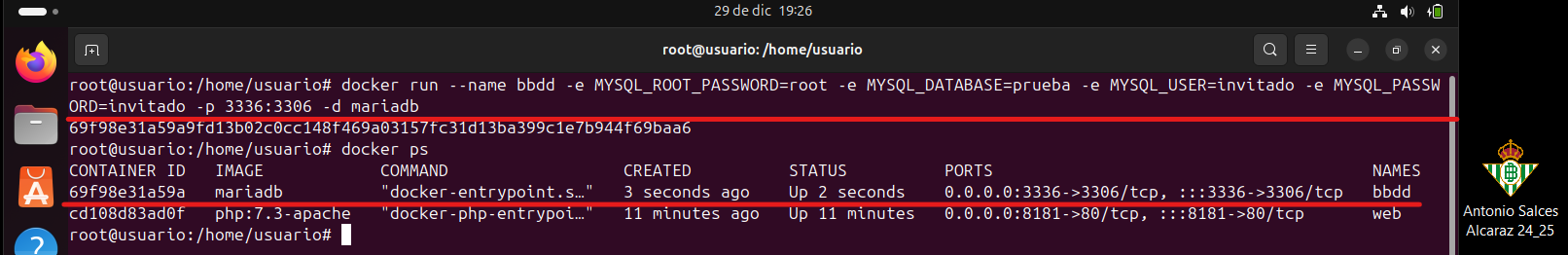
Y esto saldrá si buscamos el fichero index.php.



## Contenedor con mariadb con diferentes parámetros

Debemos arrancar un contenedor que contenga “mariadb” con los siguientes parámetros:

* Nombre: bbdd.
* Puerto: 3336.
* Contraseña de root: root.
* Crear base de datos con nombre: prueba.
* Contraseña del usuario: usuario.

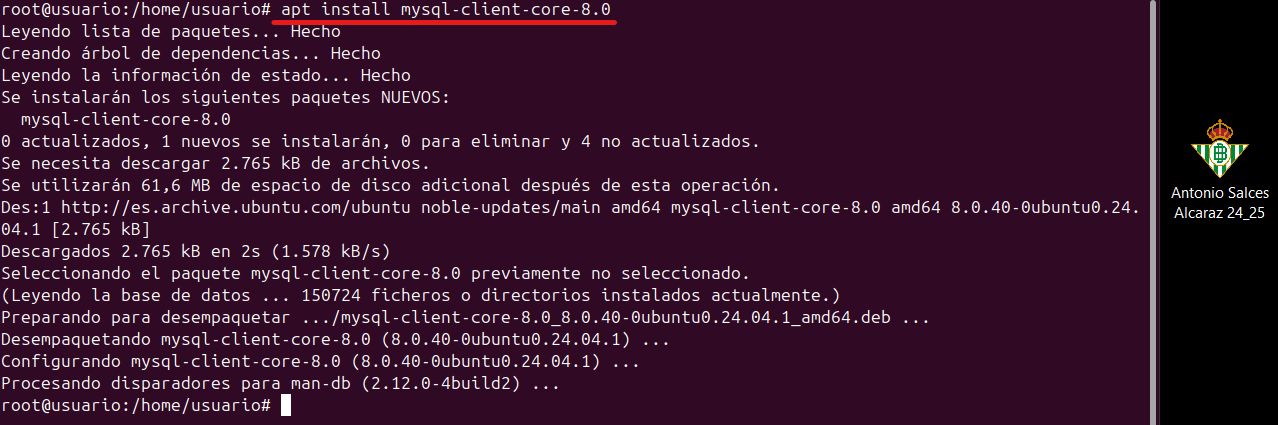


Los parámetros que se han utilizado son los siguientes:

* --name bbdd: el nombre del docker es “bbdd”.
* -e MYSQL\_ROOT\_PASSWORD=root: la constraseña del root será “root”.
* -e MYSQL\_DATABASE=prueba: crea la base de datos llamada “prueba”.
* -e MYSQL\_USER=invitado: crea el usuario "invitado".
* -e MYSQL\_PASSWORD=invitado: la contraseña del usuario “invitado” será “invitado”.
* -p 3336:3306: indica el puerto a utilizar.
* -d mariadb: lanzaremos “mariadb” en segundo plano.

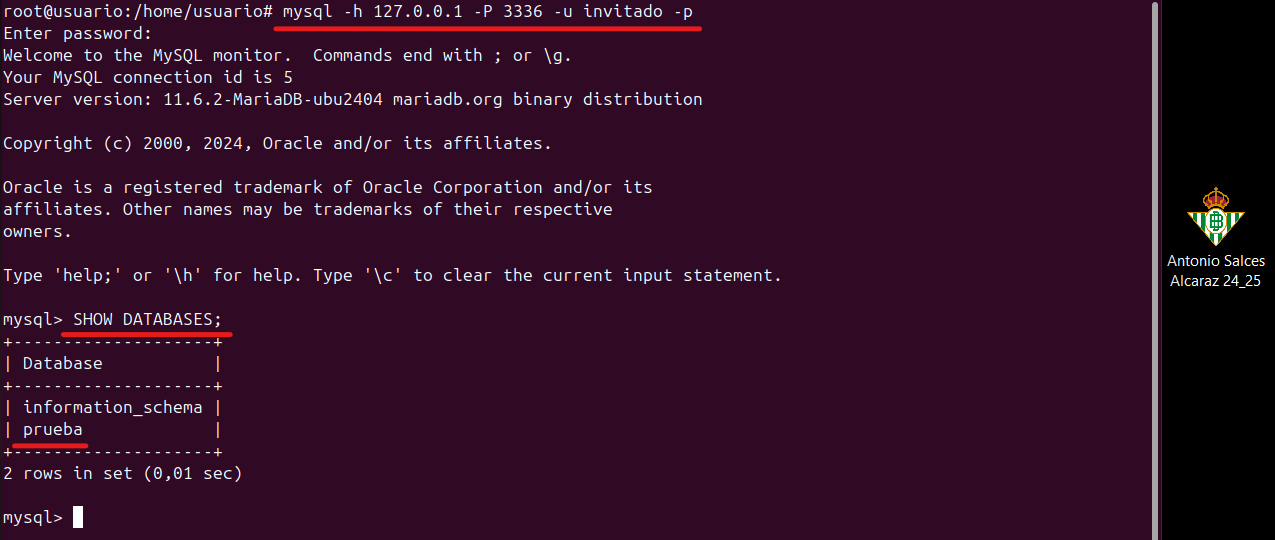
## Conectar con un cliente de base de datos

En primer lugar, debemos de instalar un cliente de base de datos. En mi caso, utilizaré un cliente de terminal (mysql-client). Utilizaremos el siguiente comando para instalarlo: “***apt install <paquete>***”.



Con el primer comando conseguiremos conectarnos con la base de datos del contenedor. El comando es el siguiente: “***mysql <-h IP> <-P puerto> <-u usuario> [-p]***”. “-p” indica que tenemos que introducir una contraseña para entrar.

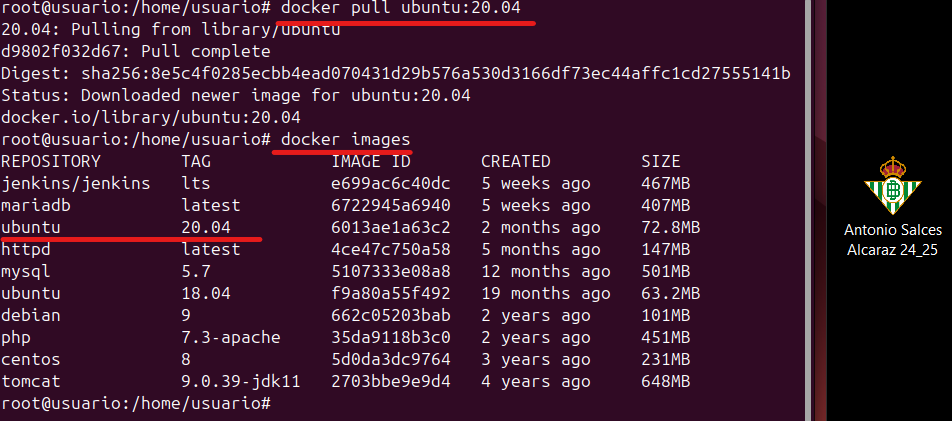
Una vez hayamos conectado con la base de datos, podemos utilizar el comando “***SHOW DATABASES;***” para ver las bases de datos creadas. Podemos ver en la imagen que la base de datos “prueba” se ha creado.



# Gestión de imágenes

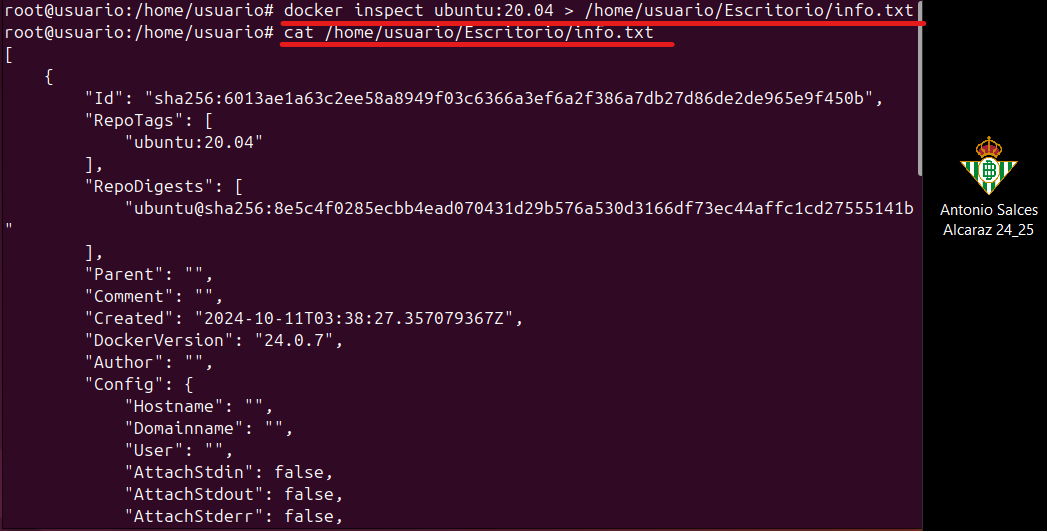
## Descargar imagen Ubuntu:20.04

Para descargar la imagen, utilizaremos el comando “***docker pull ubuntu:20.04***”, y una vez se haya completado la descarga, para comprobar que está en nuestro sistema, utilizaremos el comando “***docker images*”.**



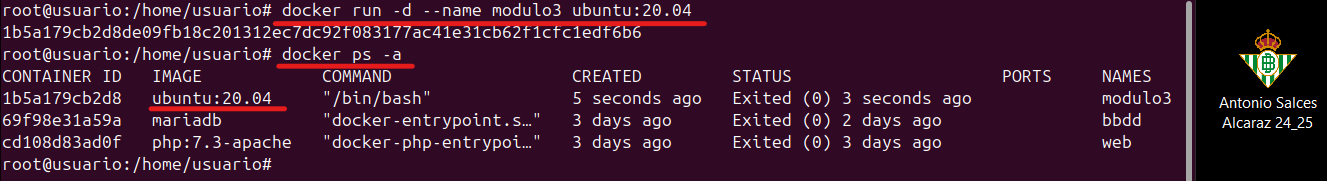
## Volcar la información de la nueva imagen en un archivo

Para ver la información de la nueva imagen y volcarla en un archivo, utilizaremos el comando “***docker inspect <imagen> > <ruta>***”. Para ver la información guardada en el archivo, utilizamos el comando “***cat <ruta>***”.



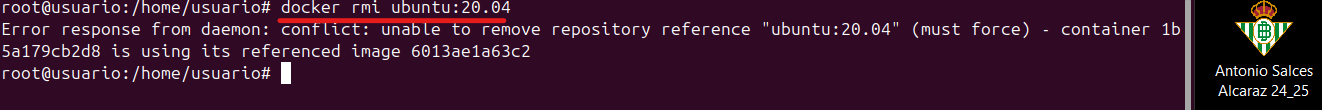
## Crear contenedor con Ubuntu:20.04 llamada “modulo3”

Utilizaremos el comando “***docker run -d –name modulo3 ubuntu:20.04***” para crear el contenedor con los parámetros especificados. Con el comando “***docker ps -a***” veremos el contenedor.



## Borrar contenedor con Ubuntu:20.04

Para borrar una imagen, utilizaremos el comando “***docker rmi <imagen>***”. Como podemos ver, no podemos borrar la imagen. Esto es debido a que la imagen se está utilizando en un contenedor, por lo que no podremos borrarla mientras el contenedor exista.



## Hacer acciones necesarias para borrar el contenedor Ubuntu:20.04

Para borrar la imagen, en primer lugar debemos parar el contenedor con “***docker stop <contenedor>***”, luego borrar con “***docker rm <contenedor>***”, y por ultimo borrar la imagen con “***docker rmi <imagen>***”.

