

Arduinom

Instalación desde cero en Raspberry Pi 3 Model B v1.2

Instalación Raspbian, Node.js y descargas de Repositorios Github

Raspberry Pi 3 Model B V1.2

Manual de instalación escrito por:

Andres Camilo Vargas Román (e-mail: andresvargasr@gmail.com, Github: andresvargasr)

1. Raspbian

1.2. Descarga

Para el caso a tratar se trabaja con el Sistema Operativo Raspbian versión Stretch de Septiembre del 2017¹, ver Figura 1.

¹ Disponible en: <https://www.raspberrypi.org/downloads/raspbian/>.

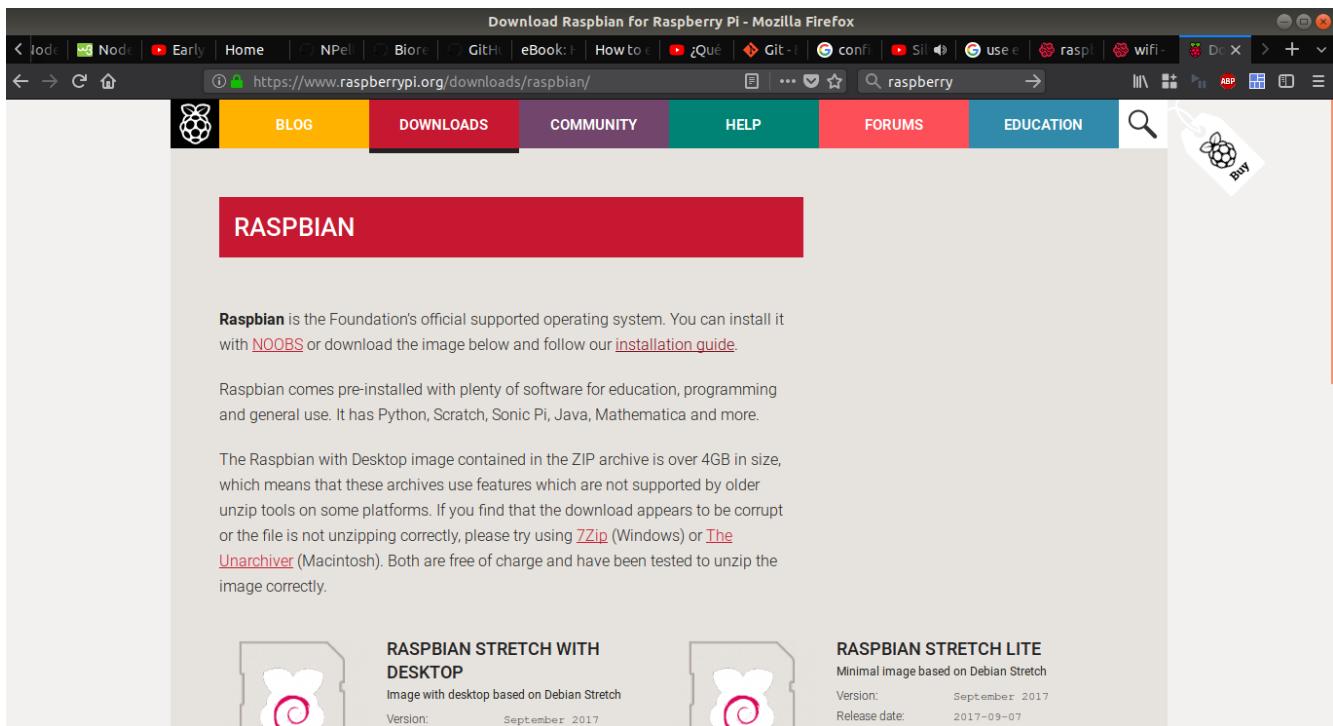


Figura 1. Página de descarga de Raspbian versión Stretch.

En este primer caso se ha optado por emplear la versión con escritorio (*Raspbian Stretch with Desktop*).

1.3. Instalación

Para realizar la instalación del Sistema Operativo se recomienda emplear una tarjeta microSD clase 10 que garantiza una velocidad mínima de escritura de 10 MB/s²; igualmente es útil realizar todo desde un entorno Linux, puesto que varias opciones de trabajo requeridas en este paso se realizan sin la necesidad de alguna instalación adicional de software o componentes.

En la página de *Raspberry* (ver Figura 2) se puede encontrar la información de instalación³, dicho enlace contiene las recomendaciones para copiar todos los datos en la tarjeta que será la memoria de la *Raspberry*.

-
- 2 Para conocer mejor las características de almacenamiento y velocidad de las memorias microSD, visite: <https://www.xataka.com/especiales/como-elegir-la-tarjeta-microsd-adecuada-para-tu-smartphone>.
 - 3 Para la guía completa de instalación en inglés, visite: <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>.

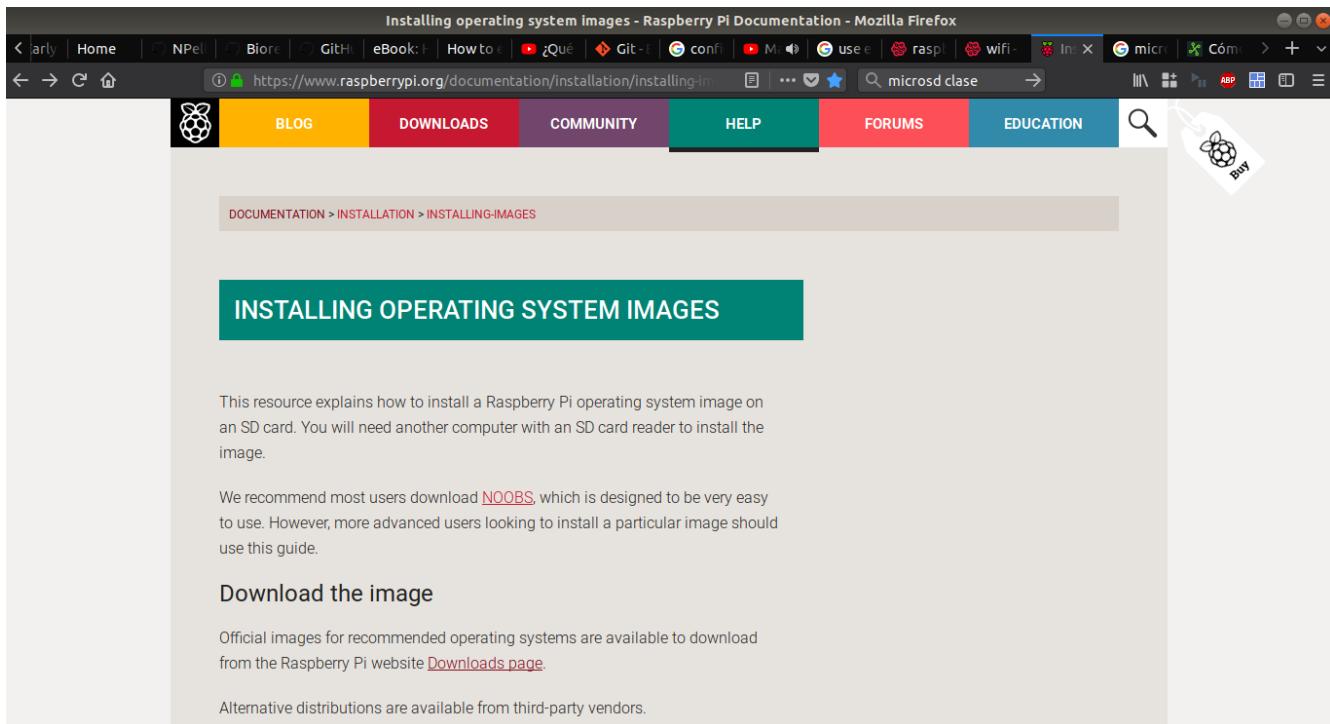


Figura 2. Guía de instalación de Raspbian.

En este caso, accedemos al enlace al final de esta página y seleccionamos la instalación a través de Linux⁴ (ver Figura 3), el cual presenta una página con los pasos a seguir a través del Terminal del S.O. (Sistema Operativo).

⁴ El Sistema Operativo empleado para transferir los datos es Ubuntu 17.10 de 64 bits.

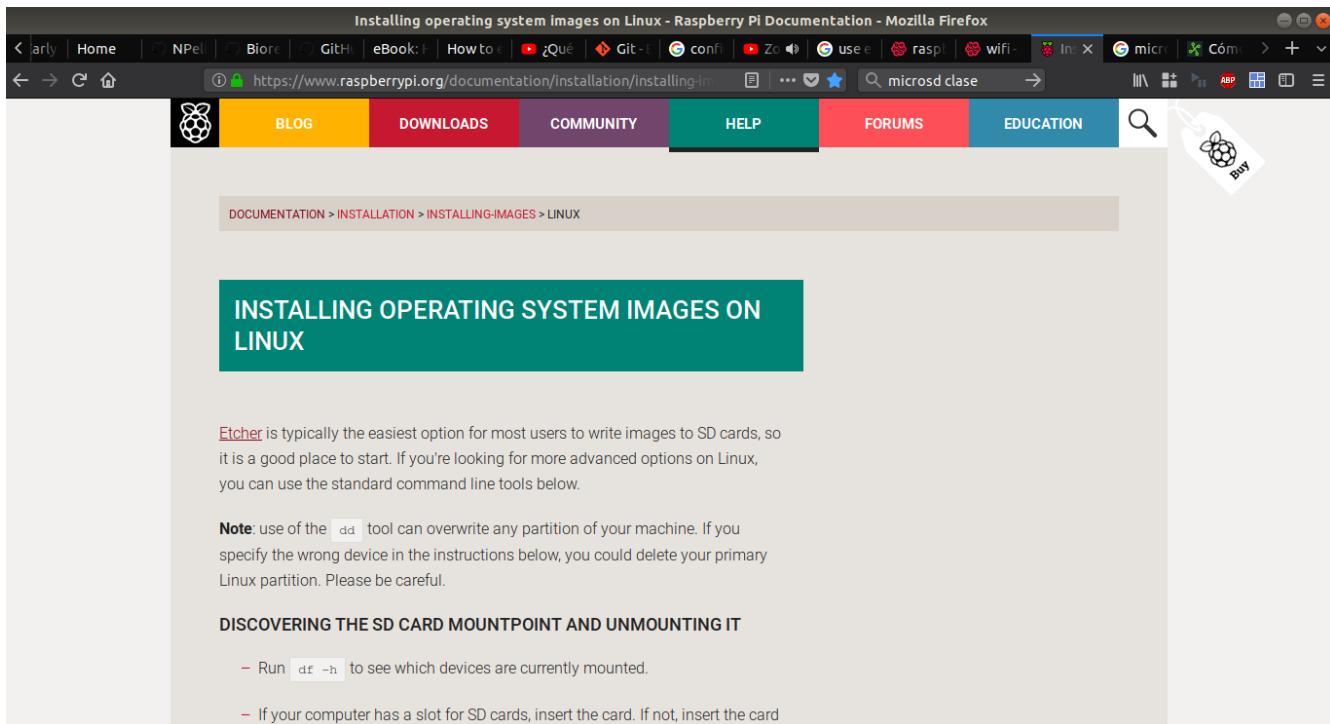


Figura 3. Instrucciones de instalación para Linux.

- Inserte la tarjeta al computador e identifique el punto de montaje (*mountpoint*).
- Ejecute *df -h* para conocer cuál dispositivo esta actualmente montado (ver Figura 4).

The screenshot shows a terminal window with the title "andres@ACV-03: ~". The menu bar includes "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The terminal command "df -h" is run, displaying disk usage information. The output shows various partitions and their usage. Two new partitions are listed under the SD card mount point:

Sistema de Archivos	Tamaño	Usados	Disp	Uso%	Montado en
udev	3,9G	0	3,9G	0%	/dev
tmpfs	787M	9,7M	777M	2%	/run
/dev/sda6	67G	61G	2,6G	96%	/
tmpfs	3,9G	46M	3,8G	2%	/dev/shm
tmpfs	5,0M	4,0K	5,0M	1%	/run/lock
tmpfs	3,9G	0	3,9G	0%	/sys/fs/cgroup
/dev/loop0	173M	173M	0	100%	/snap/atom/56
/dev/loop1	53M	53M	0	100%	/snap/go/1016
/dev/loop2	84M	84M	0	100%	/snap/core/3440
/dev/loop3	186M	186M	0	100%	/snap/pycharm-community/33
tmpfs	787M	40K	787M	1%	/run/user/1000
/dev/mmcblk0p2	15G	4,6G	9,4G	33%	/media/andres/b4ea8e46-fe87-4ddd-9e94-506c37005ac5
/dev/mmcblk0p1	42M	21M	21M	51%	/media/andres/boot

Figura 4. Punto de montaje de la tarjeta SD.

En este caso, al tener instalado un Sistema Raspbian anterior⁵ aparecerán dos particiones para la tarjeta SD⁶: `/dev/mmcblk0p2` y `/dev/mmcblk0p1` con un tamaño de 15GB y 42MB respectivamente.

- Ejecute `umount /dev/mmcblk0p1` donde `mmcblk0p1` son cada partición de la tarjeta SD. A continuación corra nuevamente `df -h` para identificar que se han desmontado correctamente las particiones (ver Figura 5).

⁵ Se recomienda realizar previamente una copia de seguridad usando esta guía: <https://www.atareao.es/raspberry/copia-de-seguridad-de-tu-raspberry-pi/>.

⁶ Usualmente se identifican 2 nombres para las particiones `/dev/mmcblk0` ó `/dev/sdX`.

```

andres@ACV-03: ~
Archivo Editar Ver Buscar Terminal Ayuda
tmpfs           3,9G      0  3,9G  0% /sys/fs/cgroup
/dev/loop0     173M    173M  0 100% /snap/atom/56
/dev/loop1      53M    53M  0 100% /snap/go/1016
/dev/loop2      84M    84M  0 100% /snap/core/3440
/dev/loop3     186M   186M  0 100% /snap/pycharm-community/33
tmpfs          787M    60K  787M  1% /run/user/1000
/dev/mmcblk0p1   42M    21M  21M  51% /media/andres/boot
/dev/mmcblk0p2   15G   4,6G  9,4G  33% /media/andres/b4ea8e46-fe87-4ddd-9e94-50
6c37005ac5
andres@ACV-03:~$ umount /dev/mmcblk0p1 && umount /dev/mmcblk0p2
andres@ACV-03:~$ df -h
S.ficheros      Tamaño Usados  Disp Uso% Montado en
udev            3,9G      0  3,9G  0% /dev
tmpfs          787M    9,7M  777M  2% /run
/dev/sda6        67G    61G  2,6G  96% /
tmpfs          3,9G    31M  3,9G  1% /dev/shm
tmpfs          5,0M   4,0K  5,0M  1% /run/lock
tmpfs          3,9G      0  3,9G  0% /sys/fs/cgroup
/dev/loop0     173M    173M  0 100% /snap/atom/56
/dev/loop1      53M    53M  0 100% /snap/go/1016
/dev/loop2      84M    84M  0 100% /snap/core/3440
/dev/loop3     186M   186M  0 100% /snap/pycharm-community/33
tmpfs          787M    56K  787M  1% /run/user/1000
andres@ACV-03:~$ 

```

Figura 5. Ejecución de `umount` y verificación de particiones montadas.

- Copiando los datos a la tarjeta SD

Con el fin de copiar el S.O. en la tarjeta SD, se debe ubicar en la carpeta donde se encuentra el archivo .zip descargado de la página oficial⁷. A continuación:

- ejecute el siguiente comando `unzip -p 2017-09-07-raspbian-stretch.zip | sudo dd of=/dev/mmcblk0 bs=4M status=progress conv=fsync` (Figura 6).

⁷ Disponible en: <https://www.raspberrypi.org/downloads/raspbian/>.


```
andres@ACV-03: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
└─sda3      8:3    0 607,5G  0 part
└─sda4      8:4    0     1K  0 part
└─sda5      8:5    0   7,9G  0 part
└─sda6      8:6    0   68,3G  0 part /
sr0        11:0   1 1024M  0 rom
mmcblk0    179:0   0 14,9G  0 disk
└─mmcblk0p1 179:1   0  41,8M  0 part
└─mmcblk0p2 179:2   0   4,5G  0 part
andres@ACV-03:~/Descargas$ unzip -p 2017-09-07-raspbian-stretch.zip | sudo dd of
=/dev/mmcblk0 bs=4M status=progress conv=sync
[sudo] password for andres:
15921577984 bytes (16 GB, 15 GiB) copied, 1265,09 s, 12,6 MB/s
dd: error al escribir en '/dev/mmcblk0': No queda espacio en el dispositivo
0+3799 registros leídos
3798+0 registros escritos
15931539456 bytes (16 GB, 15 GiB) copied, 1344,04 s, 11,9 MB/s
andres@ACV-03:~/Descargas$ unzip -p 2017-09-07-raspbian-stretch.zip | sudo dd of
=/dev/mmcblk0 bs=4M status=progress conv=fsync
[sudo] password for andres:
4912513024 bytes (4,9 GB, 4,6 GiB) copied, 342,006 s, 14,4 MB/s
0+52372 registros leídos
0+52372 registros escritos
4916019200 bytes (4,9 GB, 4,6 GiB) copied, 420,106 s, 11,7 MB/s
andres@ACV-03:~/Descargas$
```

Figura 6. Código para Instalación de Raspbian en tarjeta SD.

- Verificar instalación

Después de que se complete la copia de los archivos a la SD, se extrae la tarjeta del computador y se conecta a la *Raspberry* junto con una pantalla con conexión HDMI, un teclado y un ratón (ver Figura 7).

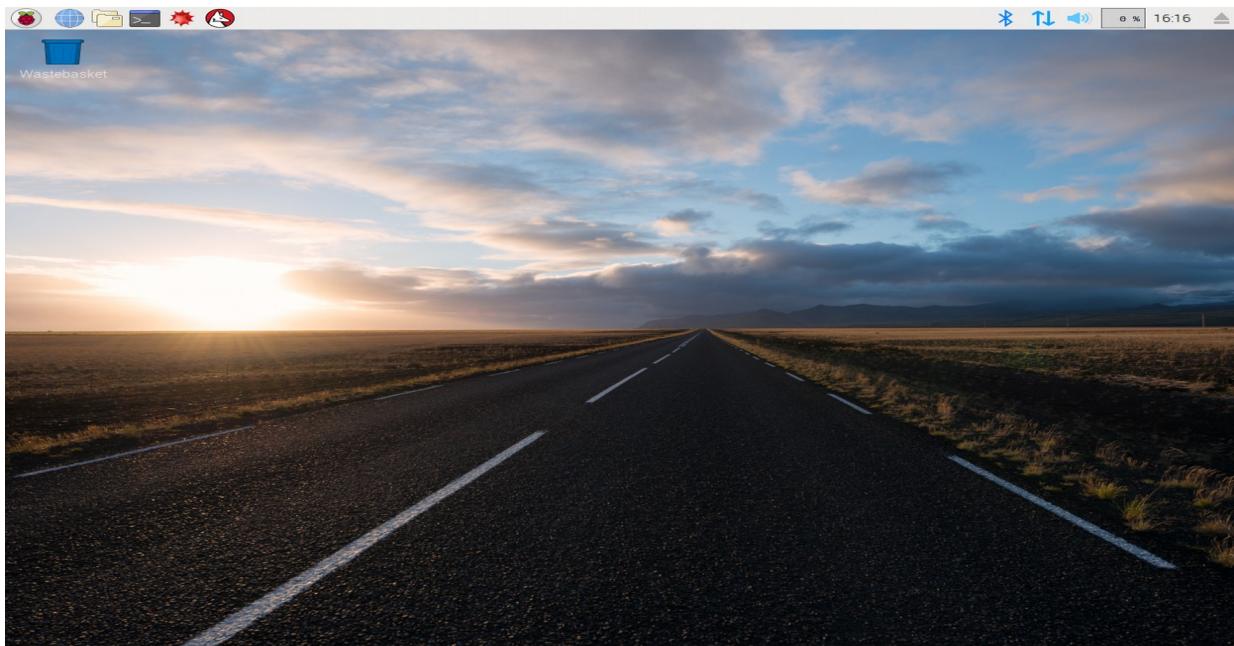


Figura 7. Escritorio Raspbian en Raspberry Pi 3 Model B v1.2

2. (Opcional) Descargar actualizaciones y configuración de red

Los pasos descritos a continuación se pueden encontrar en la página de W3Schools⁸

2.2. Conexión a la red inalámbrica

- Abra la terminal y configure la red del Raspberry Pi
 - Escanee las redes inalámbricas disponibles ejecutando el siguiente código: `sudo iwlist wlan0 scan` (ver Figura 8).

⁸ Visitar: https://www.w3schools.com/nodejs/nodejs_raspberryipi.asp.

```

pi@raspberrypi:~ $ sudo iwlist wlan0 scan
wlan0      Scan completed :
          Cell 01 - Address: 74:3E:2B:39:4A:F8
          Channel:1
          Frequency:2.412 GHz (Channel 1)
          Quality=70/70  Signal level=-25 dBm
          Encryption key:on
          ESSID:"univalle"
          Bit Rates:12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s; 48 Mb/s
                     54 Mb/s
          Mode:Master
          Extra:tsf=0000000000000000
          Extra: Last beacon: 90ms ago
          IE: Unknown: 0008756E6976616C6C65
          IE: Unknown: 01069824B048606C
          IE: Unknown: 030101
          IE: Unknown: 0706434F20010B24
          IE: Unknown: 2A0100
          IE: Unknown: DD180050F2020101850003A4000027A4000042435E00623
22F00
          IE: IEEE 802.11i/WPA2 Version 1
              Group Cipher : CCMP
              Pairwise Ciphers (1) : CCMP
              Authentication Suites (2) : PSK unknown (4)

```

Figura 8. Escaneo de redes inalámbricas.

- Después de identificar la red a la cuál se va a conectar debe recordar el ESSID, para este caso es: ESSID:"univalle". Ejecute el comando para agregar la red y conectarse a través de la red inalámbrica: *sudo nano /etc/wpa_supplicant/wpa_supplicant.conf* (ver Figura 9).

```

pi@raspberrypi:~ $ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
network={

          Authentication Suites (1) : PSK
          IE: Unknown: 32043048606C
          IE: Unknown: 2D1AEF1103FFFF000000000000000000000080000000000000000
0000000000
          IE: Unknown: 331AEF1103FFFF000000000000000000000080000000000000000
0000000000
          IE: Unknown: 3D16050D000000000000000000000000000000000000000000000000000000
0
          IE: Unknown: 3416050D0000000000000000000000000000000000000000000000000000000000000
0
          IE: WPA Version 1
              Group Cipher : CCMP
              Pairwise Ciphers (1) : CCMP
              Authentication Suites (1) : PSK
          IE: Unknown: DD180050F2020101850003A4000027A4000042435E00623
22F00
          IE: Unknown: DD0900037F01010000FF7F
          IE: Unknown: DD9B0050F204104A0001101044000102103B00010310470
010000102030405060708090A0B0C0D0E0F1021000754502D4C494E4B1023000A544C2D575238343
1485010240003322E3010420003312E301054000800060050F20400011011001A576972656C65737
320526F7574657220544C2D57523834314850100800020086103C000101104900140024E26002000
101600000020001600100020001
pi@raspberrypi:~ $ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

```

Figura 9. Comando para modificar la red inalámbrica.

- A continuación, escriba la siguiente información en el archivo (ver Figura 10):

network={

```

ssid="univalle"
psk="Univalle"
}

```

```

pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.7.4  File: /etc/wpa_supplicant/wpa_supplicant.conf
country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="univalle"
    psk="Univalle"
}

[ Read 8 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

Figura 10. Almacenamiento de los datos de la red inalámbrica.

- Reinicie el S.O. para ajustar la configuración: *sudo reboot*.
- Para conocer si efectivamente se ha establecido la comunicación con la red inalámbrica, se ejecuta el comando *ifconfig wlan0* de modo que se logre observar la IPv4 con su respectiva máscara y difusión (ver Figura 11).

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.84.17.38 netmask 255.255.224.0 broadcast 10.84.31.255
              inet6 fe80::1aa9:95b8:8110:71ac prefixlen 64 scopeid 0x20<link>
                ether b8:27:eb:99:6a:86 txqueuelen 1000 (Ethernet)
                  RX packets 2071 bytes 190639 (186.1 KiB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 75 bytes 10035 (9.7 KiB)
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
pi@raspberrypi:~ $ 

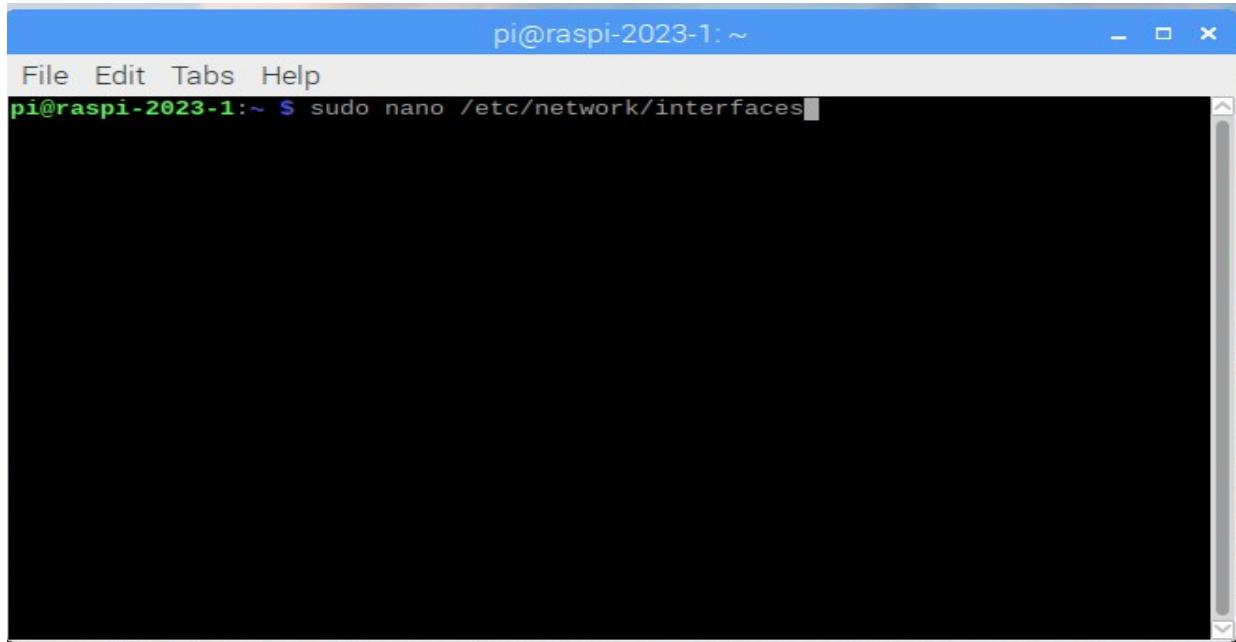
```

Figura 11. Verificación de acceso a la red inalámbrica.

2.2. Establecer conexión Ethernet (punto a punto – ad hoc)

En el caso de la red univalle la comunicación a través de ssh no se puede establecer debido a que el puerto de comunicación se encuentra bloqueado, por consiguiente se establecerá la comunicación a través de una comunicación Ethernet y ajustando los parámetros de conexión de modo manual.

- Modificar el archivo *interfaces* mediante el siguiente código: *sudo nano /etc/network/interfaces* (*Figura 12*).

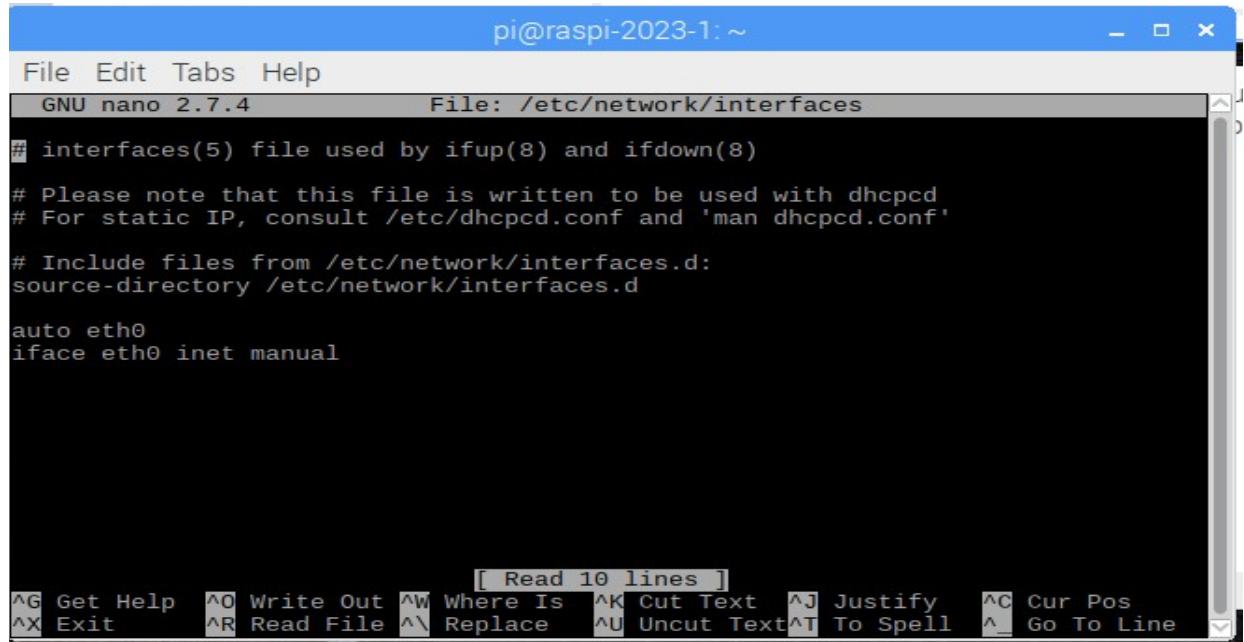


The screenshot shows a terminal window with a blue header bar containing the text 'pi@raspi-2023-1: ~'. Below the header is a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The main area of the terminal is black, and the command 'pi@raspi-2023-1:~ \$ sudo nano /etc/network/interfaces' is visible at the top of the screen. A vertical scroll bar is on the right side of the terminal window.

Figura 12. Comando para acceder a la configuración de los puertos de red.

- Ingrese el siguiente texto en el documento para configurar el puerto eth0 de modo manual y después guarde el archivo presionando *Ctrl+x* sin cambiar el nombre (ver Figura 13):

```
auto eth0
iface eth0 inet manual
```



```
pi@raspi-2023-1: ~
File Edit Tabs Help
GNU nano 2.7.4          File: /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto eth0
iface eth0 inet manual

[ Read 10 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit     ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

Figura 13. Configuración manual para el acceso por puerto ethernet - eth0.

- Después, se debe modificar el archivo: `sudo nano /etc/dhcpcd.conf`, en este caso basta con eliminar los numerales (#) tanto de *interfaces eth0* como de *static ip_address=192.168.0.10/24* (ver Figura 14).


```

pi@raspi-2023-1: ~
File Edit Tabs Help
GNU nano 2.7.4          File: /etc/dhcpcd.conf          Modified
# Respect the network MTU. This is applied to DHCP routes.
option interface_mtu

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

# Example static IP configuration:
interface eth0
Static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^L Replace  ^U Uncut Text  ^T To Spell  ^_ Go To Line

```

Figura 14. Asignación de IP fija al puerto ethernet eth0.

- Guarde el archivo y reinicie la Raspberry.
- Finalmente, se pueden revisar las conexiones de red escribiendo el comando *ifconfig* donde se encuentra la configuración para *eth0* y *wlan0* (ver Figura 15).

```

pi@raspi-2023-1: ~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.10  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::3355:f28c:adc6:d084  prefixlen 64  scopeid 0x20<link>
          ether b8:27:eb:cc:3f:d3  txqueuelen 1000  (Ethernet)
            RX packets 24  bytes 2389 (2.3 KiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 30  bytes 4248 (4.1 KiB)
            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 1  (Local Loopback)
            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

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.84.17.38  netmask 255.255.224.0  broadcast 10.84.31.255
        inet6 fe80::b70d:b11c:ac9:8e66  prefixlen 64  scopeid 0x20<link>
          ether b8:27:eb:99:6a:86  txqueuelen 1000  (Ethernet)
            RX packets 508  bytes 63602 (62.1 KiB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 64  bytes 8626 (8.4 KiB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
pi@raspi-2023-1: ~ $ 

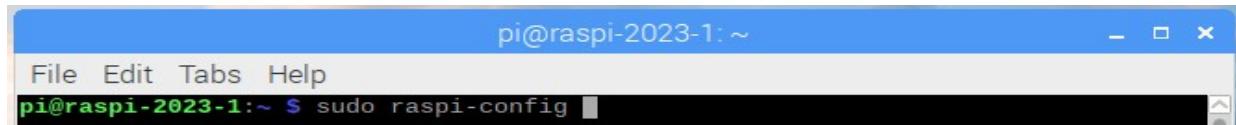
```

Figura 15. Conexiones inalámbrica e ethernet.

Después de estas configuraciones, el Raspberry tendrá acceso a internet a través de su conexión inalámbrica sobre la red *univalle* y se podrá administrar remotamente mediante el protocolo ssh utilizando *ethernet*.

2.3. Habilitando SSH, cambiando el *Hostname* y contraseña

- Ejecute el siguiente comando para acceder a las configuraciones del Raspberry Pi: *sudo raspi-config* (*Figura 16*).



```
pi@raspi-2023-1:~ $ sudo raspi-config
```

Figura 16. Comando de configuración de la Raspberry.

- (Figura 17) Diríjase al numeral 5 *Interfacing Options*.

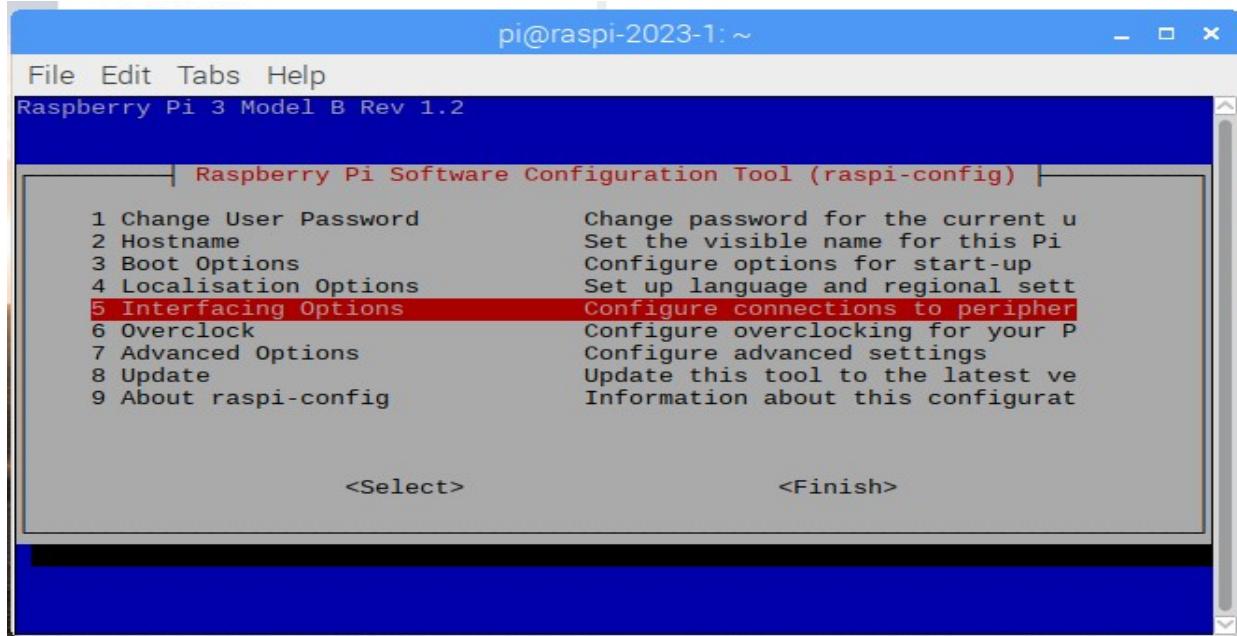


Figura 17. Opciones de interfaz.

- A continuación, seleccione P2 SSH (ver Figura 18).

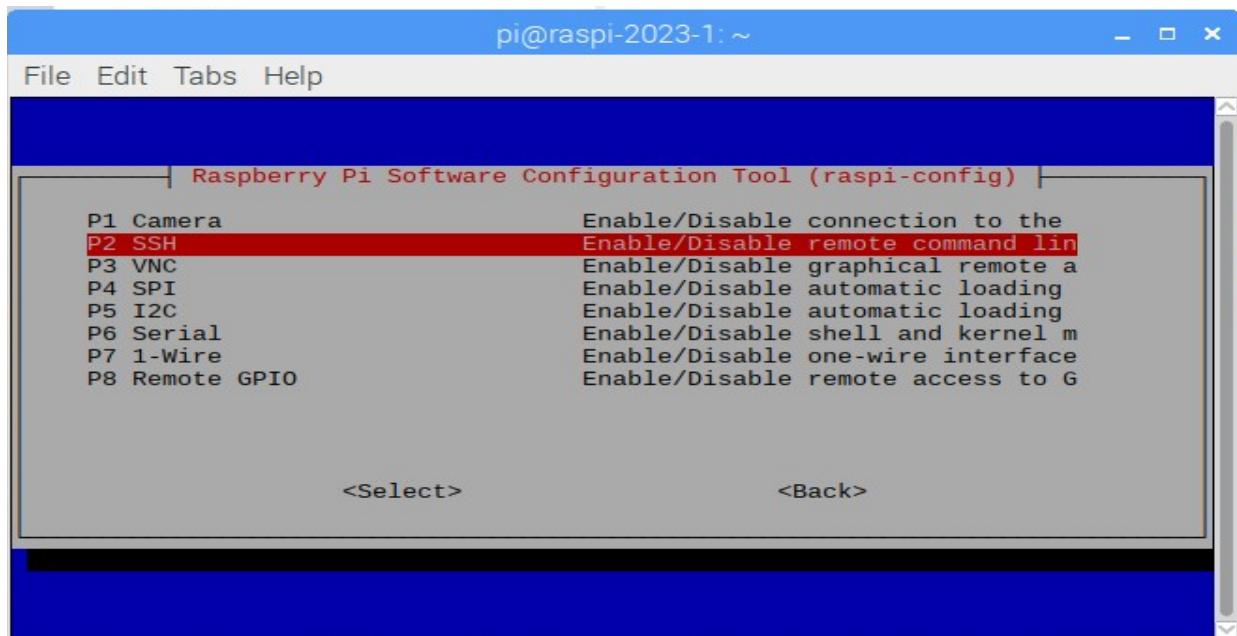


Figura 18. Habilitar SSH.

- Después de habilitar la conexión SSH, se debe configurar la contraseña de usuario, por lo tanto, ingrese a la opción 1 *Change User Password* (ver Figura 19).

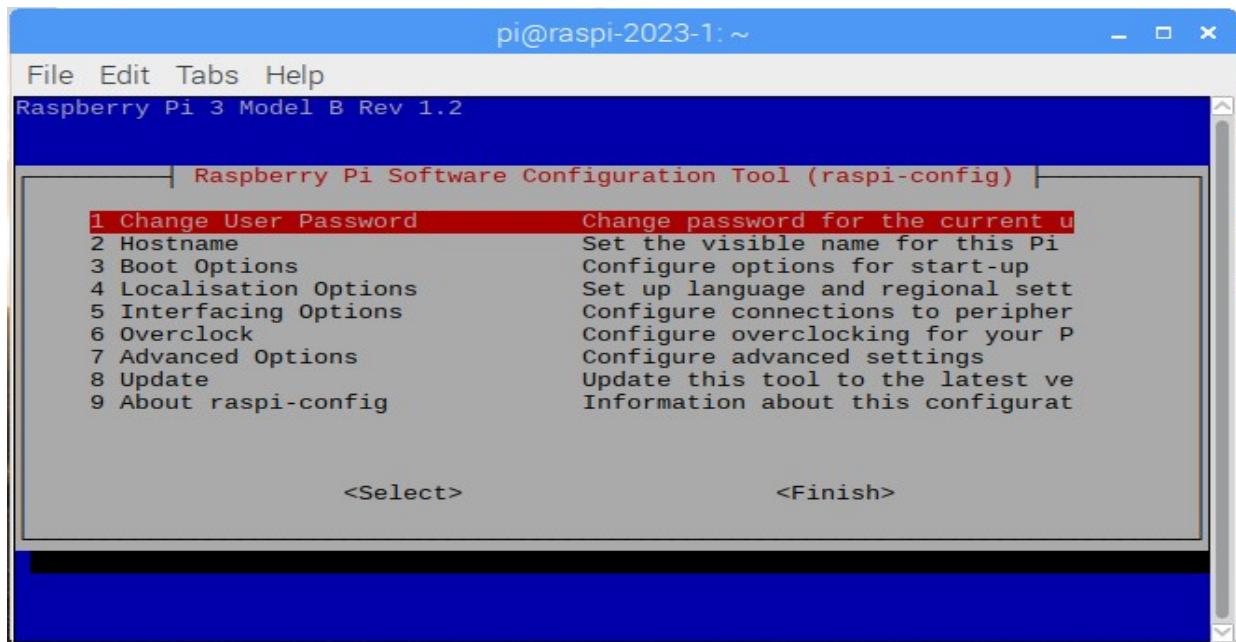


Figura 19. Cambiar contraseña de usuario.

- Para este caso, la contraseña se ha establecido como: “instru2023” (Figura 20).

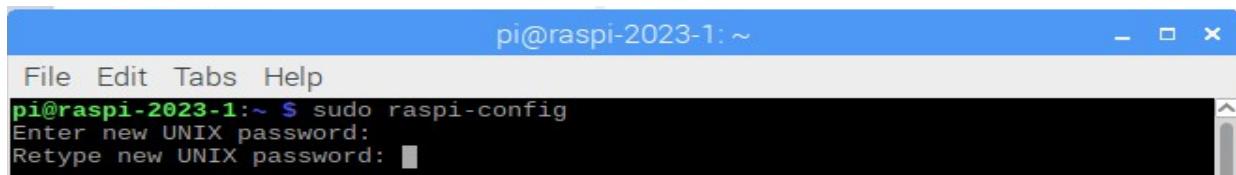


Figura 20. Ingreso de contraseña.

- Para el siguiente paso, se ingresa a la opción 2 *Hostname* y se escribe *raspi-2023-1* (ver Figura 21 y Figura 22).

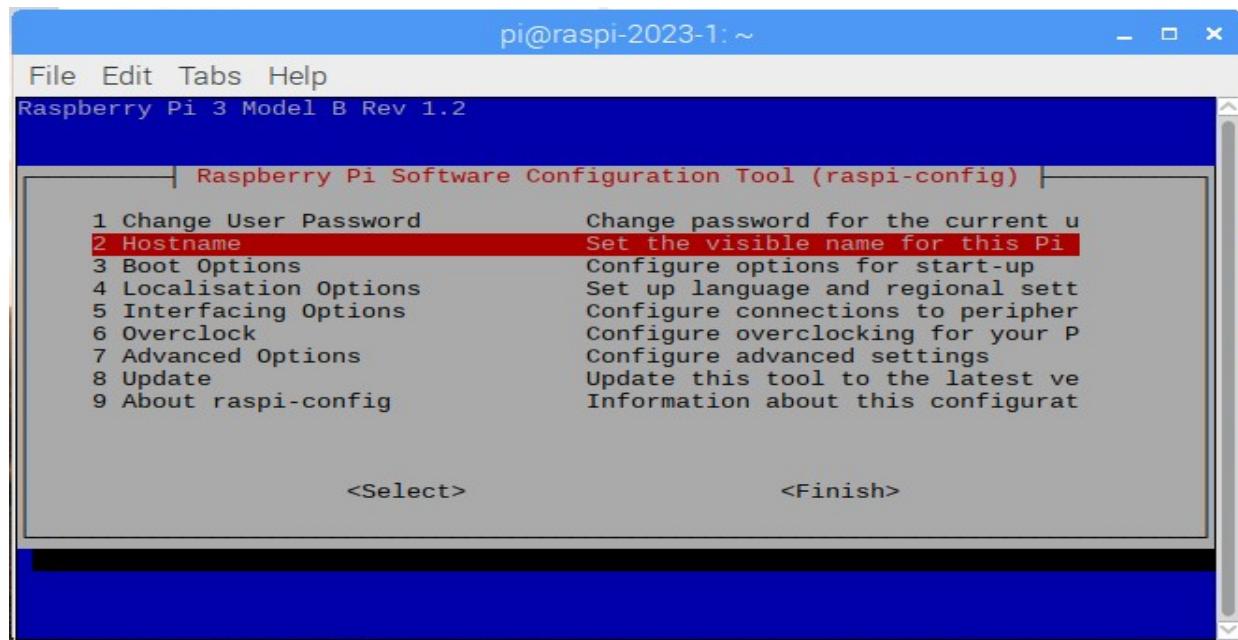


Figura 21. Cambio de nombre host.

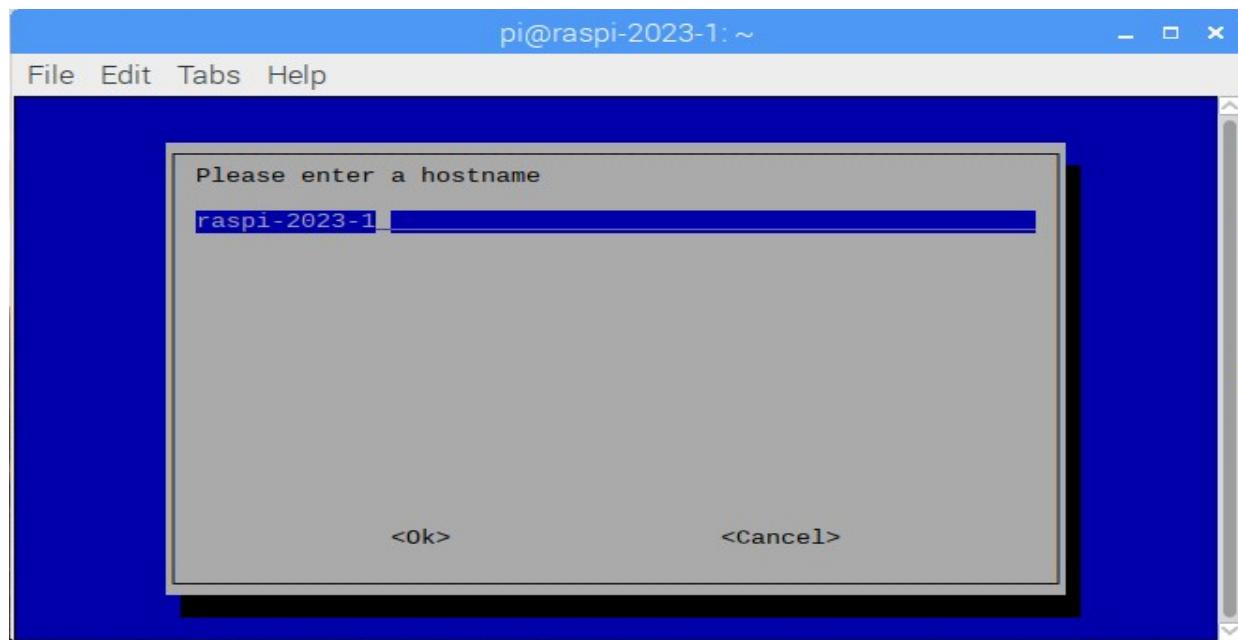


Figura 22. Ingreso de nuevo nombre host.

- Para finalizar la configuración, se solicita reiniciar el sistema (ver Figura 23).



Figura 23. Reiniciar para completar configuración del sistema.

2.3.1. Establecer conexión SSH en Linux

- En la terminal del computador conectado por *ethernet* a la Raspberry se escribe: ssh [-Y] pi@192.168.0.10, a continuación se solicita la contraseña que se creó anteriormente (ver Figura 24).

```
andres@ACV-03: ~/Descargas$ ssh pi@192.168.0.10
The authenticity of host '192.168.0.10 (192.168.0.10)' can't be established.
ECDSA key fingerprint is SHA256:+o1K0PJHWesB1Lz580WKdRx3PVlmYbRtK9o9ouIBK8M.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.10' (ECDSA) to the list of known hosts.
pi@192.168.0.10's password:
Linux raspi-2023-1 4.9.41-v7+ #1023 SMP Tue Aug 8 16:00:15 BST 2017 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.
```

Figura 24. Establecer comunicación a través de SSH.

- Antes de instalar Node.js, se recomienda actualizar la lista de paquetes del sistemas mediante *sudo apt-get update* y posterior a esto, actualizar los paquetes a su última versión con el comando *sudo apt-get dist-upgrade*⁹ (ver Figura 25).

```
andres@ACV-03: ~/Descargas
Archivo Editar Ver Buscar Terminal Ayuda
Last login: Thu Sep 7 17:02:17 2017
pi@raspi-2023-1:~ $ sudo apt-get update
Get:1 http://mirrordirector.raspbian.org/raspbian stretch InRelease [15.0 kB]
Get:2 http://mirrordirector.raspbian.org/raspbian stretch/main armhf Packages [1
1.7 MB]
Get:3 http://archive.raspberrypi.org/debian stretch InRelease [25.3 kB]
Get:4 http://archive.raspberrypi.org/debian stretch/main armhf Packages [123 kB]
Get:5 http://archive.raspberrypi.org/debian stretch/ui armhf Packages [27.0 kB]
Get:6 http://mirrordirector.raspbian.org/raspbian stretch/non-free armhf Package
s [95.2 kB]
Fetched 11.9 MB in 1min 12s (165 kB/s)
Reading package lists... Done
pi@raspi-2023-1:~ $ sudo apt-get dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
```

Figura 25. Actualización del sistema operativo.

3. Node.js

⁹ Se recomienda realizar este paso frecuentemente con el fin de mantener al día el S.O.

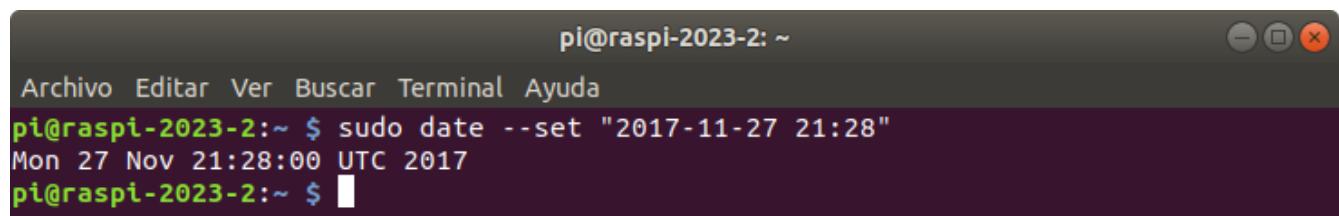
Con el fin de evitar errores y obtener las actualizaciones recientes y estables de los componentes empleados, se descargan las versiones del *Node Version Manager – NVM* (v0.33.6), Node.js (v8.9.1 LTS), Serialport (6.0.4 con soporte para Node.js v8.9.1 LTS) y MongoDB (v2.2.30).

3.1. Actualización de fecha

Con el fin de evitar posibles errores al momento de descargar o instalar los paquetes, se ajusta la fecha del sistema.

- Ejecutar el comando (ver Figura 26):

```
date --set "2017-11-24 17:27"
```



```
pi@raspi-2023-2: ~
Archivo Editar Ver Buscar Terminal Ayuda
pi@raspi-2023-2:~ $ sudo date --set "2017-11-27 21:28"
Mon 27 Nov 21:28:00 UTC 2017
pi@raspi-2023-2:~ $ █
```

Figura 26. Configuración de la hora y fecha del sistema.

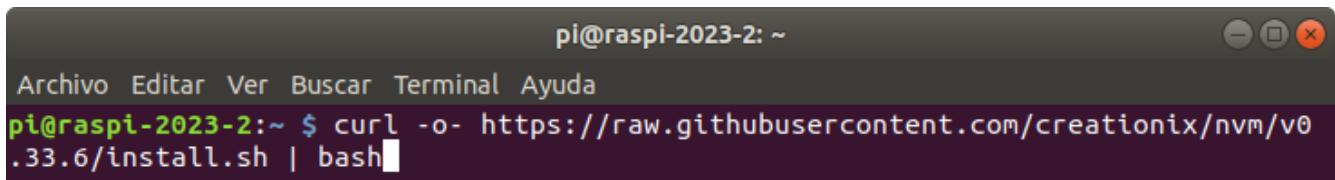
3.2. Instalación de NVM

Para obtener la versión más reciente de Node.js, se emplea NVM¹⁰, puesto que al ser un gestor de paquetes para Node.js, brinda las versiones más actualizadas de cada uno de los complementos necesarios.

- Ejecutar el siguiente comando de instalación (ver Figura 27):

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.6/install.sh | bash
```

10 Visitar: <https://github.com/creationix/nvm>.



```
pi@raspi-2023-2: ~
Archivo Editar Ver Buscar Terminal Ayuda
pi@raspi-2023-2:~ $ curl -o- https://raw.githubusercontent.com/creationix/nvm/v0
.33.6/install.sh | bash
```

Figura 27. Instalación de Node Version Manager - NVM.

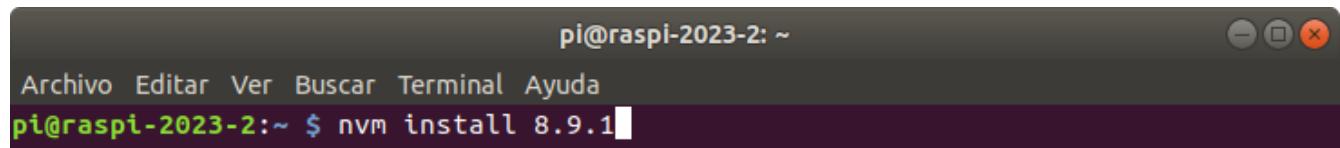
3.3. Instalación de Node.js

En este caso, teniendo instalado NVM, la instalación de Node.js se ejecuta desde este paquete, de modo que es posible seleccionar la versión a instalar, que para este caso, se instala la versión 8.9.1 LTS.

3.3.1. Ajuste de la fecha y hora

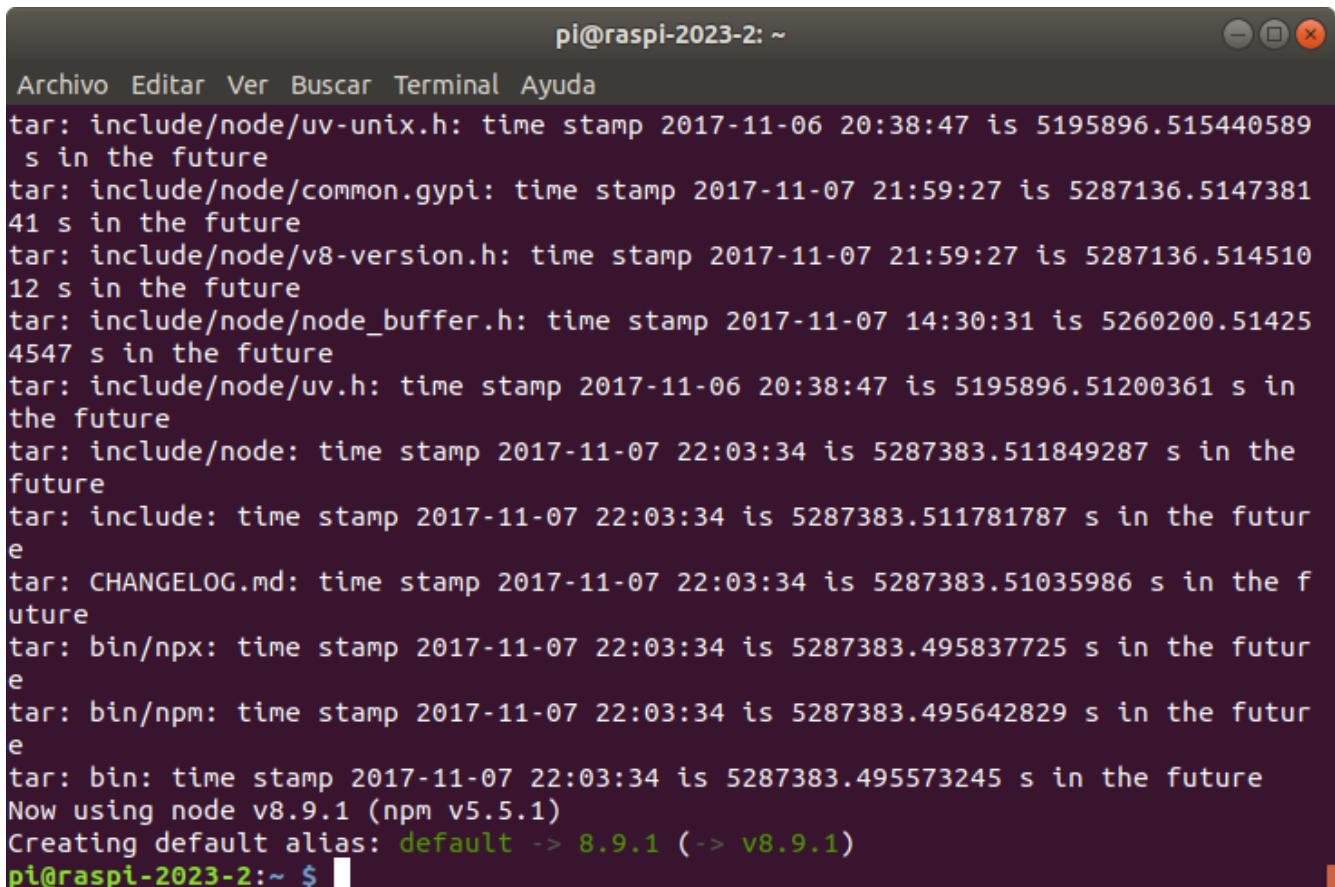
- Ejecuta el siguiente comando (ver Figura 28 y Figura 29):

```
nvm install 8.9.1
```



A screenshot of a terminal window titled "pi@raspi-2023-2: ~". The window has a dark grey header bar with the title and a dark grey footer bar. The main area is white. At the top of the white area, there is a menu bar with options: Archivo, Editar, Ver, Buscar, Terminal, Ayuda. Below the menu bar, the command "pi@raspi-2023-2:~ \$ nvm install 8.9.1" is visible, with the prompt "\$" highlighted in red.

Figura 28. Comando de instalación de Node.js.



```
pi@raspi-2023-2: ~
Archivo Editar Ver Buscar Terminal Ayuda
tar: include/node/uv-unix.h: time stamp 2017-11-06 20:38:47 is 5195896.515440589
  s in the future
tar: include/node/common.gypi: time stamp 2017-11-07 21:59:27 is 5287136.5147381
41 s in the future
tar: include/node/v8-version.h: time stamp 2017-11-07 21:59:27 is 5287136.514510
12 s in the future
tar: include/node/node_buffer.h: time stamp 2017-11-07 14:30:31 is 5260200.51425
4547 s in the future
tar: include/node/uv.h: time stamp 2017-11-06 20:38:47 is 5195896.51200361 s in
the future
tar: include/node: time stamp 2017-11-07 22:03:34 is 5287383.511849287 s in the
future
tar: include: time stamp 2017-11-07 22:03:34 is 5287383.511781787 s in the futur
e
tar: CHANGELOG.md: time stamp 2017-11-07 22:03:34 is 5287383.51035986 s in the f
uture
tar: bin/npx: time stamp 2017-11-07 22:03:34 is 5287383.495837725 s in the futur
e
tar: bin/npm: time stamp 2017-11-07 22:03:34 is 5287383.495642829 s in the futur
e
tar: bin: time stamp 2017-11-07 22:03:34 is 5287383.495573245 s in the future
Now using node v8.9.1 (npm v5.5.1)
Creating default alias: default -> 8.9.1 (-> v8.9.1)
pi@raspi-2023-2:~ $
```

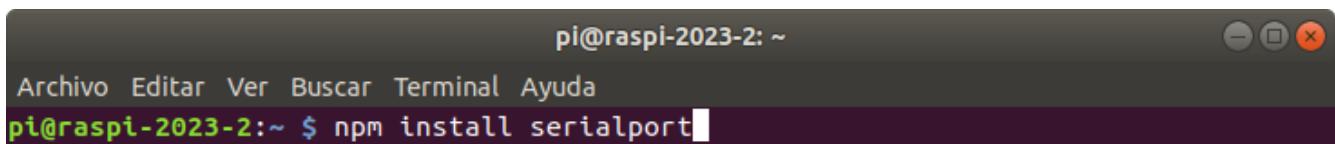
Figura 29. Finalización correcta de la instalación para Node.js.

3.4. Instalación de Serialport

En instalaciones previas se ha encontrado un error asociado a la inexistencia del paquete *serialport* empleado originalmente, por lo tanto, se descarga una versión más actual y con soporte para la versión de Node.js instalada.

- Ejecutar el siguiente comando (ver Figura 30 y):

```
npm install serialport
```



```
pi@raspi-2023-2: ~
Archivo Editar Ver Buscar Terminal Ayuda
pi@raspi-2023-2:~ $ npm install serialport
```

Figura 30. Comando de instalación para serialport.


```
pi@raspi-2023-2: ~
Archivo Editar Ver Buscar Terminal Ayuda
r.gz
prebuild-install http 404 https://github.com/EmergingTechnologyAdvisors/node-serialport/releases/download/v6.0.4/serialport-v6.0.4-node-v57-linux-arm.tar.gz
prebuild-install WARN install No prebuilt binaries found (target=8.9.1 runtime=node arch=arm platform=linux)
make: Entering directory '/home/pi/node_modules/serialport/build'
  CXX(target) Release/obj.target/serialport/src/serialport.o
  CXX(target) Release/obj.target/serialport/src/serialport_unix.o
  CXX(target) Release/obj.target/serialport/src/poller.o
  SOLINK_MODULE(target) Release/obj.target/serialport.node
  COPY Release/serialport.node
make: Leaving directory '/home/pi/node_modules/serialport/build'
npm WARN saveError ENOENT: no such file or directory, open '/home/pi/package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open '/home/pi/package.json'
npm WARN pi No description
npm WARN pi No repository field.
npm WARN pi No README data
npm WARN pi No license field.

+ serialport@6.0.4
added 58 packages in 31.87s
pi@raspi-2023-2:~ $
```

Figura 31. Finalización correcta de la instalación para serialport.

3.5. Instalación de MongoDB

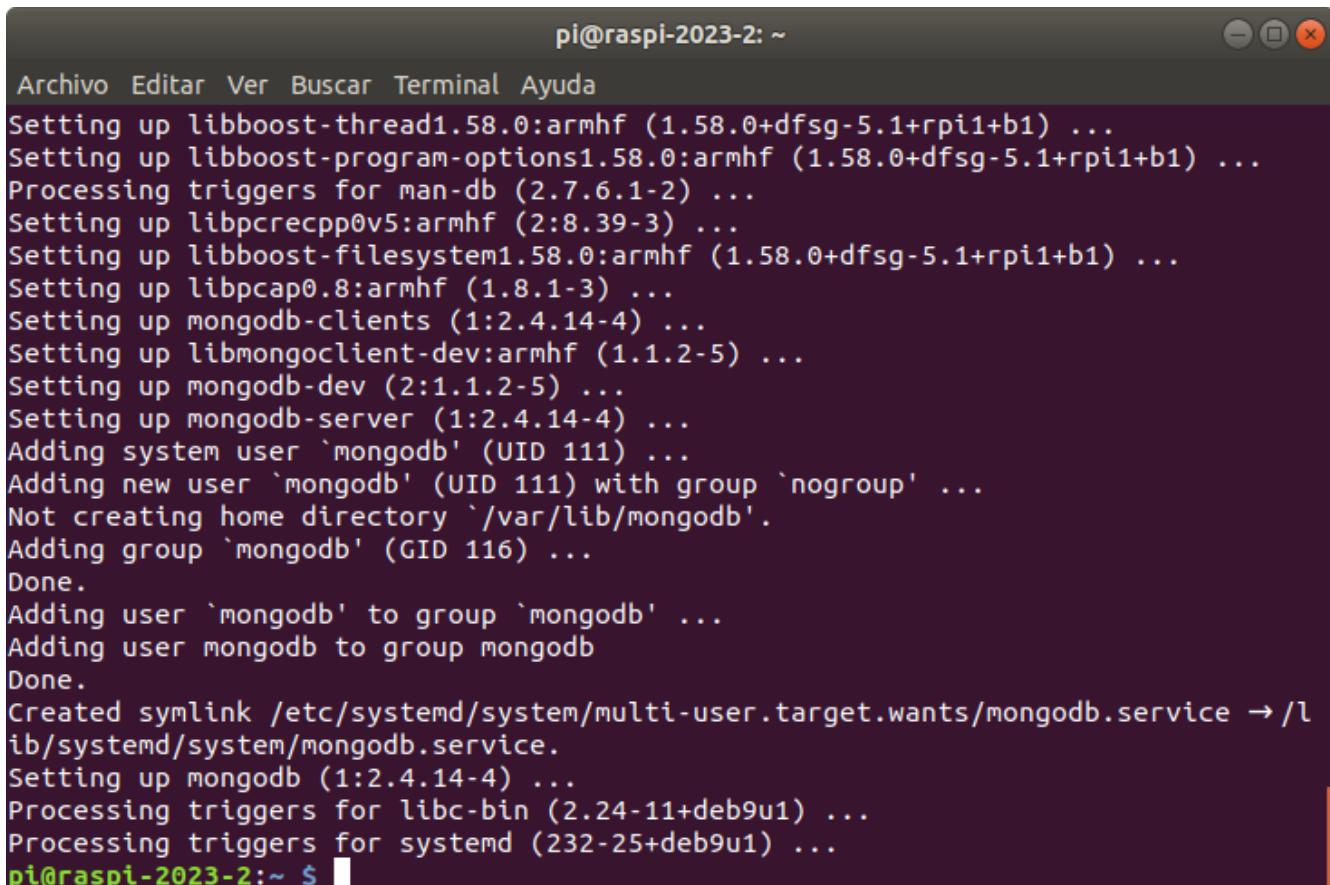
Para que el sistema esté en la capacidad de almacenar los datos entregados por el sistema Arduino compatible, se debe instalar una versión de MongoDB, la cual se encarga de almacenar dichos datos.

- ejecute el siguiente comando (ver Figura 32 y Figura 33):

```
sudo apt-get install mongodb
```

```
pi@raspi-2023-2: ~
Archivo Editar Ver Buscar Terminal Ayuda
pi@raspi-2023-2:~ $ sudo apt-get install mongodb
```

Figura 32. Comando de instalación de MongoDB.



```
pi@raspi-2023-2: ~
Archivo Editar Ver Buscar Terminal Ayuda
Setting up libboost-thread1.58.0:armhf (1.58.0+dfsg-5.1+rpi1+b1) ...
Setting up libboost-program-options1.58.0:armhf (1.58.0+dfsg-5.1+rpi1+b1) ...
Processing triggers for man-db (2.7.6.1-2) ...
Setting up libpcrecpp0v5:armhf (2:8.39-3) ...
Setting up libboost filesystem1.58.0:armhf (1.58.0+dfsg-5.1+rpi1+b1) ...
Setting up libpcap0.8:armhf (1.8.1-3) ...
Setting up mongodb-clients (1:2.4.14-4) ...
Setting up libmongoclient-dev:armhf (1.1.2-5) ...
Setting up mongodb-dev (2:1.1.2-5) ...
Setting up mongodb-server (1:2.4.14-4) ...
Adding system user `mongodb' (UID 111) ...
Adding new user `mongodb' (UID 111) with group `nogroup' ...
Not creating home directory `/var/lib/mongodb'.
Adding group `mongodb' (GID 116) ...
Done.
Adding user `mongodb' to group `mongodb' ...
Adding user mongodb to group mongodb
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/mongodb.service → /lib/systemd/system/mongodb.service.
Setting up mongodb (1:2.4.14-4) ...
Processing triggers for libc-bin (2.24-11+deb9u1) ...
Processing triggers for systemd (232-25+deb9u1) ...
pi@raspi-2023-2:~ $
```

Figura 33. Finalización correcta de la instalación para MongoDB.

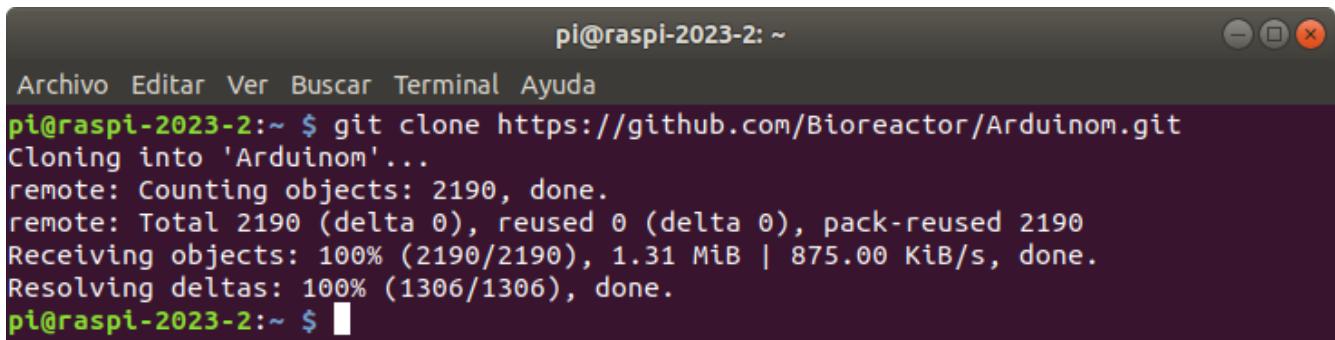
4. Arduinom

Ahora que ya se dispone de todos los elementos necesarios para instalar el proyecto, es momento de clonarlo desde su repositorio en Github y ejecutar los comandos necesarios para su instalación y funcionamiento en el servidor Node.js.

4.2. Clonar el repositorio

- Para esto, se debe ejecutar el siguiente comando (ver Figura 34):

```
git clone https://github.com/Bioreactor/Arduinom.git
```



```
pi@raspi-2023-2: ~
Archivo Editar Ver Buscar Terminal Ayuda
pi@raspi-2023-2:~ $ git clone https://github.com/Bioreactor/Arduinom.git
Cloning into 'Arduinom'...
remote: Counting objects: 2190, done.
remote: Total 2190 (delta 0), reused 0 (delta 0), pack-reused 2190
Receiving objects: 100% (2190/2190), 1.31 MiB | 875.00 KiB/s, done.
Resolving deltas: 100% (1306/1306), done.
pi@raspi-2023-2:~ $
```

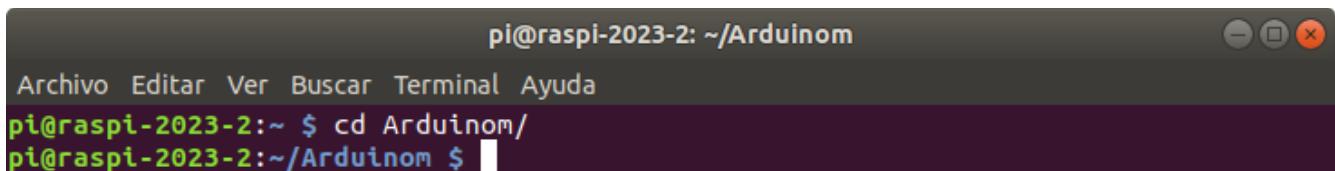
Figura 34. Clonación del repositorio Arduinom en Github.

4.2. Instalar Arduinom

4.2.1. Acceder a la carpeta del proyecto

- Ejecutar el comando (ver Figura 35):

```
cd Arduinom/
```

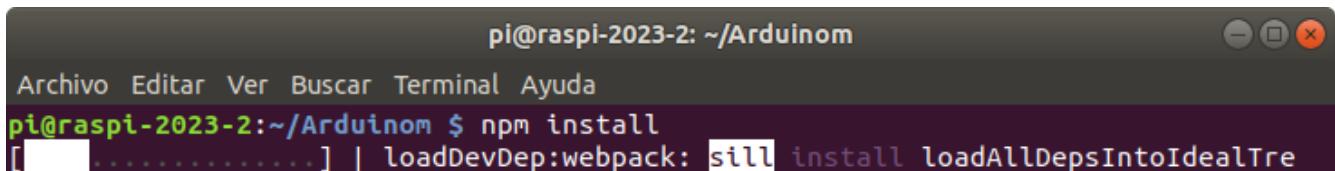


```
pi@raspi-2023-2: ~/Arduinom
Archivo Editar Ver Buscar Terminal Ayuda
pi@raspi-2023-2:~ $ cd Arduinom/
pi@raspi-2023-2:~/Arduinom $
```

Figura 35. Acceso a la carpeta del proyecto Arduinom.

- Para instalar el proyecto, se debe ejecutar dentro de la carpeta el siguiente comando (ver Figura 36 y Figura 37):

```
npm install
```



```
pi@raspi-2023-2: ~/Arduinom
Archivo Editar Ver Buscar Terminal Ayuda
pi@raspi-2023-2:~/Arduinom $ npm install
[██████████] | loadDevDep:webpack: sill install loadAllDepsIntoIdealTre
```

Figura 36. Instalación del proyecto Arduinom.

```
pi@raspi-2023-2: ~/Arduinom
Archivo Editar Ver Buscar Terminal Ayuda
src/devices/OpenSpectro/deviceInformation.js -> lib/devices/OpenSpectro/deviceIn
formation.js
src/devices/Solar/Solar.js -> lib/devices/Solar/Solar.js
src/devices/Solar/deviceInformation.js -> lib/devices/Solar/deviceInformation.js
src/electron/index.js -> lib/electron/index.js
src/public/app.js -> lib/public/app.js
src/public/socket-io.js -> lib/public/socket-io.js
src/server/routes/db.js -> lib/server/routes/db.js
src/server/server.js -> lib/server/server.js
src/server/socketio/devices.js -> lib/server/socketio/devices.js
src/utilities/IncrementalPoll.js -> lib/utilities/IncrementalPoll.js
src/utilities/parser.js -> lib/utilities/parser.js
src/utilities/util.js -> lib/utilities/util.js
src/view/index.js -> lib/view/index.js
npm WARN arduinom@0.0.0 No repository field.
npm WARN arduinom@0.0.0 scripts['server'] should probably be scripts['start'].
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.2 (node_modules/fse
vents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@
1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"arm"})
)
added 1144 packages in 320.438s
pi@raspi-2023-2:~/Arduinom $
```

Figura 37. Finalización correcta de la instalación para Arduinom.

4.3. Ejecución del servidor Node.js

Finalmente, para observar si el proyecto a compilado correctamente, se ejecuta el comando:

npm run server (ver Figura 38)

```
pi@raspi-2023-1: ~/Arduinom
Archivo Editar Ver Buscar Terminal Ayuda
src/server/server.js -> lib/server/server.js
src/server/socketio/devices.js -> lib/server/socketio/devices.js
src/utilities/IncrementalPoll.js -> lib/utilities/IncrementalPoll.js
src/utilities/parser.js -> lib/utilities/parser.js
src/utilities/util.js -> lib/utilities/util.js
src/view/index.js -> lib/view/index.js
npm WARN arduinom@0.0.0 No repository field.
npm WARN arduinom@0.0.0 scripts['server'] should probably be scripts['start'].
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.2 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"arm"})
)
added 1144 packages in 324.219s
pi@raspi-2023-1:~/Arduinom $ npm run server

> arduinom@0.0.0 server /home/pi/Arduinom
> node src/server/server.js --sync-db

koa deprecated Support for generators will be removed in v3. See the documentation for examples of how to convert old middleware https://github.com/koajs/koa/blob/master/docs/migration.md src/server/server.js:17:5
```

Figura 38. Ejecución del servidor Node.js.

Y se observa a través de un navegador de internet, si existe alguna página albergada en el puerto 3000.

- Si se invocó el comando -Y al emplear SSH, ejecute (ver Figura 39):

chromium-google

```
pi@raspi-2023-2: ~/Arduinom
Archivo Editar Ver Buscar Terminal Ayuda
pi@raspi-2023-2:~/Arduinom $ chromium-browser
Gtk-Message: Failed to load module "canberra-gtk-module"
Gtk-Message: Failed to load module "canberra-gtk-module"
[11015:11015:1127/222649.979170:ERROR:shared_image_x11.cc(129)] SHM attach failed
```

Figura 39. Ejecución del navegador web mediante ssh -Y.

- En caso contrario, abra una nueva terminal y ejecute (ver Figura 40):

ssh -X pi@192.168.0.12 chromium-browser

andres@ACV-03: ~

Archivo Editar Ver Buscar Terminal Ayuda

andres@ACV-03:~\$ ssh -X pi@192.168.0.12 chromium-browser

Figura 40. Ejecución del navegador web mediante ssh -X.

- Ahora, en la barra de navegación, ingrese la siguiente dirección (ver Figura 41):

localhost:3000

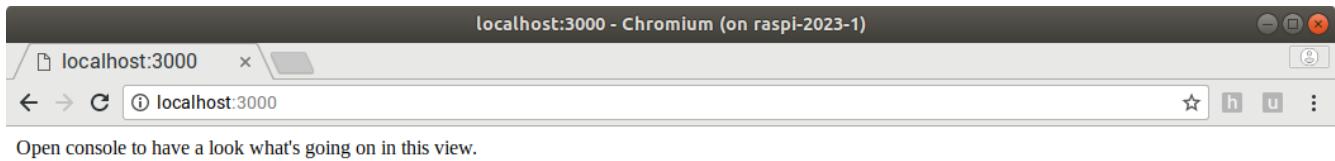


Figura 41. Acceso correcto a la página http://localhost:3000.

- Para conocer los datos que se deben recibir del equipo, se accede directamente al enlace *general*, el cual despliega la interfaz presente en la Figura 42:

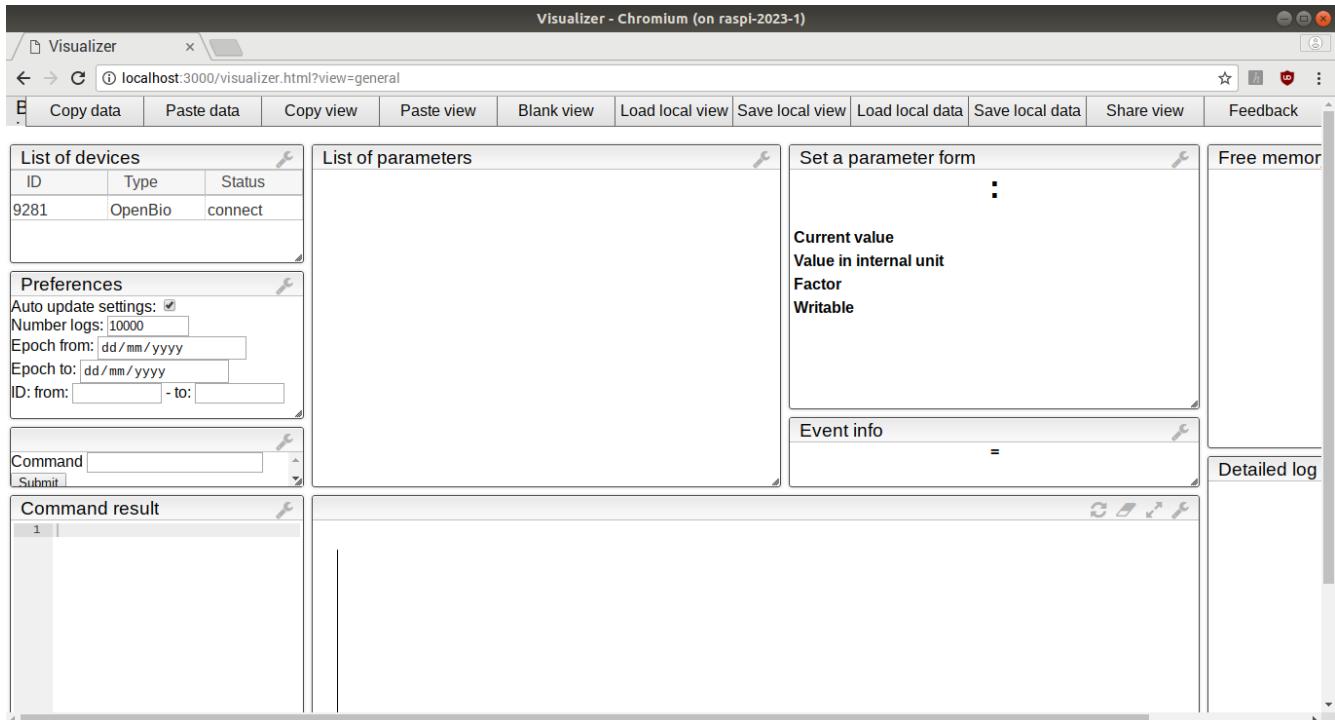


Figura 42. Interfaz de usuario del servidor sin conexión de dispositivos.

- Al momento de conectar el equipo Bioreactor a la Raspberry Pi y refrescar la página, se puede ver el *ID* del dispositivo y con este, observar las gráficas de los datos que entrega, dicha información se puede apreciar en la Figura 43:

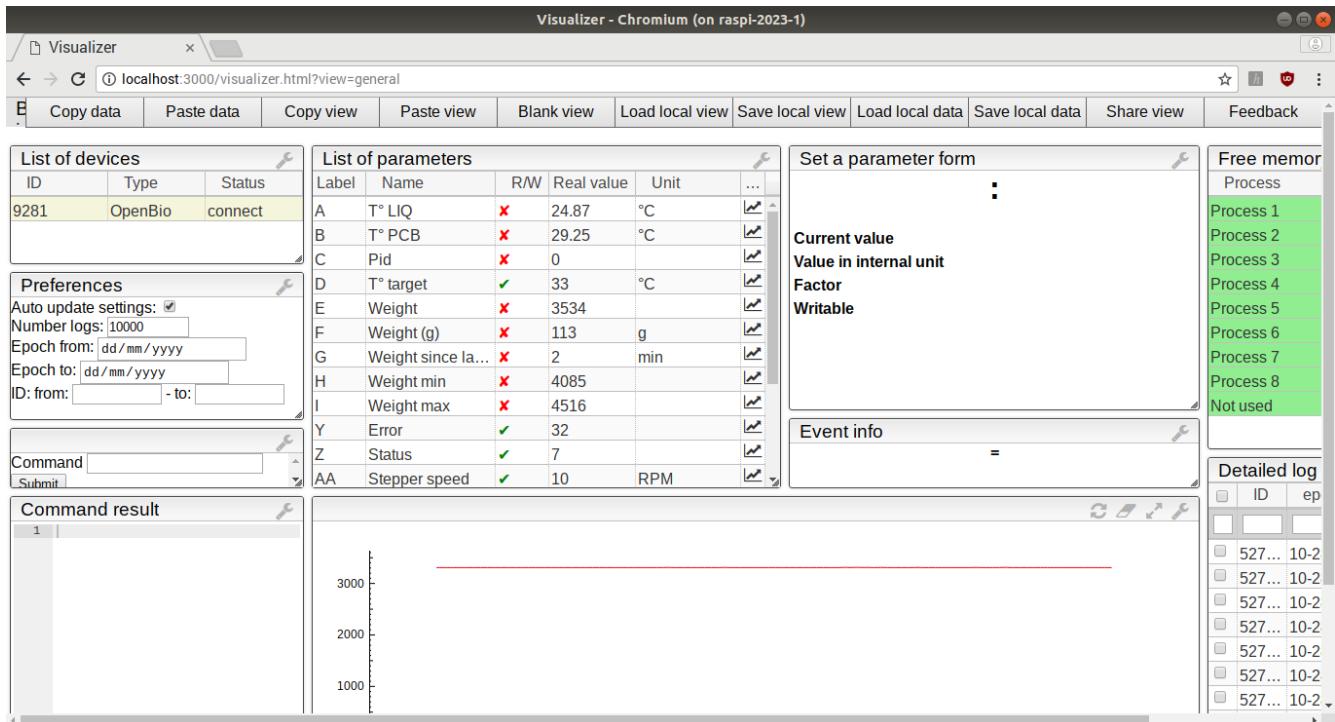


Figura 43. Interfaz de usuario del servidor con dispositivos conectados.

5. Problemas frecuentes

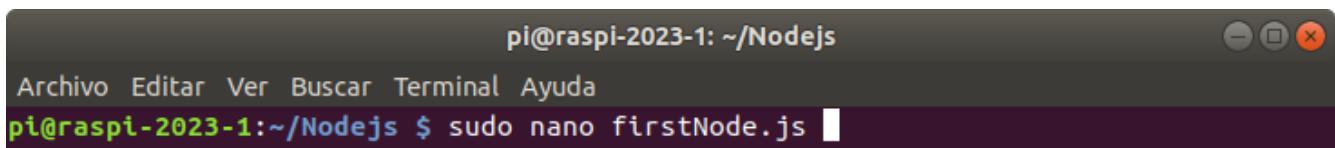
- La versión del Node.js es inferior a la requerida para el correcto funcionamiento del proyecto.
 - Esto sucede cuando se instala la versión del Node.js empleando el comando:
 - sudo apt-get install node
 - Para solucionar dicho problema se debe, primero, desinstalar la versión actual:
 - sudo apt-get unistall node-v59
 - Luego, se debe proceder a instalar NVM (sección 3.2) y finalmente Node.js (sección 3.3).
- Tiempo excesivo en instalar NVM.
 - Este error puede ser asociado a una fecha mal configurada, para resolver esto, cancele la instalación presionando *Ctrl+c*, posterior a esto, configura la fecha y hora actuales (ver sección 3.3.1).
- *MongoError: failed to connect to server [localhost:27017] on first connect [MongoError: connect ECONNREFUSED 127.0.0.1:27017]*.

- Este error puede deberse a que MongoDB no se encuentra instalado y/o la versión instalada no es actual. Para corregir el error, instale MongoDB mediante el comando:
 - `sudo apt-get install mongodb`.

6. Anexo A: Prueba del servidor Node.js previo a la instalación de Arduinom.

Como ya se tiene instalado el servidor Node.js sobre Raspbian, el siguiente paso es desarrollar un pequeño código que podamos ver desde el computador con Ubuntu.

- Primero, se ubica en la carpeta donde se desea almacenar el documento y se escribe `sudo nano firstNode.js` (ver Figura 44).



```
pi@raspi-2023-1:~/Nodejs $ sudo nano firstNode.js
```

Figura 44. Creación del archivo de prueba `firstNode.js`.

- Posteriormente, dentro del archivo se escribe el siguiente código (ver Figura 45):

```
var http = require('http');

http.createServer(function (req, res) {
  res.write('Hello World!');
  res.end();
}).listen(8080);
```

pi@raspi-2023-1: ~/Nodejs

Archivo Editar Ver Buscar Terminal Ayuda

GNU nano 2.7.4 File: firstNode.js

```
var http = require('http');

http.createServer(function (req, res) {
    res.write('Hello World!');
    res.end();
}).listen(8080);
```

[Read 6 lines]

^G Get Help **^O** Write Out **^W** Where Is **^K** Cut Text **^J** Justify **^C** Cur Pos
^X Exit **^R** Read File **^V** Replace **^U** Uncut Text **^T** To Spell **^** Go To Line

Figura 45. Código del archivo *firstNode.js*.

- Al guardar el archivo, ya es posible invocar a Node.js para que ejecute el archivo. La ejecución se realiza mediante el comando *node firstNode.js* y desde el computador con Ubuntu u otro equipo conectado a la red es posible acceder al código escribiendo en un navegador web *192.168.0.10:8080* (ver Figura 46).



Figura 46. Verificación del funcionamiento del servidor Node.js.