

# UNIDAD TRABAJO 5: VIRTUALIZACIÓN. CONTENEDORES. DOCKER.

Jose Francisco Murcia Fuentes

Direcciones del repositorio

<https://github.com/murcieta-gva/Tareas.git>

<https://github.com/murcieta-gva/Tarea-JFMF.git>

## 2.Caso práctico: Instalando Docker

[Instalación de Docker usando la versión de escritorio en mac nuestro información desde el terminal]

```
josefranciscozurciafuentes — -zsh — 68x26
~ — -zsh
Last login: Wed Dec  4 20:07:46 on ttys000
josefranciscozurciafuentes@MacBook-Pro-de-Jose ~ % docker --version
Docker version 27.3.1, build ce12230
josefranciscozurciafuentes@MacBook-Pro-de-Jose ~ %
```

[Instalación de Docker usando la versión de escritorio en mac nuestro información desde el terminal de la aplicación]

```
Terminal
josefranciscozurciafuentes@MacBook-Pro-de-Jose ~ % sudo docker run hello-world
Password:
Unable to find image 'hello-world:latest' locally
2024/12/09 20:38:54 must use ASL logging (which requires CGO) if running as root
latest: Pulling from library/hello-world
478afc919002: Download complete
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fcb966
Status: Downloaded newer image for hello-world:latest

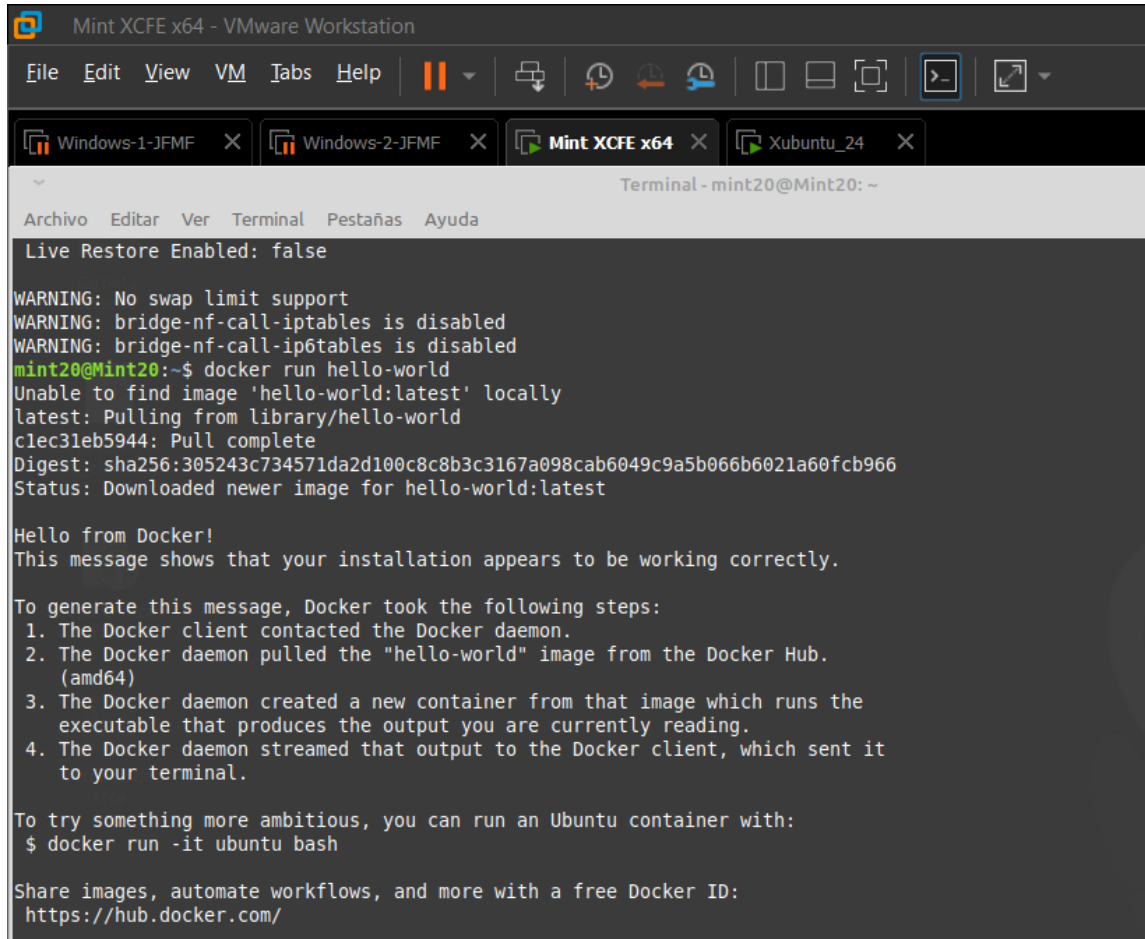
Hello from Docker!
This message shows that your installation appears to be working correctly.
```

## UNIDAD TRABAJO 5: VIRTUALIZACIÓN.CONTENEDORES.

DOCKER.5-5

### 2.1. Instalación de Docker en Ubuntu / Linux Mint3.

*[Instalación en Ubuntu-Linux en MV con Linux Mint utilizando el terminal, aunque yo para esta práctica voy a utilizar MAC que trabajan Línea de comandos CLI es igual.]*



```
Mint XCFE x64 - VMware Workstation
File Edit View VM Tabs Help
Windows-1-JFMF Windows-2-JFMF Mint XCFE x64 Xubuntu_24
Terminal - mint20@Mint20: ~
Archivo Editar Ver Terminal Pestañas Ayuda
Live Restore Enabled: false

WARNING: No swap limit support
WARNING: bridge-nf-call-iptables is disabled
WARNING: bridge-nf-call-ip6tables is disabled
mint20@Mint20:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:305243c734571da2d100c8c8b3c3167a098cab6049c9a5b066b6021a60fcb966
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
```

### 3.Comandos más habituales para la gestión de contenedores.

[Captura de pantallas con la salida del comando **docker --help**]

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose ~ % docker --help
Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
run      Create and run a new container from an image
exec     Execute a command in a running container
ps       List containers
build    Build an image from a Dockerfile
pull     Download an image from a registry
push     Upload an image to a registry
images   List images
login    Authenticate to a registry
logout   Log out from a registry
search   Search Docker Hub for images
version  Show the Docker version information
info     Display system-wide information

Management Commands:
ai*      Ask Gordon - Docker Agent
builder  Manage builds
buildx*  Docker Buildx
compose* Docker Compose
container Manage containers
context  Manage contexts
debug*   Get a shell into any image or container
desktop* Docker Desktop commands (Alpha)
dev*     Docker Dev Environments
extension* Manages Docker extensions
feedback* Provide feedback, right in your terminal!
image    Manage images
init*    Creates Docker-related starter files for your project
manifest Manage Docker image manifests and manifest lists
network  Manage networks
plugin   Manage plugins
sbom*    View the packaged-based Software Bill Of Materials (SBOM) for an image
scout*   Docker Scout
system   Manage Docker
trust    Manage trust on Docker images
volume   Manage volumes
```

## UNIDAD TRABAJO 5: VIRTUALIZACIÓN.CONTENEDORES.

## DOCKER.5-5

```
Swarm Commands:
  swarm      Manage Swarm

Commands:
  attach      Attach local standard input, output, and error streams to a running container
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  events      Get real time events from the server
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  import      Import the contents from a tarball to create a filesystem image
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  save        Save one or more images to a tar archive (streamed to STDOUT by default)
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  wait        Block until one or more containers stop, then print their exit codes
```

```
Global Options:
  --config string      Location of client config files
                        (default
                        "/Users/josefranciscomurciafuentes/.docker")
  -c, --context string  Name of the context to use to
                        connect to the daemon (overrides
                        DOCKER_HOST env var and default
                        context set with "docker context use")
  -D, --debug           Enable debug mode
  -H, --host list       Daemon socket to connect to
  -l, --log-level string Set the logging level ("debug",
                        "info", "warn", "error", "fatal")
                        (default "info")
  --tls                Use TLS; implied by --tlsverify
  --tlscacert string    Trust certs signed only by this CA
                        (default
                        "/Users/josefranciscomurciafuentes/.docker/ca.pem")
  --tlscert string      Path to TLS certificate file
                        (default
                        "/Users/josefranciscomurciafuentes/.docker/cert.pem")
  --tlskey string       Path to TLS key file (default
                        "/Users/josefranciscomurciafuentes/.docker/key.pem")
  --tlsverify           Use TLS and verify the remote
  -v, --version         Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/
josefranciscomurciafuentes@MacBook-Pro-de-Jose ~ %
```

### 3.1. Ejercicios propuestos:

#### Explicación resumida qué es Docker.

[Es un software de código abierto para desplegar aplicaciones dentro de contenedores virtuales con independencia del sistema operativo o plataforma además es compatible con la mayoría de los lenguajes y aplicaciones, sus principales características son que es ligero, portable, inmutable, escalable y seguro.]



## ¿Cuál es la diferencia entre Docker y una máquina virtual?

*[La principal diferencia es que Docker utiliza contenedores para ejecutar aplicaciones varios contenedores comparten el sistema operativo lo que los hace más ligeros que las máquinas virtuales que utilizan un hipervisor cada maquina tiene su propio sistema operativo lo que no es tan eficiente.]*

**Averigüe cual es la versión de Docker que tiene instalada en su sistema operativo.**

```
josefranciscomurciafuentes — -zsh — 68x26
~ — -zsh
Last login: Wed Dec  4 20:07:46 on ttys000
josefranciscomurciafuentes@MacBook-Pro-de-Jose ~ % docker --version
Docker version 27.3.1, build ce12230
josefranciscomurciafuentes@MacBook-Pro-de-Jose ~ %
```

].

## ¿Cuál es la diferencia entre *Docker Server Engine* y *Docker Client Engine*?

*[Docker Server Engine es el componente que ejecuta y gestiona los contenedores, es el encargado de construir, ejecutar y distribuir los contenedores, además es con el que se administran los recursos del sistema, como la memoria y la CPU mientras que en el Docker Client es la interface de línea de comandos donde se administran los contenedores las imágenes y los volúmenes.]*

**Ejecute los siguientes comandos:**

**Cuestiones: - ¿Cual es el resultado de la ejecución del contenedor *hello-world*?**

*[Captura de pantalla salida terminal comando **run hello-world**]*

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose ~ % docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (arm64v8)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

josefranciscomurciafuentes@MacBook-Pro-de-Jose ~ %
```

## ¿Cuál es el tamaño de la imagen descargada?

[El tamaño de la imagen descargada es de 139mb]

```
[
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app % docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          latest      80dd3c3b9c6c  3 weeks ago   139MB
hello-world     latest     305243c73457  19 months ago 21kB
]
```

### 4.1. Instalación de Entorno de desarrollo Python en Ubuntu:

[Captura de ejecución del comando desde directorio padre **my-python-app** y con el nombre personalizado del contenedor **jfmf-ubuntu-python** y acceso a la máquina.]

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app % ls
ejemplo.py  tests
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app % sudo docker run -it --name jfmf-ubuntu-python -v ${PWD}/my
-python-app:/app -w /app ubuntu
Password:
root@ffa236b52971:/app#
```

RAM 0.61 GB CPU 0.00% Disk 58.24 GB avail. of 62.67 GB

#### 4.1.1. Dockerfile

Podemos obtener una imagen automatizada, similar a la obtenida mediante los comandos anteriores, empleando *Dockerfile* y los siguientes comandos:

10. En primer lugar nos situamos en el directorio *my-python-app* y mediante un editor de textos construimos el **Dockerfile**:

[Captura de dockfile comentamos la línea # Run pip install -r requirements.txt]

## UNIDAD TRABAJO 5: VIRTUALIZACIÓN.CONTENEDORES.

## DOCKER.5-5

```
my-python-app — pico — sudo — 68x26
~/my-python-app — pico — sudo
UW PICO 5.09 File: Dockerfile.txt Modified
FROM ubuntu:latest
RUN apt update && apt install -y \
python3 \
pip \
python3-pytest
WORKDIR /app
COPY . /app/
RUN pip install -r requirements.txt
ENTRYPOINT [ "pytest" ]
```

[Captura del tamaño de las imágenes **my-ubuntu-python**]

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app % docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
my-ubuntu-python    latest     bc161fcc44e1  3 minutes ago  806MB
ubuntu-python       v1.0      ae563b060e37  9 hours ago   823MB
ubuntu              latest     80dd3c3b9c6c  3 weeks ago   139MB
bash                latest     4bbfbe07ecce  2 months ago  25.6MB
hello-world         latest     305243c73457  19 months ago 21kB
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app %
```

## 4.2. Entorno de desarrollo Python en Alpine:

[Captura de la construcción del contenedor **my-alpine-python**]

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app % docker build -t my-alpine-python .
[+] Building 3.1s (11/11) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 622B                                0.0s
=> [internal] load metadata for docker.io/library/alpine:latest  0.0s
=> [internal] load .dockerignore                                   0.0s
=> => transferring context: 2B                                       0.0s
=> [1/4] FROM docker.io/library/alpine:latest@sha256:21dc6063fd678b478f57c0e13f47560d0ea4eeba26dfc947b2a4f81f686 0.9s
=> => resolve docker.io/library/alpine:latest@sha256:21dc6063fd678b478f57c0e13f47560d0ea4eeba26dfc947b2a4f81f686 0.9s
=> [internal] load build context                                   0.0s
=> => transferring context: 3.41kB                                    0.0s
=> [auth] library/alpine:pull token for registry-1.docker.io     0.0s
=> [auth] library/alpine:pull token for registry-1.docker.io     0.0s
=> CACHED [2/4] RUN apk update && apk add python3 && apk add py3-pip 0.0s
=> CACHED [3/4] WORKDIR /app                                       0.0s
=> [4/4] COPY . /app/                                              0.0s
=> exporting to image                                              2.1s
=> => exporting layers                                              1.7s
=> => exporting manifest sha256:831347c489883a0db238106bc3fa43df51ece9cbac5df5040e3b0277720f323c 0.0s
=> => exporting config sha256:6793e27d9a6fc79c902db615ff13bab126a7665d0f0efb4396c2552706d95cbc 0.0s
=> => exporting attestation manifest sha256:31a0991e144ddd9a0becccd5e3b393a055ab2a575ad289a3fe6837eb1f147e0cb 0.0s
=> => exporting manifest list sha256:aaa748ce8acb4384e59913b240bc0dc3c2764b000544a2bd0cba00f7421c3d35 0.0s
=> => naming to docker.io/library/my-alpine-python:latest         0.0s
=> => unpacking to docker.io/library/my-alpine-python:latest      0.3s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/pmavmd13jzno6ome8dw594mbu
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app %
```



[Captura del tamaño de las imágenes contenedor **alpine-python**]

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app % docker commit -m "Entorno de desarrollo Python sobre Alpine
Jose Francisco" -a "José Francisco Murcia
Fuentes <josmurfue@alu.edu.gva.es>" jfmf-alpine-python alpine-python:v1.0
sha256:782fb5d43a1aa47bb33422a20fba1f5097b3be2aad19f6646d8dcddc06eeefc1
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app % docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
alpine-python	v1.0	782fb5d43a1a	11 seconds ago	115MB
my-ubuntu-python	latest	bc161fcc44e1	About an hour ago	806MB
ubuntu-python	v1.0	ae563b060e37	10 hours ago	823MB
alpine	latest	21dc6063fd67	6 days ago	12.8MB
ubuntu	latest	80dd3c3b9c6c	3 weeks ago	139MB
bash	latest	4bbfbc07ecee	2 months ago	25.6MB
hello-world	latest	305243c73457	19 months ago	21kB

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app %
```

#### 4.2.1. Dockerfile

[Contenedor en ejecución]

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app % docker ps -l
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
cdc683854b62	my-alpine-python	"pytest"	24 minutes ago	Created		goofy_shirley

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app %
```

[Captura del tamaño de las imágenes]

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app % docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
my-alpine-python	latest	eea748ce8acb	3 minutes ago	110MB
alpine-python	v1.0	782fb5d43a1a	21 minutes ago	115MB
my-ubuntu-python	latest	bc161fcc44e1	2 hours ago	806MB
ubuntu-python	v1.0	ae563b060e37	11 hours ago	823MB
alpine	latest	21dc6063fd67	6 days ago	12.8MB
ubuntu	latest	80dd3c3b9c6c	3 weeks ago	139MB
bash	latest	4bbfbc07ecee	2 months ago	25.6MB
hello-world	latest	305243c73457	19 months ago	21kB

```
josefranciscomurciafuentes@MacBook-Pro-de-Jose my-python-app %
```