



**UNIVERSIDAD  
CATÓLICA  
DE CÓRDOBA**  
JESUITAS

**INGENIERÍA DE SOFTWARE 3  
TRABAJO PRÁCTICO N°2**

**Profesor:** Ing. Bono.

**Alumno:** Menel Angelo (1804789)

- Se instaló docker y con **docker version** vemos toda la información de la version instalada de docker.

```

~ docker version
Client:
 Version:      20.10.17
 API version:  1.41
 Go version:   go1.18.3
 Git commit:   100c70180f
 Built:        Sat Jun 11 23:27:28 2022
 OS/Arch:     linux/amd64
 Context:      default
 Experimental: true

Server:
 Engine:
  Version:      20.10.17
  API version:  1.41 (minimum version 1.12)
  Go version:   go1.18.3
  Git commit:   a89b84221c
  Built:        Sat Jun 11 23:27:14 2022
  OS/Arch:     linux/amd64
  Experimental: false
 containerd:
  Version:      v1.6.6
  GitCommit:    10c12954828e7c7c9b6e0ea9b0c02b01407d3ae1.m
 runc:
  Version:      1.1.3
  GitCommit:
 docker-init:
  Version:      0.19.0
  GitCommit:    de40ad0

```

- Al correr **docker run busybox**, no se obtiene ningún resultado porque se debe instanciar un contenedor de esa imagen, una imagen es solamente un directorio con todo el programa.
- Cuando se ejecuta **docker ps -a**, nos devuelve los diferentes contenedores creados, devolviendo el ID del contenedor, la imagen que se usó, el status muestra si está levantado o no el contenedor, los puertos que utiliza y el nombre del contenedor (puede ser auto-generado o se le puede dar uno).

```

~ docker run busybox
~ docker run busybox echo "Hola Mundo"
Hola Mundo
~ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
~ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED        STATUS              PORTS   NAMES
85a242edc251   busybox   "echo 'Hola Mundo'"   20 seconds ago   Exited (0) 19 seconds ago           elastic_mahavira
6c032fbf933d   busybox   "sh"      48 seconds ago   Exited (0) 47 seconds ago           cranky_lichterman

```

- Con **`docker run -it {container_name} sh`** se usa para levantar el contenedor y entrar a la terminal del contenedor.

```

~ ➤ docker run -it busybox sh
/ # ps
PID   USER     TIME   COMMAND
    1  root      0:00   sh
    7  root      0:00   ps
/ # uptime
 17:38:23 up  3:38,  0 users,  load average: 1.26, 1.79, 1.60
/ # free
              total        used        free      shared  buff/cache   available
Mem:        16167184       3017244       9062456       1029836       4087484       11788616
Swap:           0           0           0
/ # ls -l
total 36
drwxr-xr-x  2 root    root          12288 Jul 29 01:32 bin
drwxr-xr-x  5 root    root           360 Aug 18 17:38 dev
drwxr-xr-x  1 root    root          4096 Aug 18 17:38 etc
drwxr-xr-x  2 nobody nobody        4096 Jul 29 01:32 home
dr-xr-xr-x 320 root    root           0 Aug 18 17:38 proc
drwx----- 1 root    root          4096 Aug 18 17:38 root
dr-xr-xr-x 13 root    root           0 Aug 18 17:38 sys
drwxrwxrwt  2 root    root          4096 Jul 29 01:32 tmp
drwxr-xr-x  3 root    root          4096 Jul 29 01:32 usr
drwxr-xr-x  4 root    root          4096 Jul 29 01:32 var

```

- Con **`docker rm {container_name}`** se elimina un contenedor.

```

~ ➤ docker rm elastic_mahavira
elastic_mahavira
~ ➤ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
5fe145cec7f5411f4c15672e67836d11a2a83478047d4526d3c53d165595bcb6
6c032fbfb933d8a25ad9455031f5d10dbc6b2f5886f074008aa36e2d741da69f5

Total reclaimed space: 26B

```

- **`docker pull {image_name}`** nos descarga una imagen de DockerHub.

```

~ ➤ docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
50783e0dfb64: Pull complete
Digest: sha256:ef320ff10026a50cf5f0213d35537ce0041ac1d96e9b7800bafd8bc9eff6c693
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
~ ➤ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
busybox       latest    7a80323521cc   2 weeks ago   1.24MB

```

- Cuando corremos el contenedor de nyan-cat y hacemos un `docker ps`, podemos observar los puertos que estan reservados para ese contenedor, pero no lo podemos acceder porque tenemos que configurar un puerto local que este escuchando al puerto reservado del contenedor.
- Con **`docker kill {container_name}`** detenemos el contenedor de forma forzosa.
- Con **`docker run -d -p 80:80 {container_name}`** usamos el `-d` para ... y el `-p` configuramos el puerto que va a estar escuchando al puerto reservado.

```

❯ 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
2327843e59f89ff6cfc21fd61c05f0149c4944785d5183b2e693842263f5612d
❯ docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED        STATUS        PORTS                    NAMES
2327843e59f8   daviey/nyan-cat-web  "nginx -g 'daemon of..." 11 seconds ago Up 9 seconds   80/tcp, 443/tcp         determined_golick
❯ docker kill determined_golick
determined_golick
❯ docker rm determined_golick
determined_golick
❯ docker run -d -p 80:80 daviey/nyan-cat-web
27037a0927bc3f728f20e18379337b4c890dc0c4f6cd795340c7a915905f6af8
❯ docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED        STATUS        PORTS                                NAMES
27037a0927bc   daviey/nyan-cat-web  "nginx -g 'daemon of..." 23 seconds ago Up 22 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp, 443/tcp  vibrant_ferml
❯ docker stop vibrant_ferml
vibrant_ferml
❯ docker ps
CONTAINER ID   IMAGE             COMMAND                  CREATED        STATUS        PORTS                    NAMES

```

- El `-v` sirve para vincular un directorio local a uno dentro del contenedor. En este ejemplo dentro del contenedor, en el escritorio creamos un archivo de texto “hola.txt” y como esta vinculado al escritorio local, se crea tambien el mismo archivo de texto. Si modificamos algo en el hola.txt del contenedor, se modifica localmente tambien.

```

❯ docker run -it -v /home/angelo/Escritorio:/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 Aug 18 18:00 hola.txt
/ #

```

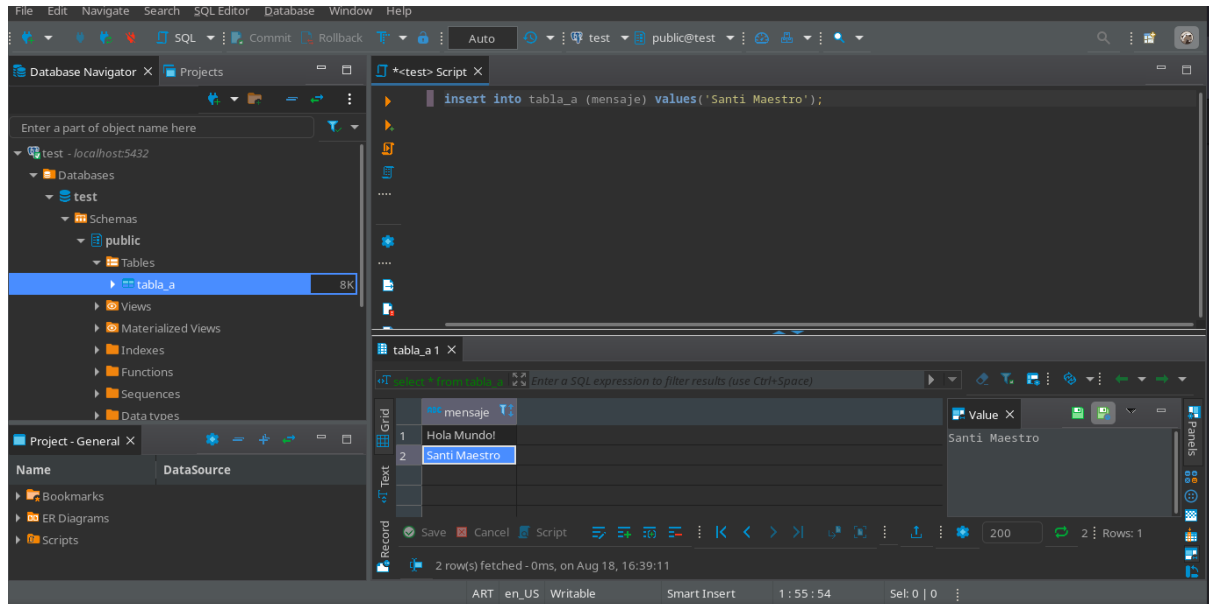
- En este ejercicio lo que hicimos fue crear una carpeta `.postgres` el cual vamos a usar para el volumen. Creamos un contenedor de postgres, donde usamos el `-e` para setear las password y username de la conexion. El `-p` para configurar el puerto que escuche al del contenedor.

```

❯ mkdir .postgres
❯ docker run --name my-postgres -e POSTGRES_PASSWORD=M4rv3lv5DC -v /home/angelo/.postgres:/var/lib/postgresql/data -p 5432:5432 -d postgres:9.4
Unable to find image 'postgres:9.4' locally
9.4: Pulling from library/postgres
619014d83c02: Pull complete
7ec0fe6664f6: Pull complete
9ca7ba8f7764: Pull complete
9e1155d037e2: Pull complete
febcb7f8870: Pull complete
8c78c79412b5: Pull complete
5a35744405c5: Pull complete
37717022e067: Pull complete
36f0c5255550: Pull complete
dbf0a396f422: Pull complete
ec4c06ea33e5: Pull complete
e8dd33eba6d1: Pull complete
51c81b3b2c20: Pull complete
2a03dd76f5d7: Pull complete
Digest: sha256:42a7a6a647a602efa9592edd1f5635980d079b93fa52c5d92244c58ac4a2ab9
Status: Downloaded newer image for postgres:9.4
36b1190118c7224efd70de2dc8f3c0bb62c6db04725b4955332dcf779561106a

```

- Usamos un cliente que pueda conectarse a postgres, en este caso es DBeaver. Para conectar el DBeaver al contenedor de postgres creado, necesitamos poner el host (127.0.0.1), el puerto (5432), el nombre del database creado (test), el username y password.



9 - Con **docker run** se levanta un contenedor, de manera que hasta que no se ejecute el comando de pararlo va a seguir levantado. Con **docker exec** solamente se está levantando ese contenedor durante su uso, si salimos del contenedor se para en el instante.