

3. Imagen Docker de Raspberry Pi 3 con Qemu

Se genero la siguiente imagen docker de raspberry pi 3 que puede ser ejecutada desde linea de comandos

<https://hub.docker.com/r/soaunlam/emulador-raspberry-cmd>

Para ello se crearon 2 versiones de la imagen, en donde lo único que varían es el tamaño de la misma:

1. Una versión que ocupa 16 GB. Se la llamo V1
2. Una versión que ocupa 8 GB. Se la llamo V2

Pasos para ejecutar la imagen de Docker: ↻

Para ejecutarla se debe acceder de la siguiente forma

1. Instalar Docker
2. Una vez instalado Docker, se debe ejecuta el siguiente comando desde la consola, dependiendo de la version de la imagen que se desea trabajar

```
1 Docker run -it -p 5022:5022 soaunlam/emulador-raspberry-cmd:v1
```

ó una versión que ocupa menos espacio:

```
1 Docker run -it -p 5022:5022 soaunlam/emulador-raspberry-cmd:v2
```

3.Luego se empezará a descargar la imagen y finalmente empezara a ejecutarse la Raspberry desde la consola, tal como se muestra en la siguiente imagen.

```
[ OK ] Started System Logging Service.
[ OK ] Started triggerhappy global hotkey daemon.
[ OK ] Finished Remove Stale Onli...ext4 Metadata Check Snapshots.
[ FAILED ] Failed to start Check for Raspberry Pi EEPROM updates.
See 'systemctl status rpi-eeprom-update.service' for details.
0V? [ OK ] Finished Raise network interfaces.
[ OK ] Started WPA supplicant.
[ OK ] Started Authorization Manager.
[ OK ] Started Avahi mDNS/DNS-SD Stack.
[ OK ] Reached target Network.
[ OK ] Listening on Load/Save RF ...itch Status /dev/rfkill Watch.
Starting Modem Manager...
Starting /etc/rc.local Compatibility...
Starting OpenBSD Secure Shell server...
Starting Permit User Sessions...
[ OK ] Started LSB: rng-tools (Debian variant).
[ OK ] Finished dphys-swapfile - ...mount, and delete a swap file.
[ OK ] Started /etc/rc.local Compatibility.
[ OK ] Started LSB: Switch to ond...(unless shift key is pressed).
[ OK ] Started User Login Management.
[ OK ] Finished Permit User Sessions.
[ OK ] Started Getty on tty1.
[ OK ] Started Serial Getty on ttyAMA0.
[ OK ] Reached target Login Prompts.
usbnet: failed control transaction: request 0x2143 value 0xe index 0x0 length 0x0
usbnet: failed control transaction: request 0x2143 value 0xe index 0x0 length 0x0
usbnet: failed control transaction: request 0x2143 value 0xe index 0x0 length 0x0
[ OK ] Started OpenBSD Secure Shell server.
[ OK ] Started Modem Manager.
usbnet: failed control transaction: request 0x2143 value 0xe index 0x0 length 0x0
usbnet: failed control transaction: request 0x2143 value 0xe index 0x0 length 0x0
usbnet: failed control transaction: request 0x2143 value 0xe index 0x0 length 0x0

Debian GNU/Linux 11 raspberrypi ttyAMA0


raspberrypi login: usbnet: failed control transaction: request 0x2143 value 0xe index 0x0 length 0x0
raspberrypi login: |
```

4. Cuando aparezca el login y el password, ingresar los siguientes datos:
 - login: *pi*

- **Password:** *raspberry*

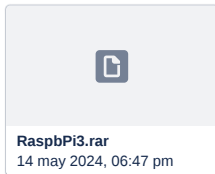
5. Una vez ingresado al sistema, tendra acceso al prompt del S.O  RaspbPi3.rar Raspbian

6. Si se quiere acceder desde windows por SSH al SO. Raspbian, se acceder a traves del puerto 5022 de localhost

 Una vez instalado, si se inicia el contenedor desde DockerContainer, se deberá attachearse a la consola de QEMU desde PowerShell usando el comando **docker ps**, obtener el **IdContainer** y usar el comando **docker attach IdContainer** para conectarse a la consola

Notas de la creación de la imagen:

Se tomo como base el archivo .rar investigado por dario



y el Dockefile generado por

 [GitHub - braghetto/rpivm: Docker container to emulate a Rpi 3B+ board using qemu native emulation.](#)

Los cambios que se debieron realizar en el Dockerfile documentados en el archivo word que se adjunta fueron los siguientes.

