

## 1- Instalar Docker Community Edition

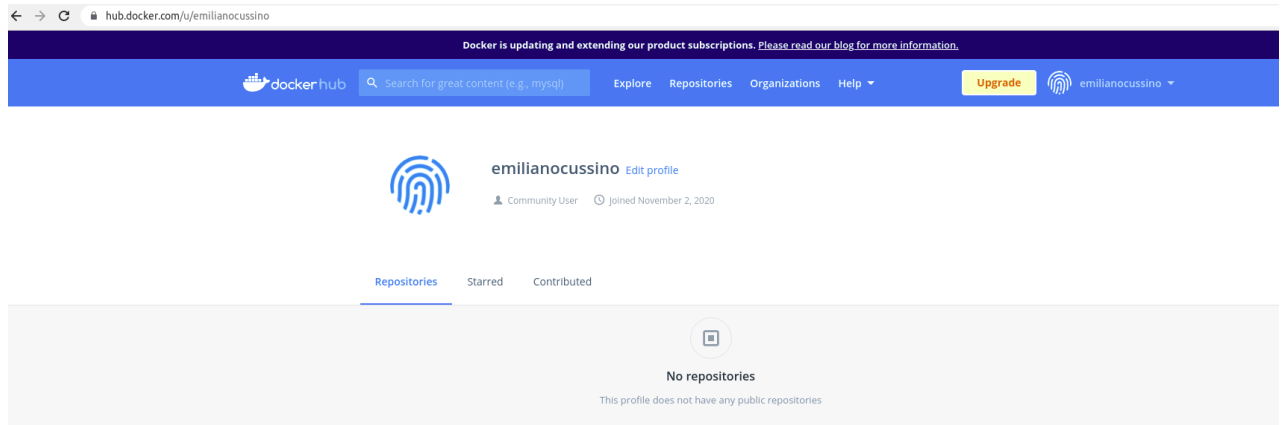
- Diferentes opciones para cada sistema operativo
- <https://docs.docker.com/>
- Ejecutar el siguiente comando para comprobar versiones de cliente y demonio.

```
emiliano@emiliano-HP-Pavilion-Notebook:~$ docker version
Client: Docker Engine - Community
Version: 20.10.8
API version: 1.41
Go version: go1.16.6
Git commit: 3967b7d
Built: Fri Jul 30 19:54:27 2021
OS/Arch: linux/amd64
Context: default
Experimental: true

Server: Docker Engine - Community
Engine:
Version: 20.10.8
API version: 1.41 (minimum version 1.12)
Go version: go1.16.6
Git commit: 75249d8
Built: Fri Jul 30 19:52:33 2021
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: 1.4.9
GitCommit: e25210fe30a0a703442421b0f60afac609f950a3
runc:
Version: 1.0.1
GitCommit: v1.0.1-0-g4144b63
docker-init:
Version: 0.19.0
GitCommit: de40ad0
```

## 2- Explorar DockerHub

- Regístrate en docker hub: <https://hub.docker.com/>
- Familiarizarse con el portal



## 3- Obtener la imagen BusyBox

- Ejecutar el siguiente comando, para bajar una imagen de DockerHub

```
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
8ec32b265e94: Pull complete
Digest: sha256:b37dd066f59a4961024cf4bed74cae5e68ac26b48807292bd12198afa3ecb778
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
```

- Verificar qué versión y tamaño tiene la imagen bajada, obtener una lista de imágenes locales:

```
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
busybox	latest	42b97d3c2ae9	2 weeks ago	1.24MB
jupyter/pyspark-notebook	latest	67bc3e9ab3da	11 months ago	3.25GB
mysql/mysql-server	5.7	9c31a29b3f30	14 months ago	322MB
mysql/mysql-server	latest	8a3a24ad33be	14 months ago	366MB
magn3333/pentahoserver	latest	57e6978b559f	14 months ago	4.49GB
mysql	5.7.27	383867b75fd2	24 months ago	373MB
mysql	8.0.17	b8fd9553f1f0	24 months ago	445MB
workbenchdata/workbenchdata	latest	64e0aa724feb	3 years ago	945MB

Es la ultima version de busybox y tienen un tamaño de 1.24 MB

## 4- Ejecutando contenedores

- Ejecutar un contenedor utilizando el comando run de docker:

```
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker images
REPOSITORY          TAG         IMAGE ID      CREATED       SIZE
busybox              latest      42b97d3c2ae9 2 weeks ago   1.24MB
jupyter/pyspark-notebook latest      67bc3e9ab3da 11 months ago 3.25GB
mysql/mysql-server   5.7         9c31a29b3f30 14 months ago 322MB
mysql/mysql-server   latest      8a3a24ad33be 14 months ago 366MB
magn3333/pentahoserver latest      57e6978b559f 14 months ago 4.49GB
mysql                5.7.27     383867b75fd2 24 months ago 373MB
mysql                8.0.17     b8fd9553f1f0 24 months ago 445MB
workbenchdata/workbenchdata latest      64e0aa724feb 3 years ago   945MB
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker run busybox
```

- Explicar porque no se obtuvo ningún resultado

No se obtienen resultados porque la imagen “busybox” esta descargada en el equipo pero no se ha construido, no se hizo un build sobre la imagen

- Especificamos algún comando a correr dentro del contenedor, ejecutar por ejemplo:

```
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker run busybox echo "Hola Mundo"
Hola Mundo
```

- Ver los contenedores ejecutados utilizando el comando ps:

```
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS    NAMES
```

- Vemos que no existe nada en ejecución, correr entonces:

```
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS    NAMES
8d0befe198a    busybox    "echo 'Hola Mundo'"      2 minutes ago    Exited (0) 2 minutes ago    beautiful_khorana
d85f4bf7c489    busybox    "sh"                      7 minutes ago    Exited (0) 7 minutes ago    goofy_taussig
69f3387ac07f    busybox    "sh"                      15 minutes ago    Exited (0) 15 minutes ago    gifted_dirac
c6bee94ed4bf    57e6978b559f    "/opt/pentaho-server..." 10 months ago    Exited (137) 10 months ago    charming_matsumoto
3e2f245900c9    magn3333/pentahoserver    "/opt/pentaho-server..." 10 months ago    Exited (137) 10 months ago    pentahoserver2
52e8502555d6    magn3333/pentahoserver    "/opt/pentaho-server..." 10 months ago    Created    pentahoserver
a3242db6cd47    mysql:8.0.17    "docker-entrypoint.s..." 10 months ago    Exited (0) 10 months ago    emiliano_mysql-produccion_1
2a2b73ce653a    64e0aa724feb    "sh -c 'xvfb-run -a ...'" 10 months ago    Exited (1) 10 months ago    happy_burnell
c48a5b5d89a1    mysql:5.7.27    "docker-entrypoint.s..." 10 months ago    Exited (0) 10 months ago    mysql57
5262eeb04e6a    9c31a29b3f30    "/entrypoint.sh mysql..." 10 months ago    Exited (0) 10 months ago    wonderful_jepsen
c4907a894c8d    mysql/mysql-server:5.7    "/entrypoint.sh mysql..." 12 months ago    Exited (0) 12 months ago    mysql-container
cf45de146ae3    mysql/mysql-server:5.7    "/entrypoint.sh mysql..." 12 months ago    Exited (0) 12 months ago    mysql-container
```

Con el comando “docker ps -a” listamos las imágenes que existen en el equipo, las que están corriendo o estén detenidas, el “-a” hace referencia a todas las imágenes.

## 5- Ejecutando en modo interactivo

- Ejecutar el siguiente comando
- Para cada uno de los siguientes comandos dentro de contenedor, mostrar los resultados:

```
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker run -it busybox sh
/ # ps
PID      USER     TIME     COMMAND
   1      root         0:00    sh
   7      root         0:00    ps
/ # uptime
16:34:33 up 2:21,  0 users,  load average: 1.04, 1.10, 1.19
/ # free
              total        used        free      shared  buff/cache   available
Mem:      12104492       2042340       348816        534148        9713336       9205556
Swap:      2097148           256       2096892
/ # ls -l /
total 36
drwxr-xr-x  2 root    root          12288 Aug 20 16:21 bin
drwxr-xr-x  5 root    root           360 Sep  6 16:33 dev
drwxr-xr-x  1 root    root          4096 Sep  6 16:33 etc
drwxr-xr-x  2 nobody  nobody        4096 Aug 20 16:21 home
dr-xr-xr-x 280 root    root           0 Sep  6 16:33 proc
drwx----- 1 root    root          4096 Sep  6 16:34 root
dr-xr-xr-x 13 root    root           0 Sep  6 16:33 sys
drwxrwxrwt  2 root    root          4096 Aug 20 16:21 tmp
drwxr-xr-x  3 root    root          4096 Aug 20 16:21 usr
drwxr-xr-x  4 root    root          4096 Aug 20 16:21 var
/ #
```

## 6- Borrando contenedores terminados

- Obtener la lista de contenedores
- Para borrar podemos utilizar el id o el nombre (autogenerado si no se especifica) de contendor que se desee, por ejemplo:

```
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
839fc9d96537   busybox       "sh"                    3 minutes ago Exited (0)    41 seconds ago         serene_bose
8d0belfe198a   busybox       "echo 'Hola Mundo'"     9 minutes ago Exited (0)    9 minutes ago         beautiful_khorana
d85f4bf7c489   busybox       "sh"                    14 minutes ago Exited (0)    14 minutes ago         goofy_taussig
69f3387ac07f   busybox       "sh"                    22 minutes ago Exited (0)    22 minutes ago         gifted_dirac
c6bee94ed4bf   57e6978b559f "/opt/pentaho-server..." 10 months ago Exited (137) 10 months ago         charming_matsumoto
3e2f245900c9   magm3333/pentahoserver "/opt/pentaho-server..." 10 months ago Exited (137) 10 months ago         pentahoserver2
52e8502555d6   magm3333/pentahoserver "/opt/pentaho-server..." 10 months ago Created                                     pentahoserver
a3242db6cd47   mysql:8.0.17  "docker-entrypoint.s..." 10 months ago Exited (0)    10 months ago         emiliano_mysql-produccion_1
2a2b73ce653a   64e0aa724feb  "sh -c 'xvfb-run -a ..." 10 months ago Exited (1)    10 months ago         happy_burnell
cd8a5b5d89a1   mysql:5.7.27  "docker-entrypoint.s..." 10 months ago Exited (0)    10 months ago         mysql57
5262eeb04e6a   9c31a29b3f30  "/entrypoint.sh mysql..." 10 months ago Exited (0)    10 months ago         wonderful_jepsen
c4907a894c8d   mysql/mysql-server:5.7 "/entrypoint.sh mysql..." 12 months ago Exited (0)    12 months ago         mysql-container
cf45de146ae3   mysql/mysql-server:5.7 "/entrypoint.sh mysql..." 12 months ago Exited (0)    12 months ago         mysql-container
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker rm 839fc9d96537
839fc9d96537
```

- Para borrar todos los contenedores que no estén corriendo, ejecutar cualquiera de los siguientes comandos:

```
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
8d0be1fe198af8c3d09c443a2e3ec052cbc2bce1a51b9f7456b7ccbbcbba5294d
d85f4bf7c489b8fef2baeac14613fa84335622fef33ca7fa5ee38bd4f1c7e68
69f3387ac07f2bd08bf5af8ab60cc81c6555c57fa46a786f2112eea579bb095b
c6bee94ed4bf397a5864462cb735a45cb1c4664f73960fa0e9c2134b184a9c77
3e2f245900c9dada81f36540c93081cd24465dd7110d912b7a85c585f86c5ad4
52e8502555d6019e2313384708cad9b5e7ffcc5c1a4ed2f7af1cb9529eb6f210
a3242db6cd476f302e4f4cbe4c36a114e8041fec1c81d9a840513d33faef5a34
2a2b73ce653a7aad91e0f2eec6508aac1eccb26d16299441902a2d0ac66e11b4
cd8a5b5d89a1d07374a374b7ad8ecb09222116b54ff0884ef6764e47803fe546
5262eeb04e6a27a01004a6b8ace924b03d6f9521725a8865cbce6461879d2788
c4907a894c8d6a5988673e2e3427378bcbf529b0fe716725b5f8222d4ab0203d
cf45de146ae3a15d22c75d9026272aac202bd1d6b7ecef7edf43323939d13f2

Total reclaimed space: 977.1MB
emiliano@emiliano-HP-Pavilion-Notebook:~/Escritorio/Facultad/IngSoft3/Practico 2$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8d0be1fe198af8c3d09c443a2e3ec052cbc2bce1a51b9f7456b7ccbbcbba5294d	alpine:3.10.1	sleep 10	2 minutes ago	Exited (0)		alpine-3.10.1

## 7- Montando volúmenes

Hasta este punto los contenedores ejecutados no tenían contacto con el exterior, ellos corrían en su propio entorno hasta que terminaran su ejecución. Ahora veremos cómo montar un volumen dentro del contenedor para visualizar por ejemplo archivos del sistema huésped:

- Ejecutar el siguiente comando, cambiar myusuario por el usuario que corresponda. En linux/Mac puede utilizarse /home/miusuario):
- Dentro del contenedor correr
- Verificar que el Archivo se ha creado en el escritorio o en el directorio home según corresponda.

```
emiliano@emiliano-HP-Pavilion-Notebook:~$ docker run -it -v \home\emiliano:/var/escritorio busybox /bin/sh
/ # ls -l /var/escritorio
total 0
/ # touch /var/escritorio/hola.txt
/ # ls -l /var/escritorio
total 0
-rw-r--r-- 1 root root 0 Sep 6 20:17 hola.txt
/ #
```

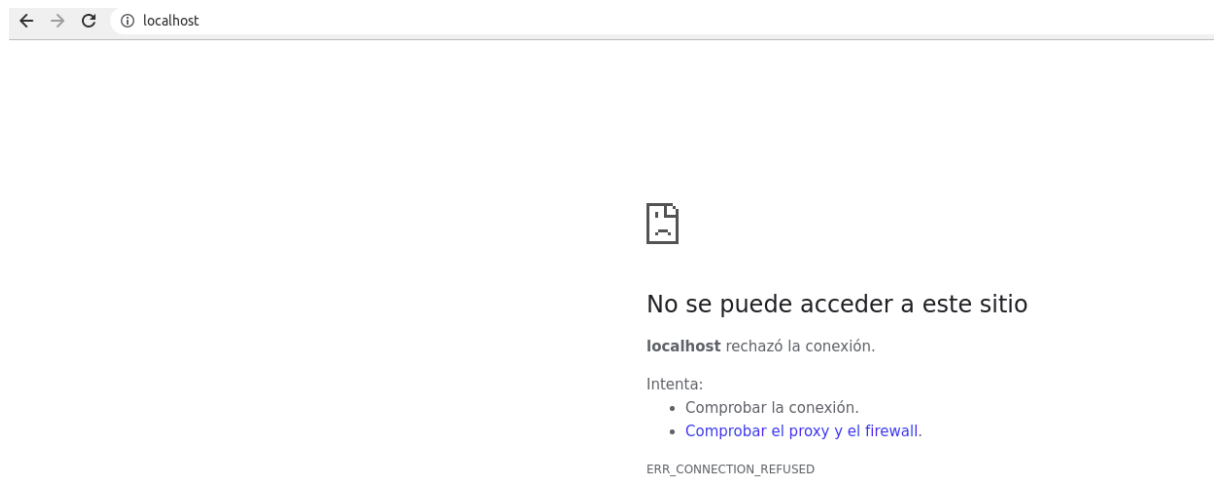
## 8- Publicando puertos

En el caso de aplicaciones web o base de datos donde se interactúa con estas aplicaciones a través de un puerto al cual hay que acceder, estos puertos están visibles sólo dentro del contenedor. Si queremos acceder desde el exterior deberemos exponerlos.

- Ejecutar la siguiente imagen, en este caso utilizamos la bandera -d (detach) para que nos devuelva el control de la consola:

```
emiliano@emiliano-HP-Pavilion-Notebook:~$ docker run -d daviey/nyan-cat-web
Unable to find image 'daviey/nyan-cat-web:latest' locally
latest: Pulling from daviey/nyan-cat-web
b7f33cc0b48e: Pull complete
5f9b58fd6dd4: Pull complete
1adeef8edfca: Pull complete
cc8a2986b124: Pull complete
7220539c61d6: Pull complete
Digest: sha256:57ac8fd383ada137e22a2894e92f74287f4566be0ae21ca97828b34a93a646c6
Status: Downloaded newer image for daviey/nyan-cat-web:latest
daacd5766db68af5e5f8ba71e9635eae9d74f8a2ad74559a5fdf904a1fd38b4e
emiliano@emiliano-HP-Pavilion-Notebook:~$ docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED        STATUS        PORTS                    NAMES
daacd5766db6   daviey/nyan-cat-web  "nginx -g 'daemon of..." 31 seconds ago Up 29 seconds 80/tcp, 443/tcp         wizardly_bose
```

- Vemos que el contenedor expone 2 puertos el 80 y el 443, pero si intentamos en un navegador acceder a <http://localhost> no sucede nada.



- Procedemos entonces a parar y remover este contenedor:

```
emiliano@emiliano-HP-Pavilion-Notebook:~$ docker kill daacd5766db6
daacd5766db6
emiliano@emiliano-HP-Pavilion-Notebook:~$ docker rm daacd5766db6
daacd5766db6
```

- Vamos a volver a correrlo otra vez, pero publicando uno de los puertos solamente, en este caso el 80 → lo cambie al puerto 81

```
emiliano@emiliano-HP-Pavilion-Notebook:~$ docker run -d -p 81:80 daviey/nyan-cat-web
9f3c6a4c0f6110be09f55d82b22142ede367e04aba266412ea9bc2bf4e8c587e
```



## 9- Utilizando una base de datos

- Levantar una base de datos PostgreSQL
- Ejecutar sentencias utilizando esta instancia

```
emiliano@emiliano-HP-Pavilion-Notebook: ~/.postgres$ docker exec -it my-postgres /bin/bash
root@a3537dd4dbfb:/# psql -h localhost -U postgres
psql (9.4.26)
Type "help" for help.

postgres=# \l
                                List of databases
  Name      | Owner   | Encoding | Collate | Ctype   | Access privileges
-----+-----+-----+-----+-----+-----
 postgres   | postgres | UTF8     | en_US.utf8 | en_US.utf8 | 
 template0  | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres
+
 gres       |          |          |          |          | postgres=Ctc/post
 template1  | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres
+
 gres       |          |          |          |          | postgres=Ctc/post
(3 rows)

postgres=# create database test;
CREATE DATABASE
postgres=# \connect test
You are now connected to database "test" as user "postgres".
test=# create table tabla_a (mensaje varchar(50));
CREATE TABLE
test=# insert into tabla_a (mensaje) values('hola mundo!');
INSERT 0 1
test=# select * from tabla_a
test-# ;
      mensaje
-----
 hola mundo!
(1 row)

test=# \q
root@a3537dd4dbfb:/# exit
exit
```

- Conectarse a la base utilizando alguna IDE (Dbeaver - <https://dbeaver.io/>, eclipse, IntelliJ, etc...). Interactuar con los objetos creados.
- Explicar que se logro con el comando `docker run` y `docker exec` ejecutados en este ejercicio.