

## Instalación de Swift en diferentes sistemas operativos

---

### Installing Swift on different operating systems

Chávez García Monsserrath

**Resúmen.** *El lenguaje de programación Swift es utilizado actualmente para desarrollar aplicaciones digitales para entornos iOS y macOS, presentes en apps de iOS, Mac, Apple TV y Apple Watch<sup>1</sup>. El lenguaje Swift tiene como principal objetivo brindar a los desarrolladores mayor libertad al ser de código abierto. Desarrollado en 2014 por Apple, este lenguaje destaca frente a otros por ser más seguro, más rápido en su desarrollo y con una sintaxis más clara y concisa, además de que sigue evolucionando continuamente, aprovechando así mejor los avances de las nuevas tecnologías, permitiendo desarrollos cada vez más complejos y funcionales para los usuarios.*

*El presente trabajo tiene como objetivo presentar algunas de las formas en las que se puede hacer una instalación del lenguaje Swift en distintos sistemas operativos tales como Linux, Windows y el propio macOS, cada uno con sus propias características, como el ser de código abierto o propietario, costos, interfaces de usuario, seguridad, desempeño, etc. Pero que en esta ocasión compartirán la característica de poder instalar Swift en ellos. Además de un elemento adicional en el que se explicará una forma de instalar Docker en un sistema operativo Linux. Esto debido a que en el contexto de Swift, Docker puede ser útil para empaquetar y distribuir aplicaciones desarrolladas en el mismo en diferentes sistemas operativos.*

**Palabras clave:** Swift, Sistemas Operativos, iOS, macOS, Apple, Windows, Linux, instalación.

**Abstract.** *The Swift programming language is currently used to develop digital applications for iOS and macOS environments, present in iOS, Mac, Apple TV and Apple Watch apps. The main objective of the Swift language is to provide developers with greater freedom by being open source. Developed in 2014 by Apple, this language stands out from others for being safer, faster in its development and with a clearer and more concise syntax, in addition to the fact that it continues to evolve continuously, thus taking better advantage of*

---

<sup>1</sup> Swift - Apple (MX). (2014). Apple. <https://www.apple.com/mx/swift>

*the advances of new technologies, allowing increasingly complex and functional developments for users.*

*This paper aims to present some of the ways in which you can make an installation of the Swift language in different operating systems such as Linux, Windows and macOS itself, each with its own characteristics, such as being open source or proprietary, costs, user interfaces, security, performance, etc.. But on this occasion they will share the feature of being able to install Swift on them. Besides an additional element in which we will explain a way to install Docker on a Linux operating system. This is because in the context of Swift, Docker can be useful to package and distribute applications developed in it in different operating systems.*

**Keywords:** Swift, Operating Systems, iOS, macOS, Apple, Windows, Linux, installation.

---

## Introducción.

Para las diferentes instalaciones se requerirán diferentes cosas, y será más rápido o sencillo según con lo que se cuente. Por ejemplo, si se desea hacer la instalación en un sistema operativo Linux pero actualmente se está trabajando en un entorno Windows, se podría optar por trabajar en una máquina virtual. Esto contando con algún software de virtualización de sistemas operativos que permita a los usuarios ejecutar múltiples sistemas operativos en un único host, como por ejemplo Oracle VirtualBox<sup>2</sup>, y posterior a ello

cargar en él el sistema operativo Linux<sup>3</sup>.

## Instalación de Swift en Linux.

Linux es un sistema operativo de código abierto y gratuito, basado en Unix, altamente personalizable y escalable, y para poder hacer la instalación del lenguaje Swift se seguirán los siguientes pasos.

- Abrir una terminal y escribir el código que nos va a permitir descargar dos paquetes fundamentales, clang y libicu-dev.

```
sudo apt-get install clang  
libicu-dev
```

---

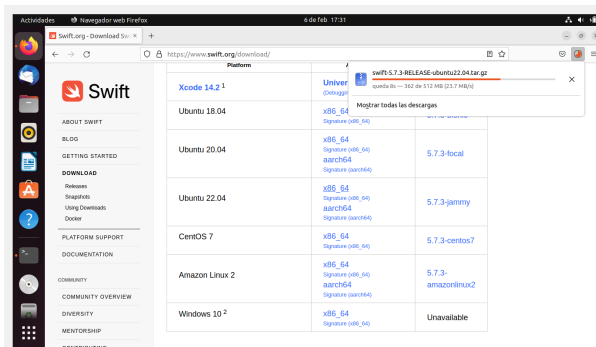
<sup>2</sup> Downloads – Oracle VM VirtualBox. (2023). Virtualbox.org.  
<https://www.virtualbox.org/wiki/Downloads>

---

<sup>3</sup> Download Ubuntu Desktop | Download | Ubuntu. (2023). Ubuntu.  
<https://ubuntu.com/download/desktop>

```
monsnec@monsnec-VirtualBox: ~  
monsnec@monsnec-VirtualBox: $ sudo apt-get install clang libicu-dev  
[sudo] contraseña para monsnec:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
libicu-dev ya está en su versión más reciente (70.1-2).  
clang ya está en su versión más reciente (1:14.0-55-exp2).  
Los paquetes indicados a continuación se instalarán de forma automática y ya no  
son necesarios.  
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver  
intel-media-va-driver libaacs5 libbass1 libbass1-dev libbasscodes5 libbassformat5  
libbassmidi1 libbassmidi2 libbass3 libbass3-dev libbass3-dev libbass3-dev libbass3-dev  
libbcode2-1.0 libbda15 libbfile1 libbgnm libbgnm1  
libbgsfplugin-plugins-bad1.0-0 libbigo2mm1 libbily1.0 libbilm15 libbfx1  
libbmysoai1 libbom1 libbopenmp2 libbpgm-5.3-0 libbpostproc5 libbrabntnq4  
libbrubberband libbseed-0.0 libbshn3 libbshnapp1v5 libbsord-0.0 libbsration-0.0  
libbsrt1.4-gnutils libbsh-gcrypt-4 libbwsresample3 libbwscale5 libbutfread0  
libva-drm2 libva-wayland2 libva-x11-2 libva2 libbvpai1 libbvdts1.1  
libbx265-199 libbxdcode4 libbzig2 libbzn3 libbzdai-common libbzio  
mesa-va-drivers mesa-vaapi-drivers pocketsphinx-en-us systemd-hwe-hwdb  
va-driver-all vdpau-driver-all  
Utilice 'sudo apt autoremove' para eliminarlos.  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 272 no actualizados.  
monsnec@monsnec-VirtualBox: $
```

- Posterior a ello, tenemos que dirigirnos a un navegador a la página oficial de Swift <sup>4</sup> en la que podremos encontrar las versiones a descargar según nuestro sistema operativo, que en este caso es Linux. Y proceder a hacer la descarga pertinente.



- Una vez descargado dirigiéndonos a la carpeta en la que se guardó, la extraemos y por mayor comodidad podemos renombrarlo únicamente con el nombre y su versión.
- Luego, regresando a la línea de comando nos ubicamos en el directorio en el que se guardaron los paquetes.

```
cd .swift-5.7.3/usr/bin/
```

<sup>4</sup> Apple Inc. (2019, February 28). Swift.org. Swift.org. <https://www.swift.org/download/>

```

nonssec@nonssec-VirtualBox: $ cd ~/.swift-5.7.3/usr/bin/
nonssec@nonssec-VirtualBox: ~/.swift-5.7.3/usr/bin$ ls
clang          llvm-cov       swiftc
clang++        llvm-profdata  swift-demangle
clang-13       plutil         swift-driver
clang-cl       repl_swift     swift-frontend
clang-cpp      sdk-module-lists swift-help
clangd         sourcekit-lsp  swift-package
lcc            swift          swift-package-collection
ld64.lld       swift-api-checker.py swift-package-registry
ld.lld         swift-api-digester swift-run
lld            swift-api-extract swift-symbolgraph-extract
lldb           swift-autolink-extract swift-test
lldb-argdumper swift-build     wasn-ld
lldb-server    swift-build-sdk-interfaces
lld-link       swift-build-tool
nonssec@nonssec-VirtualBox: ~/.swift-5.7.3/usr/bin$

```

- En consecuencia a todos los pasos anteriores con el comando `ls` podemos verificar que se encuentren todas las cosas en dicho directorio, para después mediante el comando `pwd` ubicarnos en la ruta en la que estamos y copiarla, pues la ocuparemos más adelante.

```

monsssec@monsssec-VirtualBox: ~/swift-5.7.3/usr/bin
libx265-199 libxvidcore4 libzimg2 libzng5 libzvb1-common libzvb1b0
mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us systemd-hwdb
va-driver-all vdpau-driver-all

Utilice "sudo apt autoremove" para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 272 no actualizados.
monsssec@monsssec-VirtualBox: $ cd ~/swift-5.7.3/usr/bin/
monsssec@monsssec-VirtualBox: ~/swift-5.7.3/usr/bin$ ls
clang          llvm-cov          swiftc
clang++        llvm-profdata     swift-demangle
clang-13       plutl            swift-driver
clang-cl       repl_swift        swift-frontend
clang-cpp      sdk-module-lists
clangd         sourcekit-lsp
docc           swift
ld64.lld       swift-api-checker.py
ld.lld         swift-api-digester
lld            swift-api-extract
lldb           swift-autolink-extract
lldb-argdumper swift-build
lldb-server    swift-build-sdk-interface
lld-link       swift-build-tool

monsssec@monsssec-VirtualBox: ~/swift-5.7.3/usr/bin$
/home/monsssec/.swift-5.7.3/usr/bin
monsssec@monsssec-VirtualBox: ~/swift-5.7.3/usr/bin$

```

- El siguiente paso será abrir el archivo de configuración del bash mediante el comando `nano .bashrc` en otra terminal.

[illegible]

- Ahora lo que se hará es agregar lo copiado anteriormente al path del sistema mediante el comando `export PATH="(lo`

copiado  
anteriormente): "\${PATH}"

```
GNU nano 6.2 .bashrc
# If not running under csh, update the PATH to include the location of the
# bash completion scripts.
if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
if [ -f /usr/share/bash-completion/bash_completion ]; then
. /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
. /etc/bash_completion
fi
fi

export PATH="/home/nonssec/.swift-5.7.3/usr/bin:${PATH}"
```

- Casi por concluir realizamos un source para recargar lo realizado mediante el comando `source .bashrc`.
- Y listo, para verificar que esté instalada correctamente se teclea el comando `swift --version`.
- Para concluir entramos con el comando `swift` y como método de prueba utilizamos un “hola mundo”.

```
monsssec@monsssec-VirtualBox: ~
monsssec@monsssec-VirtualBox:~$ ls -a
. .bashrc .cache .config .cshrc .local .profile .swift-5.7.3
.. .bash_history .bash_logout .cache .config .cshrc .local .profile .swift-5.7.3
monsssec@monsssec-VirtualBox:~$ source .bashrc
monsssec@monsssec-VirtualBox:~$ swift --version
Swift version 5.7.3 (swift-5.7.3-RELEASE)
Target: x86_64-unknown-linux-gnu
monsssec@monsssec-VirtualBox:~$ swift repl
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ModuleNotFoundError: No module named 'lldb'
Welcome to Swift version 5.7.3 (swift-5.7.3-RELEASE).
Type :help for assistance.
1> print ("Hola mundo")
Hola mundo
2>
```

## Instalación de Swift en Windows.

Windows es un sistema operativo desarrollado por Microsoft, el cual ofrece una interfaz de usuario

gráfica y aplicaciones de escritorio como sus puntos principales.

Para la instalación de swift en este sistema operativo se seguirán los siguientes pasos:

- Ingresar a la página oficial de Swift<sup>5</sup> en la que nos encontraremos los diversos sistemas operativos en la que los podremos instalar, en este caso el correspondiente a Windows 10.

<div> <div>ABOUT SWIFT</div> <div>BLOG</div> <div>GETTING STARTED</div> <div>DOWNLOAD</div> <div>Releases</div> <div>Snapshots</div> <div>Using Downloads</div> <div>Docler</div> <div>PLATFORM SUPPORT</div> </div>	Ubuntu 22.04	Signature (x86_64) aarch64 Signature (aarch64)	5.7.3-jammy
	CentOS 7	x86_64 Signature (x86_64)	5.7.3-centos7
	Amazon Linux 2	x86_64 Signature (x86_64) aarch64 Signature (aarch64)	5.7.3-amazonlinux2
	Windows 10 <sup>2</sup>	x86_64 Signature (x86_64)	Unavailable

- Una vez descargado seguir la típica metodología de continuar con cada ventana.
- Y para concluir y checar que fue descargado de manera correcta abriendo la línea de comando y tecleando el comando `swift --version` para comprobar que fue instalada de manera exitosa.

```
C:\Users\monss>swift --version
compnrd.org Swift version 5.7.3 (swift-5.7.3-RELEASE)
Target: x86_64-unknown-windows-msvc

C:\Users\monss>swift

Welcome to Swift!

Subcommands:

swift build      Build Swift packages
swift package    Create and work on packages
swift run        Run a program from a package
swift test       Run package tests
swift repl       Experiment with Swift code interactively

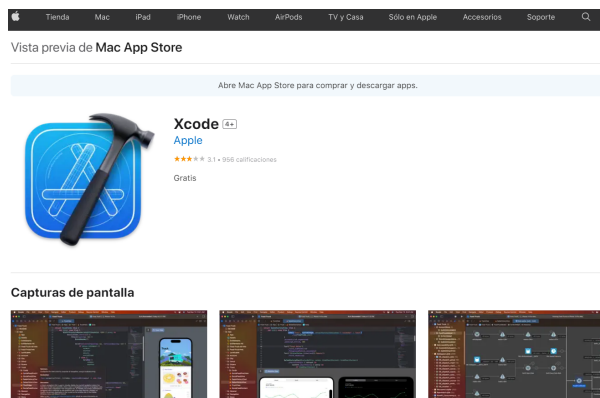
Use `swift --help` for descriptions of available options and
Use `swift help <subcommand>` for more information about a s
```

<sup>5</sup> Apple Inc. (2019, February 28). Swift.org. Swift.org. <https://www.swift.org/download/>

## Instalación de Swift en macOS.

MacOS es el sistema operativo de Apple para sus computadoras de escritorio y portátiles la cual nos ofrece una interfaz intuitiva, integración con productos Apple y una amplia gama de aplicaciones diseñadas específicamente para el sistema.

- Para su instalación únicamente nos dirigiremos a la tienda oficial de aplicaciones propia de Apple la “App Store”<sup>6</sup>, y buscaremos la aplicación llamada “Xcode”, posterior a eso daremos click en instalar.



## Instalación de Docker en Linux.

Docker es una plataforma de contenedores que permite empaquetar y distribuir aplicaciones de manera eficiente en cualquier entorno, estos contenedores

facilitan la implementación y el despliegue de aplicaciones. Para su instalación en Linux nos podemos ayudar de la guía presente en la página oficial<sup>7</sup> y seguir los siguientes pasos.

- Se comienza por abrir una terminal y escribir el comando `sudo apt update` para tener todos los repositorios actualizados correctamente.

```
monssec@monssec-VirtualBox: ~
monssec@monssec-VirtualBox:~$ sudo apt update
[sudo] contraseña para monssec:
Des:1 https://download.docker.com/linux/ubuntu jammy InRelease [48.9 kB]
Obj:2 http://mx.archive.ubuntu.com/ubuntu jammy InRelease
Obj:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
Err:1 https://download.docker.com/linux/ubuntu jammy InRelease
  Las firmas siguientes no se pudieron verificar porque su clave pública no
  disponible: NO_PUBKEY 7EA0A9C3F273FCD8
Obj:4 http://mx.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:5 http://mx.archive.ubuntu.com/ubuntu jammy-backports InRelease
Leyendo lista de paquetes... Hecho
W: Error de GPG: https://download.docker.com/linux/ubuntu jammy InRelease: L
  irmas siguientes no se pudieron verificar porque su clave pública no está d
  ible: NO_PUBKEY 7EA0A9C3F273FCD8
E: El repositorio «https://download.docker.com/linux/ubuntu jammy InRelease»
```

- Posterior a ello, introducimos los comandos necesarios para poder tener las claves para el repositorio de Docker, las cuales son.
  - `sudo apt install apt-transport-https ca-certificates curl software-properties-common.`
  - `curl -fsSL http://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`

<sup>6</sup> Apple. (2012, February 16). Xcode. Mac App Store. <https://apps.apple.com/es/app/xcode/>

<sup>7</sup> Install on Linux. (2023, February 7). Docker Documentation. <https://docs.docker.com/desktop/install/linux-install/>



```

monssec@monssec-VirtualBox: ~$ sudo apt install apt-transport-https
es curl software-properties-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática
son necesarios.
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver
intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libav
libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint
libcodec2-1.0 libdavid5 libflite1 libgme0 libgsm1
libgstreamer-plugins-bad1.0-0 libigdgmm12 libllv-0-0 libllvm15 l
libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabi
librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 lib
libstr1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudf
libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1
libx265-199 libxvidcore4 libzimg2 libzmq5 libzvb1-common libzvb10
mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us systemd-hwe
va-driver-all vdpau-driver-all
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
libxv14 ubuntu2 software-properties software-properties-gtk

```

```

monssec@monssec-VirtualBox: ~$ curl -fsSL http://download.docker.com/linux/ub
done.
monssec@monssec-VirtualBox: ~$ curl -fsSL http://download.docker.com/linux/ub
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instea
ee apt-key(8)).
OK
monssec@monssec-VirtualBox: ~$ sudo add-apt-repository "deb [arch=amd64] htt
download.docker.com/linux/ubuntu focal stable"
Repository: «deb [arch=amd64] https://download.docker.com/linux/ubuntu foc
able»
Descripción:
Archive for codename: focal components: stable
Más información: https://download.docker.com/linux/ubuntu
Añadiendo repositorio.
Oprima [INTRO] para continuar o Ctrl+c para cancelar.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_dock
m_linux_ubuntu-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_
oad_docker_com_linux_ubuntu-jammy.list
Des:1 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]

```

- Ya que se ejecutaron los comandos anteriores, el siguiente paso es añadir el repositorio de Docker y de este modo poderlo instalar correctamente., mediante el comando `sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"`.

```

OK
monssec@monssec-VirtualBox: ~$ sudo add-apt-repository "deb [arch=amd64]
download.docker.com/linux/ubuntu focal stable"
Repository: «deb [arch=amd64] https://download.docker.com/linux/ubuntu
Descripción:
Archive for codename: focal components: stable
Más información: https://download.docker.com/linux/ubuntu
Añadiendo repositorio.
Oprima [INTRO] para continuar o Ctrl+c para cancelar.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_
m_linux_ubuntu-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-ht
oad_docker_com_linux_ubuntu-jammy.list
Des:1 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Des:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 k

```

- Ya que hicimos eso de nuevo actualizamos los repositorios con el comando `sudo apt update`.

```

monssec@monssec-VirtualBox: ~$ sudo apt update
Des:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Obj:2 https://download.docker.com/linux/ubuntu focal InRelease
Des:3 https://download.docker.com/linux/ubuntu jammy InRelease [48.9 kB]
Obj:4 http://mx.archive.ubuntu.com/ubuntu jammy InRelease
Obj:5 http://mx.archive.ubuntu.com/ubuntu jammy-updates InRelease
Des:6 http://mx.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Err:3 https://download.docker.com/linux/ubuntu jammy InRelease
Las firmas siguientes no se pudieron verificar porque su clave pública no está
disponible: NO_PUBKEY 7EA0A9C3F273FCD8
Leyendo lista de paquetes... Hecho

```

- Abrimos una nueva terminal para ahora sí comenzar la instalación de Docker, la cual la haremos con el comando `sudo apt install docker-ce`.

```

monssec@monssec-VirtualBox: ~$ sudo apt install docker-ce
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver
intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58
libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1
libcodec2-1.0 libdavid5 libflite1 libgme0 libgsm1
libgstreamer-plugins-bad1.0-0 libigdgmm12 libllv-0-0 libllvm15 libmfx1
libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4
librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libstrat0-0
libstr1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0
libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1
libx265-199 libxvidcore4 libzimg2 libzmq5 libzvb1-common libzvb10
mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us systemd-hwe-hwdb
va-driver-all vdpau-driver-all
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
docker-compose-plugin docker-scan-plugin git git-man liberror-perl libslirp0
pigz slirp4netns
paquetes sugeridos:

```

- Como último paso solo se comprobará que Docker esté funcionando correctamente mediante el comando `sudo systemctl status docker`. Si la parte de active aparece en verde y con el título (running) significa que está corriendo correctamente.

```

monssec@monssec-VirtualBox: ~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset
   Active: active (running) since Mon 2023-02-06 19:40:03 CST; 50s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 20353 (dockerd)
       Tasks: 7
      Memory: 35.4M
         CPU: 342ms
    CGroup: /system.slice/docker.service
            └─ /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont

feb 06 19:40:02 monssec-VirtualBox dockerd[20353]: time="2023-02-06T19:40:02.16
feb 06 19:40:02 monssec-VirtualBox dockerd[20353]: time="2023-02-06T19:40:02.17
feb 06 19:40:02 monssec-VirtualBox dockerd[20353]: time="2023-02-06T19:40:02.26
feb 06 19:40:02 monssec-VirtualBox dockerd[20353]: time="2023-02-06T19:40:02.48
feb 06 19:40:02 monssec-VirtualBox dockerd[20353]: time="2023-02-06T19:40:02.99
feb 06 19:40:03 monssec-VirtualBox dockerd[20353]: time="2023-02-06T19:40:03.20
feb 06 19:40:03 monssec-VirtualBox dockerd[20353]: time="2023-02-06T19:40:03.20
feb 06 19:40:03 monssec-VirtualBox dockerd[20353]: time="2023-02-06T19:40:03.26
feb 06 19:40:03 monssec-VirtualBox systemd[1]: Started Docker Application Contai
feb 06 19:40:03 monssec-VirtualBox dockerd[20353]: time="2023-02-06T19:40:03.30
[lines 1-22/22 (END)]

```

## Conclusión

En conclusión obtenemos un lenguaje de programación que nos ayudará no solo a completar las tareas requeridas de manera más eficiente, sino con una forma para aprenderla relativamente más sencilla, ayudándonos a no solo limitarnos para poder practicar con esta herramienta a su sistema operativo objetivo, sino a otros que podamos tener y utilizar para ejecutarla de igual manera.

Teniendo a su vez la facilidad de poder utilizarla si es que se nos pone delante alguno de los sistemas operativos vistos, sin estancarnos en uno y con mayor variedad de conocimiento.

<https://www.youtube.com/watch?v=2OplatS1AHo>

5. Swift - Apple (MX). (2014). Apple.  
<https://www.apple.com/mx/swift/>
6. ¿Qué es un sistema operativo? | Desarrollar Inclusión. (2017). Desarrollar Inclusión | Portal de Tecnología Inclusiva de CILSA.
7. Introducción – The Swift Programming Language en Español. (2014). Vercel.app.  
<https://swift-book-es.vercel>

## Bibliografía.

1. Swift - Apple Developer. (2023). Apple.com.  
<https://developer.apple.com/swift/>
2. Carlos, J. (2021, July 14). ¿Qué es Swift? El gran lenguaje de programación | Tokio School.
3. Swift. (2022, August 10). Lenguajes de Programación.  
<https://lenguajesdeprogramacion.net/swift/>
4. DevOps. (2022). CURSO DE DOCKER - Cómo Instalar DOCKER en LINUX [YouTube Video]. In YouTube.