

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Laboratorio de Fundamentos Embebidos

Práctica 1:

Introducción a los Sistemas Embebidos

Integrantes:

Cárdenas Cárdenas Jorge

Garrido Sánchez Samuel Arturo

Murrieta Villegas Alfonso

Reza Chavarría Sergio Gabriel

Valdespino Mendieta Joaquín

Práctica 1: Introducción a los Sistemas Embebidos

Objetivo:

Creación de una imagen de Raspbian para la tarjeta de pruebas Raspberry Pi, así como la verificación de comandos e instalación de software necesario para el manejo remoto de la tarjeta.

Introducción:

Son dispositivos multipropósito que hoy en día se caracterizan principalmente por ser de bajo consumo energético, por ser de bajo costo de producción y sobre todo por ser pequeños debido a la versatilidad que ofrecen para ser usados en distintos campos, desde aspectos hogareños hasta aspectos de carácter industrial.

Para la presente práctica el sistema embebido que emplearemos o conoceremos es la tarjeta de desarrollo Raspberry PI que se caracteriza principalmente por ser una SBC o Single Board Computer de bajo costo.

Desarrollo:

Realice cuidadosamente el siguiente procedimiento e incluya en su reporte, captura de pantalla de cada punto que se enlista a continuación.

1. Ingresar a raspberrypi.org

2. Descargar Raspberry Pi Imager for Windows

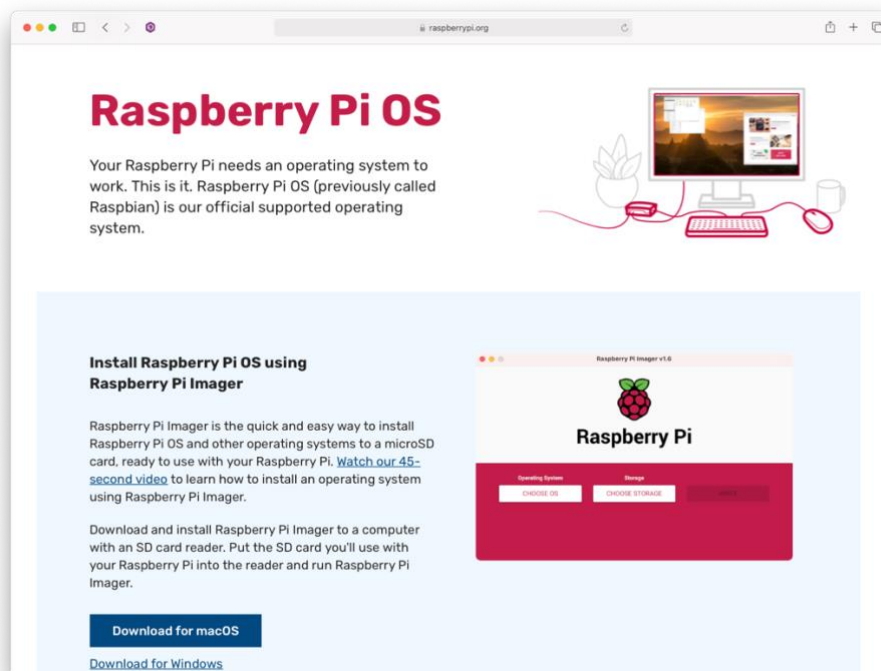


Figura 1: Página oficial de Raspberry Pi OS

3. Elegir la versión de Raspberry Pi OS y descargarla

Debido a que previamente ya se tenía un sistema operativo instalado, es por ello que se observa que el sistema operativo se llama “Raspbian” y no “Raspberry Pi Os” debido que es una versión anterior que además está optimizada para trabajar en la versión de nuestra Raspberry (Versión 3 B+)

4. Mediante Raspberry Pi Imager generar imagen en tarjeta SD

5. Una vez creada la imagen, insertar en su Raspberry y encenderla

6. Verificar el correcto funcionamiento

A continuación se muestra una captura de pantalla donde se observa instalado y funcionando Raspberry Pi Os en nuestra Raspberry Pi 3b+:

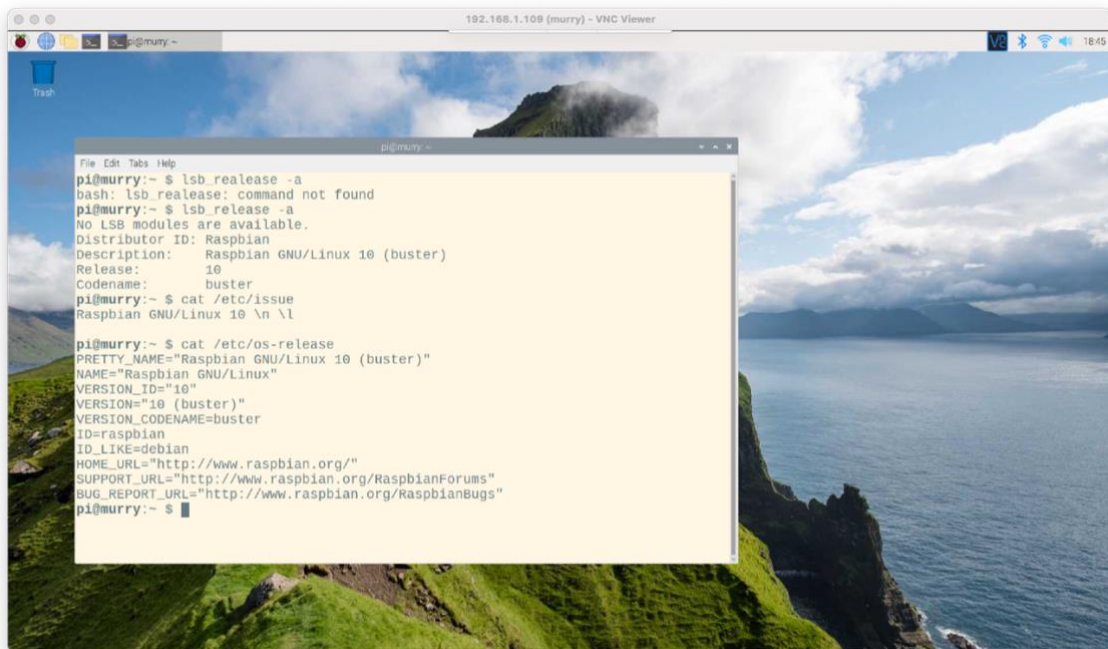
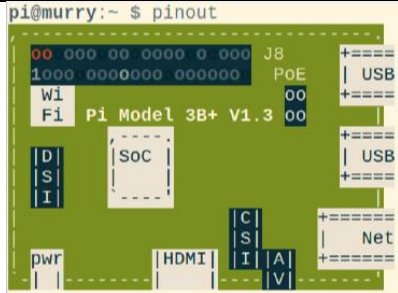
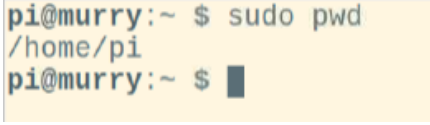
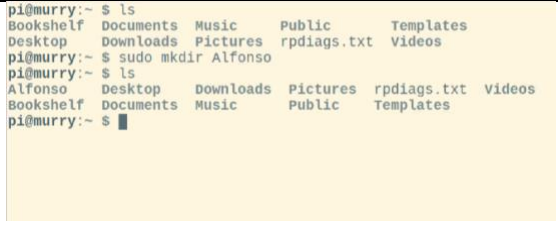
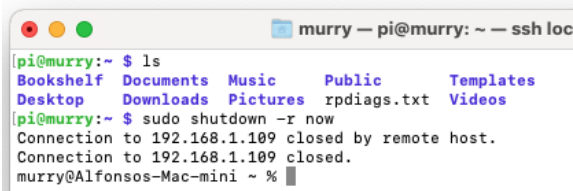
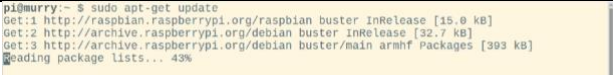
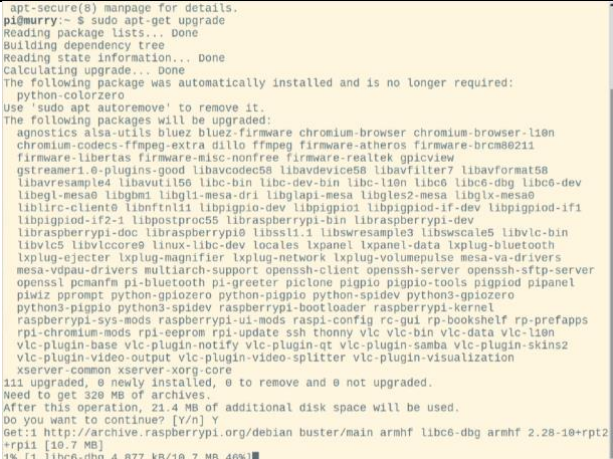


Figura 2: Versión del Sistema Operativo

7. Ingresar a símbolo de sistema, verificar el funcionamiento de algunos comandos básicos e indicar lo observado:

Comandos	Observaciones	Captura de Pantalla
pinout	Muestra un diagrama de como están las conexiones en nuestra tarjeta, además de mencionar cuál es el identificador de cada pin	 <pre> pi@murry:~ \$ pinout Revision : a020d3 SoC : BCM2837 RAM : 1024Mb Storage : MicroSD USB ports : 4 (excluding power) Ethernet ports: 1 Wi-fi : True Bluetooth : True Camera ports (CSI): 1 Display ports (DSI): 1 J8: 3V3 (1) (2) 5V GPIO2 (3) (4) 5V GPIO3 (5) (6) GND GPIO4 (7) (8) GPIO14 GND (9) (10) GPIO15 GPIO17 (11) (12) GPIO18 GPIO27 (13) (14) GND GPIO22 (15) (16) GPIO23 3V3 (17) (18) GPIO24 GPIO10 (19) (20) GND </pre>
sudo pwd	Nos regresa la dirección del sistema en la que nos encontramos	 <pre> pi@murry:~ \$ sudo pwd /home/pi pi@murry:~ \$ </pre>
sudo mkdir	Sirve para crear directorios, para ello posterior al comando debemos agregar el nombre del directorio	 <pre> pi@murry:~ \$ ls Bookshelf Documents Music Public Templates Desktop Downloads Pictures rpdiags.txt Videos pi@murry:~ \$ sudo mkdir Alfonso pi@murry:~ \$ ls Alfonso Desktop Downloads Pictures rpdiags.txt Videos Bookshelf Documents Music Public Templates pi@murry:~ \$ </pre>

<pre>sudo cat /proc/meminfo</pre>	<p>Nos muestra la información relevante de nuestra Raspberry, principalmente la asociada a nuestro hardware</p>	<pre>pi@murry:~\$ sudo cat /proc/meminfo MemTotal: 947036 kB MemFree: 523096 kB MemAvailable: 718680 kB Buffers: 28736 kB Cached: 216888 kB SwapCached: 0 kB Active: 241452 kB Inactive: 128076 kB Active(anon): 124208 kB Inactive(anon): 14748 kB Active(file): 117244 kB Inactive(file): 113328 kB Unevictable: 16 kB Mlocked: 16 kB SwapTotal: 102396 kB SwapFree: 102396 kB Dirty: 28 kB Writeback: 0 kB AnonPages: 123976 kB Mapped: 104044 kB Shmem: 15656 kB KReclaimable: 20136 kB Slab: 34712 kB SReclaimable: 20136 kB Sunreclaim: 14576 kB KernelStack: 1720 kB PageTables: 3800 kB NFS_Unstable: 0 kB Bounce: 0 kB WritebackTmp: 0 kB CommitLimit: 575912 kB Committed_AS: 981676 kB VmallocTotal: 1114112 kB VmallocUsed: 3532 kB VmallocChunk: 0 kB Percpu: 464 kB</pre>
<pre>sudo cat /proc/cpuinfo</pre>	<p>Nos muestra toda la información asociada con cada uno de los núcleos físicos de nuestro procesador o unidad central de procesador</p>	<pre>pi@murry:~\$ sudo cat /proc/cpuinfo processor : 0 model name : ARMv7 Processor rev 4 (v7l) BogoMIPS : 38.40 Features : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 e evstrm crc32 CPU implementer : 0x41 CPU architecture: 7 CPU variant : 0x0 CPU part : 0xd03 CPU revision : 4 processor : 1 model name : ARMv7 Processor rev 4 (v7l) BogoMIPS : 38.40 Features : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 e evstrm crc32 CPU implementer : 0x41 CPU architecture: 7 CPU variant : 0x0 CPU part : 0xd03 CPU revision : 4 processor : 2 model name : ARMv7 Processor rev 4 (v7l) BogoMIPS : 38.40 Features : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 e evstrm crc32 CPU implementer : 0x41 CPU architecture: 7 CPU variant : 0x0 CPU part : 0xd03 CPU revision : 4 processor : 3 model name : ARMv7 Processor rev 4 (v7l) BogoMIPS : 38.40</pre>
<pre>sudo cat /proc/version</pre>	<p>Nos muestra la información asociada a los principales elementos de software en nuestra tarjeta, la versión de nuestro Kernel como la versión del compilador de C que tiene nuestro sistema operativo</p>	<pre>pi@murry:~\$ sudo cat /proc/version Linux version 5.4.83-v7+ (dom@buildbot) (gcc version 8.4.0 (#1379 SMP Mon Dec 14 13:08:57 GMT 2020) pi@murry:~\$</pre>

sudo shutdown -r now	Apaga directamente la Raspberry	
sudo apt-get update	Actualiza la lista de paquetes disponibles y versiones	
sudo apt-get upgrade	Una vez que se ha ejecutado el comando anterior, este comando descarga cada uno de los paquetes además de instalarlos	

8. Ingresar a símbolo de sistema y teclear sudo raspi-config e indicar lo observado:

El comando nos abre dentro de la consola un menú para configurar distintos apartados tanto de software puro como de interacción con hardware

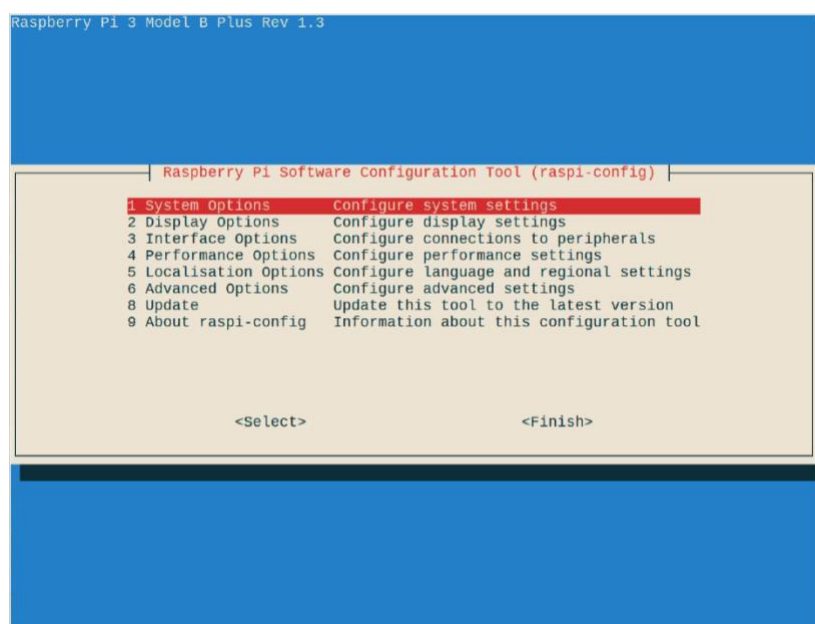


Figura 3: Pantalla de configuración del sistema mediante la terminal

9. Activar en la configuración de la Raspberry, el servicio de VNC y SSH para continuar con la siguiente parte de la práctica.

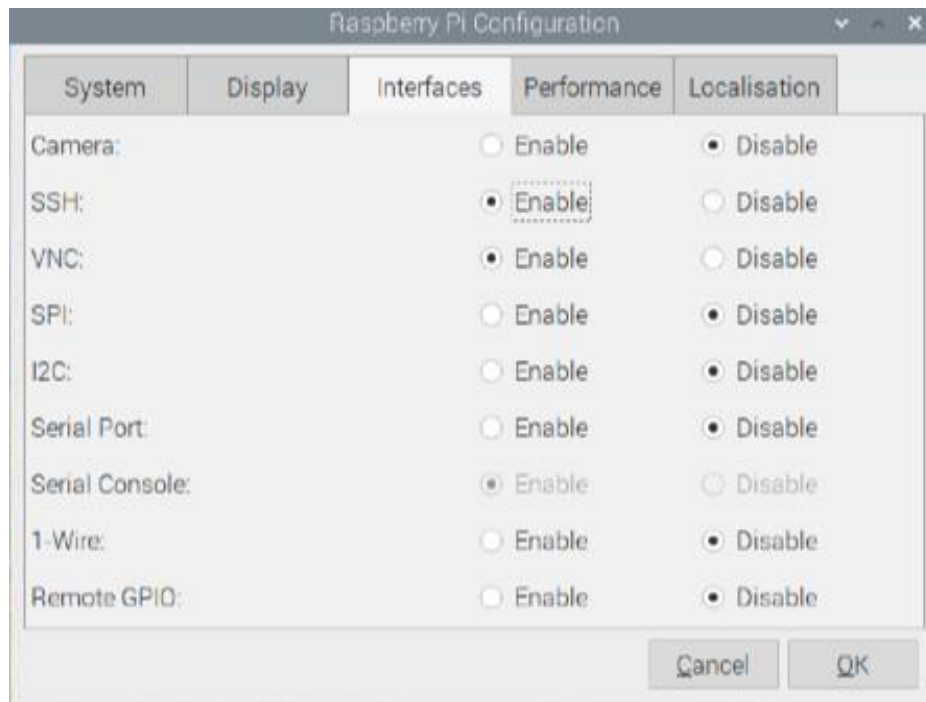


Figura 4: Habilitación de interfaces, específicamente SSH y VNC

10. Instalar en mi equipo PuTTY y Nmap para determinar la IP de la Raspberry Pi si lo requiere y utilizar nmap -sn 192.168.1.1/24 para escanear la red.

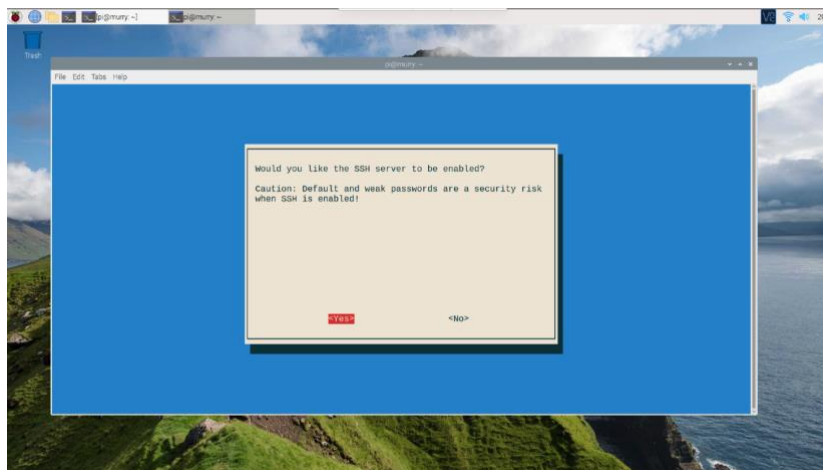
Debido a los limitados recursos para conocer nuestra dirección IP se empleó directamente el comando ifconfig en nuestra Raspberry


```
pi@murry: ~  
File Edit Tabs Help  
pi@murry:~ $ ifconfig  
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
ether b8:27:eb:7e:5b:73 txqueuelen 1000 (Ethernet)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Local Loopback)  
RX packets 5 bytes 284 (284.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 5 bytes 284 (284.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.1.109 netmask 255.255.255.0 broadcast 192.168.1.255  
inet6 fdbd:98c:ab6a:0:55a4:6f18:233f:b189 prefixlen 64 scopeid 0x0<glo  
bal>  
inet6 fe80::df4c:8a1a:cd9f:8561 prefixlen 64 scopeid 0x20<link>  
ether b8:27:eb:2b:0e:26 txqueuelen 1000 (Ethernet)  
RX packets 317275 bytes 338780530 (323.0 MiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 271913 bytes 154654645 (147.4 MiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
pi@murry:~ $
```

Figura 5: Dirección IP de nuestra Raspberry P

11.Ingresar a Raspberry Pi mediante PuTTY

Previo a ingresar con PuTTY o con algún cliente para SSH, fue necesario habilitar desde terminal SSH en nuestra tarjeta



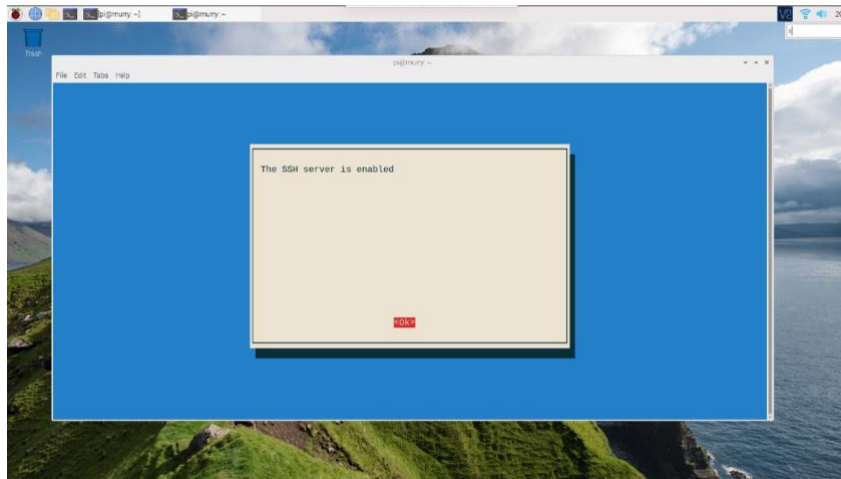


Figura 6-7: habilitación de SSH en Raspberry

Por otro lado, debido a que PuTTY como cliente no existe en Mac Os, directamente se accedió desde la terminal del sistema operativo

```

murphy — pi@murphy: ~ — ssh localhost — 97x24
murphy@Alfonso's-Mac-mini ~ % ssh pi@192.168.1.109
ssh: connect to host 192.168.1.109 port 22: Connection refused
murphy@Alfonso's-Mac-mini ~ % ssh pi@192.168.1.109
The authenticity of host '192.168.1.109 (192.168.1.109)' can't be established.
ECDSA key fingerprint is SHA256:eE/gzIkGiGojWLFKa+NNwzN3IwWj1G7LXPMWwS8o1Y.
(Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.109' (ECDSA) to the list of known hosts.
pi@192.168.1.109's password:
Linux murphy 5.4.83-v7+ #1379 SMP Mon Dec 14 13:08:57 GMT 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Sep  7 20:31:02 2021
pi@murphy:~ $
  
```

Figura 8: Conexión vía SSH a nuestra Raspberry desde una Mac Mini

12.Instalar VNC Viewer y acceder a la Raspberry Pi mediante el mismo

- A. Instalar o actualizar VNC
 - a. `sudo apt-get update`
 - b. `sudo apt-get install realvnc-vnc-server`
 - c. `sudo apt-get install realvnc-vnc-viewer`

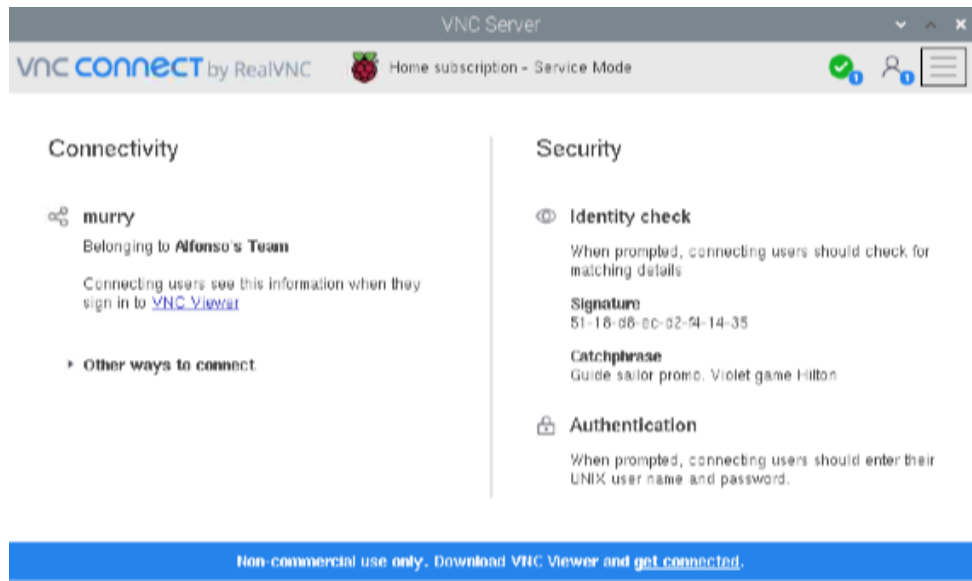


Figura 7: VNC Server corriendo en nuestra Raspberry

- B. Activar el servidor VNC
 - a. `sudo raspi-config`
- C. Instalar VNC en dispositivos externos
 - a. <https://www.realvnc.com/es/connect/download/viewer/>

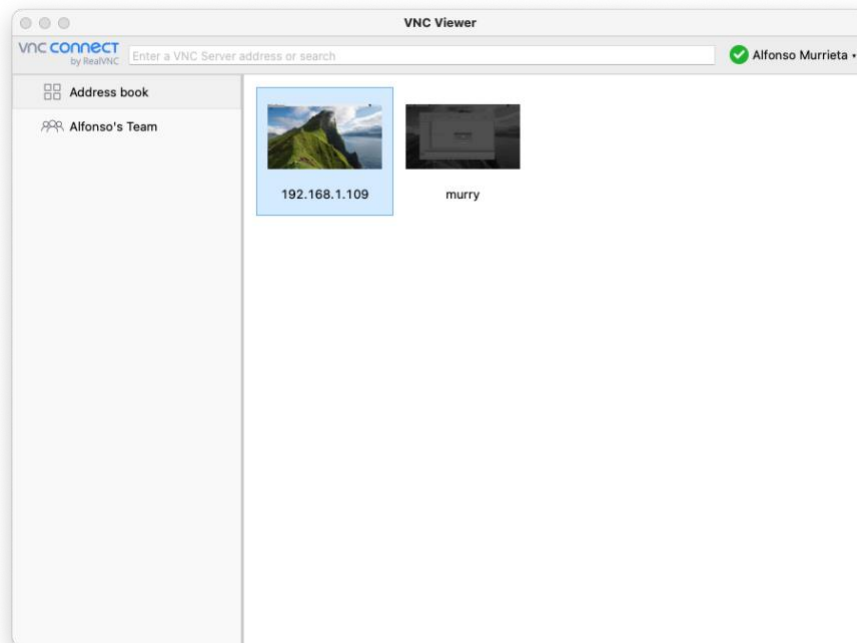


Figura 8: Funcionamiento de VNC Viewer en nuestra Raspberry Pi

13. Anotar observaciones, si es posible conectarse a la Raspberry a través de computadora, celular, tablet e identificar las diferencias.

Para este apartado debido a que previamente como equipo decidimos compartir el mismo raspberry, VNC incluso decidimos escalarlo a 1 versión Cloud lo cual nos dio la posibilidad de no solamente conectarlo localmente desde cualquier dispositivo que pueda

tener instalado VNC sino que además abrimos la posibilidad de trabajar estando remotamente.

A continuación, se muestra una captura de pantalla de la creación del ambiente en la nube de nuestra raspberry:

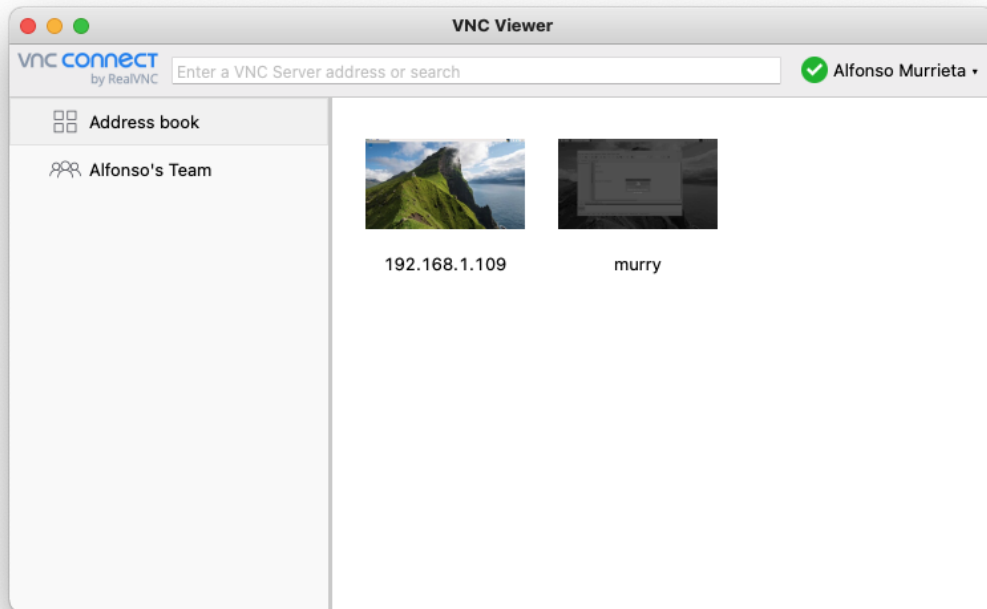


Figura 8: VNC Viewer funcionado remotamente para la conexión local (Imagen de la izquierda) y mediante Cloud (Imagen derecha)

Responda brevemente las siguientes preguntas

1. ¿Qué aprendí de esta práctica?

Como equipo aprendimos que los sistemas embebidos. Pesar de ser pequeños y de menos recursos respecto. Una computadora tradicional, siguen dando la posibilidad de poder realizar aspectos tan complejos como el montar un sistema operativo de computadora con el cual podremos realizar un sinnúmero de actividades en un futuro.

2. ¿Para qué me servirá lo aprendido en esta práctica?

Para poder trabajar en nuestras futuras prácticas de una forma más cómoda y sobre todo para trabajar de forma remota.

También para poder ampliar nuestra visión respecto a las capacidades de los sistemas embebidos

3. ¿Qué beneficios tiene el manejo remoto de mi Raspberry Pi?

En primera instancia el poder interactuar de una forma más cómoda además de no limitar la cantidad de puertos que este ofrece, pero principalmente para poder ampliar nuestras posibilidades de trabajo con los futuros sistemas que se realicen.

Conclusiones:

Cárdenas Cárdenas Jorge

Las actividades desarrolladas en esta práctica nos permitieron: instalar el sistema operativo Raspberry Pi OS (sistema operativo basado de Debian, Linux), e interactuar con el mismo, así como sus correspondientes comandos. En este sentido, y como era de esperarse, muchos de los comandos son los mismos que se emplean en cualquier distribución de Linux lo que nos da cierta familiaridad en el uso del sistema operativo de la Raspberry Pi.

Esta práctica también nos permitió probar y hacer las configuraciones necesarias para poder establecer una conexión remota con la Raspberry Pi mediante el software VNC Viewer, lo que nos permitirá, a fin de cumplir satisfactoriamente los objetivos del laboratorio, trabajar de manera colaborativa en las prácticas sucesivas.

Murrieta Villegas Alfonso

En la presente práctica aprendimos aspectos básicos de los sistemas embebidos, específicamente de uno en concreto que es Raspberry Pi. Por otro lado, nos familiarizamos de manera general con este pequeño SBC que es un ejemplo más que claro de las grandes capacidades que hoy en día un sistema embebido puede ofrecernos.

Por último, aplicamos algunos comandos básicos de Linux para observar la capacidad que nos ofrece el sistema operativo de Raspberry Pi OS además de configurar algunas herramientas para el manejo remoto de nuestro equipo.

Reza Chavarria Sergio Gabriel

La práctica dio un sustento en el manejo de la instalación y configuraciones básicas para el desarrollo de trabajos, los cuales se realizarán en las próximas actividades con Raspberry Pi.

Se manejó el Sistema Operativo de manera similar a lo que es una distribución de Linux, por lo cual el manejo de las herramientas será similar. Además, se pudo revisar la forma del trabajo remoto para tener en cuenta al trabajo cooperativo.

Valdespino Mendieta Joaquín

La practica presentada nos da una serie de información esencial para el manejo del dispositivo y ambiente integrado de la Raspberry pi, además de herramientas para el trabajo posterior en las diferentes practicas que llevaremos a cabo a lo largo del semestre, además de una forma de tener el trabajo en equipo de forma remota, como una practica introductoria al ambiente de desarrollo, he de destacar que utiliza una serie de comandos de la distribución de linux, por lo que la instalación de software y el manejo de los comandos en consola es algo familiar y fácil de entender.

Garrido Sánchez Samuel Arturo

El dispositivo Raspberry Pi es una computadora con una arquitectura ARM, y nos permitirá en este curso tener otro enfoque de manejo de interfaces de hardware conocidos como sistemas embebidos. Así, aprovecharemos los puertos disponibles en la placa para realizar conexiones a dispositivos electrónicos y manipularlos con base algoritmos con condiciones que nosotros especifiquemos. La instalación del sistema operativo se realizó vía imagen que incluye ya varios paquetes de Debian, como Python y SSH, que nos permitirá realizar las actividades. Además, se configuró para tener conexiones remotas SSH y VNC.

EXTRA:

Como extra, nos gustaría agregar que la posibilidad de trabajar remotamente mediante el servicio de Cloud ofrecido por VNC nos será de provecho para las futuras prácticas donde necesitemos trabajar como equipo al mismo tiempo y sobre todo en el mismo equipo.

Referencias:

1. Nmap. *Install Guide*. Recuperado el 11 de Septiembre de 2021, de <https://nmap.org>
2. Chiark. *PuTTY: a free SSH and Telnet Client*. Recuperado el 11 de Septiembre de 2021, de <https://www.chiark.greenend.org.uk/~sgtatham/putty/>
3. Raspberry Pi. *Raspberry Pi OS*. Recuperado el 11 de Septiembre de 2021, de <https://www.raspberrypi.org/software/>