

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN

MANUAL DEL ADMINISTRADOR

“Sistema de Telemetría de la producción de leche y control del ordeño mecánico del ganado bovino, basado en plataformas de hardware y software libre, en la Granja Experimental La Pradera de la Universidad Técnica del Norte.”

AUTOR: OLAVO GABRIEL PAVÓN IPIALES

DIRECTOR: ING. OMAR OÑA

IBARRA-ECUADOR

2016

CONTENIDO

1.	Formateo de la tarjeta SD.....	8
2.	Instalación del sistema operativo.	9
3.	Configuración de la tarjeta de red.	11
3.1	Configuración de una Dirección Ip Estática.....	12
4.	Instalación del servidor Web y Php.....	14
4.1	Actualización de los repositorios.....	15
4.2	Instalación del servidor web Apache y PHP.	15
4.3	Comprobación del funcionamiento del servidor web apache.....	16
4.4	Configuración de permisos al grupo de apache.....	16
4.5	Autenticación de usuarios en Apache con Digest	18
5.	Instalación de MySQL y phpMyAdmin.....	21
5.1	Instalación de MySQL y phpMyAdmin	21
5.2	Configuración de contraseñas de los paquetes instalados.	21
5.3	Comprobación del funcionamiento de la base de datos MySQL.	25
5.4	Creación de la base de datos del sistema.....	26
5.5	Creación de usuario y contraseña.	27
6.	Instalación de Python	28
6.1	Instalación del módulo python-rpi.gpio.	28
6.2	Instalación del módulo Python-serial.	28
6.3	Instalación del módulo python-mysqldb.	29
7.	El Servidor FTP.....	29
7.1	Instalación del servidor FTP.....	30
7.2	Configuración del servidor FTP.	30
7.3	Verificación del puerto del servidor FTP.	33
7.4	Instalación y comprobación del cliente FTP.	33

7.5	Agregando un nuevo usuario al servidor FTP.....	35
8.	VNC (Virtual Network Computing).	38
8.1	Instalación del servidor VNC.	38
8.2	Ejecución del servidor TightVNC.....	39
8.3	Inicialización en el sistema.....	39
8.4	Automatización y ejecución en el arranque del sistema.....	41
8.5	Instalación y comprobación del cliente VNC.....	41
9.	MODULO RTC (Real Time Clock).....	43
9.1	Activación de I2C en la Raspberry Pi B+.	44
9.2	Configuración de la zona horaria.....	46
9.3	Edición de los módulos.	49
9.4	Edición de la lista negra del Raspberry Pi B+.	49
9.5	Instalación de utilidades.	50
9.6	Comprobación de I2C y RTC estén habilitados.	50
9.7	Pruebas de hardware.....	51
9.8	Carga de los módulos al kernel.	51
9.9	Configuración del dispositivo I2C.....	52
10.	Apagado de todo el sistema.....	54
10.1	Creación del script de apagado del sistema en Python.....	54
10.2	Automatización del script de apagado.....	56

ÍNDICE DE FIGURAS

Figura 1 Selección de la Unidad de la tarjeta SD	8
Figura 2 Confirmación de formateo de la tarjeta SD.....	8
Figura 3 Mensaje de advertencia	9
Figura 4 Fin de formateo de la tarjeta SD.	9
Figura 5 Ejecución del software WIN32 Disk Imager	10
Figura 6 Selección del sistema operativo	10
Figura 7 Mensaje de alerta.....	11
Figura 8 Mensaje de finalización de escritura correcta	11
Figura 9 Archivo de configuración de las tarjetas de red	12
Figura 10 Asignación de parámetros para configurar la tarjeta de red.....	12
Figura 11 Fichero de configuración del DNS.....	13
Figura 12 Asignación de la dirección Ip del DNS.....	13
Figura 13 Reinicio del servicio de configuraciones de la tarjeta de red	13
Figura 14 Verificación de las configuraciones de la tarjeta de red	14
Figura 15 Ejecución de un ping	14
Figura 16 Actualización de los repositorios para Raspbian	15
Figura 17 Instalación del servidor web apache y los respectivos módulos de Php	15
Figura 18 Instalación de los módulos de Python.....	16
Figura 19 Verificación del servicio de apache	16
Figura 20 Comprobación del funcionamiento del servidor web apache	16
Figura 21 Cambio de usuario y grupo al directorio /var/www	17
Figura 22 Ejecución de permisos de lectura, escrita y ejecución en carpeta.....	17
Figura 23 Añadiendo usuario al grupo	18
Figura 24 Activación de módulo Digest.....	18
Figura 25 Edición de fichero de configuración de apache	19
Figura 26 Creación de usuario y contraseña para el servidor web apache	19
Figura 27 Verificación de privacidad en el servidor web.....	20
Figura 28 Colocación de nombre de usuario y su respectiva contraseña	20
Figura 29 Comprobación de usuario que ingreso correctamente al servidor web.....	21
Figura 30 Instalación de MySQL, phpMyAdmin y sus respectivos módulos.....	21
Figura 31 Colocación del password al usuario root de MySQL.....	22
Figura 32 Confirmación del password	22

Figura 33 Elección del servidor web instalado.....	22
Figura 34 Configuración de la base de datos para phpMyAdmin	23
Figura 35 Ingreso del password de la base de datos del usuario administrativo	23
Figura 36 Ingreso de password para phpMyAdmin	23
Figura 37 Confirmación del password y fin de la instalación	24
Figura 38 Comando para crear enlace simbólico	24
Figura 39 Ingreso a phpMyAdmin	25
Figura 40 Comando para ingresar a MySQL.....	25
Figura 41 Bases de datos por defecto	26
Figura 42 Crear base de datos denominada Ganado.....	26
Figura 43 Comprobación de la base de datos Ganado.....	27
Figura 44 Usuario y contraseña nueva creada	27
Figura 45 Otorgamiento de permisos	27
Figura 46 Instalación del módulo Python-rpi.gpio	28
Figura 47 Instalación del módulo Python-serial.....	29
Figura 48 Instalación del módulo python-mysqldb.....	29
Figura 49 Instalación del servidor vsftpd	30
Figura 50 Archivo de configuración del servidor vsftpd.....	30
Figura 51 Edición del archivo de configuración vsftpd para escritura.....	31
Figura 52 Banner de bienvenida.....	31
Figura 53 Edición del archivo de configuración vsftpd para usuario anonymous	32
Figura 54 Reinicio del servicio vsftpd.....	32
Figura 55 Verificación del puerto del servidor vsftpd.....	33
Figura 56 Ejecución del software WinSCP	33
Figura 57 Ingreso correcto del usuario anonymous	34
Figura 58 Ingreso al servidor vsftpd con el usuario pi	34
Figura 59 Ingreso correcto al usuario pi.....	35
Figura 60 Creando un nuevo usuario al servidor vsftpd.....	35
Figura 61 Ingreso al servidor vsftpd con el nuevo usuario.....	36
Figura 62 Ingreso correcto a la carpeta del nuevo usuario	36
Figura 63 Transferencia de archivo hacia el nuevo usuario (pradera)	37
Figura 64 Comprobación de que el archivo se transfirió exitosamente hacia el usuario pradera	37

Figura 65 Otorgando permisos de lectura, escritura y ejecución a todo el contenido del usuario pradera	38
Figura 66 Instalación del servidor VNC.....	38
Figura 67 Ejecución del servidor TightVNC.....	39
Figura 68 Creación de fichero vncboot	39
Figura 69 Parámetros para iniciar y parar el servicio VNC	40
Figura 70 Iniciar el servicio VNC.	40
Figura 71 Parar el servicio VNC	40
Figura 72 Automatización y ejecución del fichero VNC en el arranque del sistema.....	41
Figura 73 Reinicio del sistema	41
Figura 74 Ejecución del cliente VNC.....	42
Figura 75 Mensaje de VNC Viewer	42
Figura 76 Colocación de la contraseña para ingresar por VNC	43
Figura 77 Comprobación del ingreso correcto por VNC a la Raspberry Pi B+	43
Figura 78 Comando para realizar configuraciones del sistema.....	44
Figura 79 Opciones de configuraciones del sistema	44
Figura 80 Selección de opciones avanzadas.....	45
Figura 81 Selección de configuración de la comunicación I2C	45
Figura 82 Modulo I2C cargando por defecto	45
Figura 83 Modulo I2C cargado correctamente por defecto.....	46
Figura 84 Comprobación del módulo I2C este activado	46
Figura 85 Selección de opciones internacionales	47
Figura 86 Cambiamos la zona horaria	47
Figura 87 Seleccionamos el área geográfica	47
Figura 88 Seleccionamos la región más cercana a la que nos encontremos.....	48
Figura 89 Final de configuración de la zona horaria	48
Figura 90 Comprobación de la fecha y hora actualizadas	48
Figura 91 Fichero de configuración de los módulos	49
Figura 92 Agregando los módulos que son necesarios	49
Figura 93 Fichero de configuración de las listas negras del Raspberry.	49
Figura 94 Comentando los módulos SPI e I2C	50
Figura 95 Instalación de utilidades	50
Figura 96 Apagado del sistema	50
Figura 97 Verificación del módulo I2C.....	51

Figura 98 Verificación del módulo RTC.....	51
Figura 99 Dirección del dispositivo RTC.....	51
Figura 100 Carga de los módulos al kernel	52
Figura 101 Añadiendo el dispositivo DS1307 como reloj del sistema.....	52
Figura 102 Comprobación que el dispositivo DS1307 esta como reloj del sistema	53
Figura 103 Comando para verificar la hora del sistema.....	53
Figura 104 Creación del fichero de apagado del sistema	54
Figura 105 Script de apagado del sistema	55
Figura 106 Ejecución manual del script de apagado del sistema	55
Figura 107 Fichero de ejecución automático del script de apagado del sistema.....	56
Figura 108 Agregando el comando para que se ejecute automáticamente el script de apagado.....	56
Figura 109 Comprobación de apagado de todo el sistema	57

1. Formateo de la tarjeta SD.

Se necesita una tarjeta SD mínimo de 4GB de capacidad, se inserta la tarjeta en un PC de Windows y con la ayuda del software SDFormatter. Seguimos los pasos para formatear la tarjeta SD.

Seleccionamos la letra que corresponde a la tarjeta SD, en este caso es la unidad de disco I. Damos clic en Format.

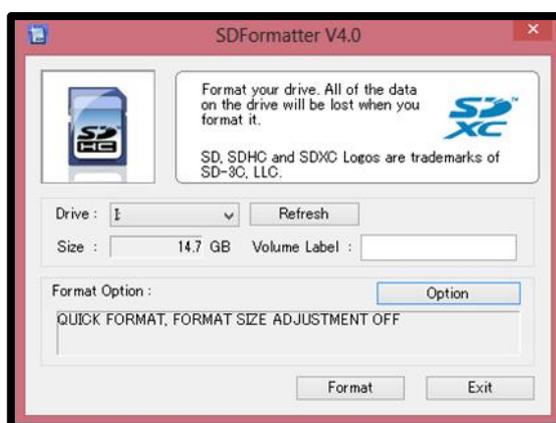


Figura 1 Selección de la Unidad de la tarjeta SD

Fuente: Autor

Seleccionamos la opción Aceptar, para continuar con el formato.

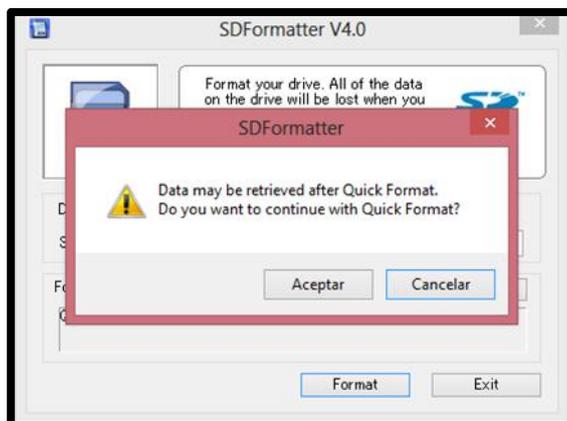


Figura 2 Confirmación de formateo de la tarjeta SD

Fuente: Autor

Nos mostrará un mensaje de advertencia indicando que no debemos expulsar la tarjeta SD durante su formateo. Damos clic en aceptar para continuar.

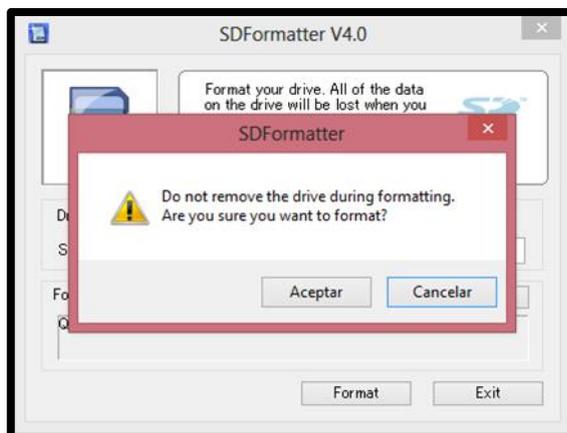


Figura 3 Mensaje de advertencia

Fuente: Autor

Después de unos segundos la tarjeta SD estará formateada y lista para instalar el sistema operativo del Raspberry Pi.

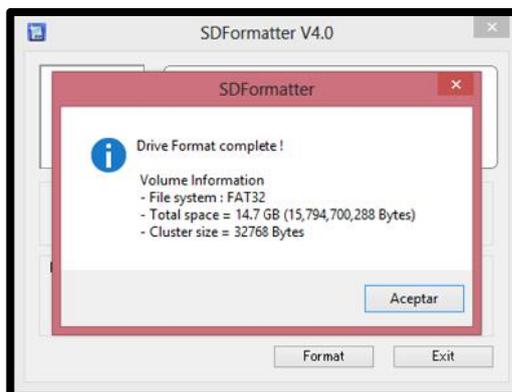


Figura 4 Fin de formateo de la tarjeta SD.

Fuente: Autor

2. Instalación del sistema operativo.

Lo primero que debemos hacer es descargar una imagen ISO de la distribución Raspberry Pi B+, denominado el sistema operativo como Raspbian. Desde el siguiente enlace de la página oficial de raspberry <https://www.raspberrypi.org/downloads/>.

A continuación debemos realizar un proceso que consiste en copiar la ISO a la tarjeta de memoria SD. En Windows podemos hacerlo de forma fácil usando el programa WIN32 Disk Imager. Ejecutamos el programa.

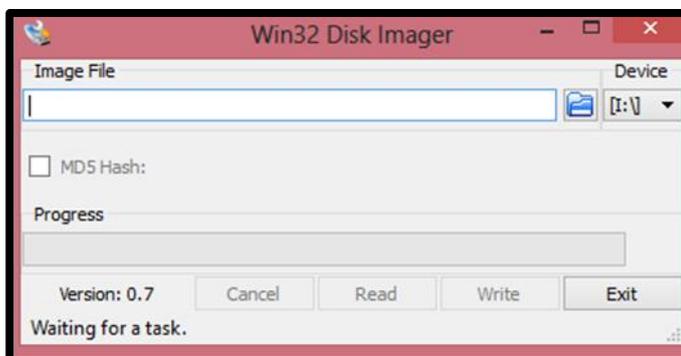


Figura 5 Ejecución del software WIN32 Disk Imager

Fuente: Autor

A continuación debemos realizar un proceso que consiste en copiar la ISO a la tarjeta de memoria SD. En Windows podemos hacerlo de forma fácil usando el programa WIN32 Disk Imager. Ejecutamos el programa.

Seleccionamos la carpeta donde hayamos descargado el sistema operativo. Y luego se habilitará la opción de Write.

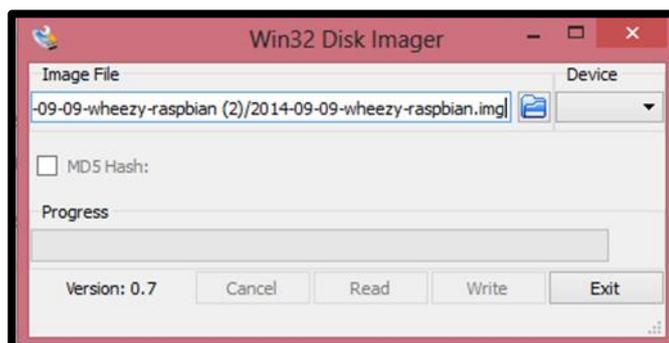


Figura 6 Selección del sistema operativo

Fuente: Autor

Después nos mostrará un mensaje de alerta, para continuar debemos dar clic en aceptar.



Figura 7 Mensaje de alerta

Fuente: Autor

Después de unos minutos nos mostrará un mensaje que se ha escrito correctamente el sistema operativo en la tarjeta SD.

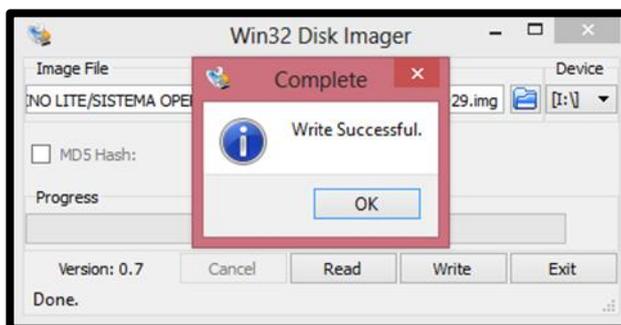


Figura 8 Mensaje de finalización de escritura correcta

Fuente: Autor

Una vez que haya terminado el proceso anterior, simplemente introducimos la tarjeta SD en nuestra Raspberry Pi y la arrancamos.

3. Configuración de la tarjeta de red.

Para establecer conexión con la Raspberry Pi B+ se puede hacer uso del puerto Ethernet que posee conectándole un cable de red o a través de un adaptador inalámbrico WIFI USB.

3.1 Configuración de una Dirección Ip Estática.

Para asignar una dirección IP fija a la Raspberry PI B+, se abre un terminal y se modifica el archivo `/etc/network/interfaces` con el comando `nano` que es un editor de texto.

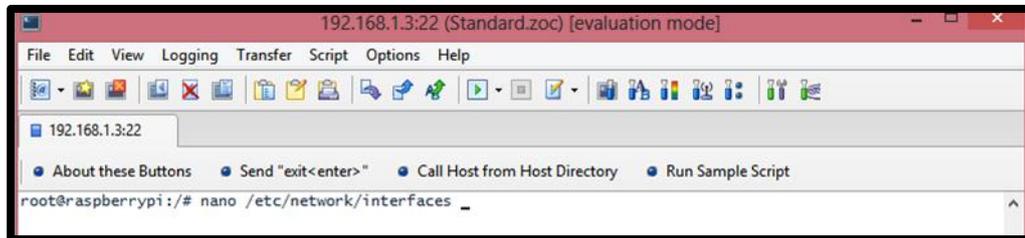


Figura 9 Archivo de configuración de las tarjetas de red

Fuente: Autor

Se coloca los parámetros necesarios como son: la dirección Ip fija, la máscara de red, la dirección Ip de red, el Gateway y la dirección IP de broadcast. Guardamos los cambios realizados.

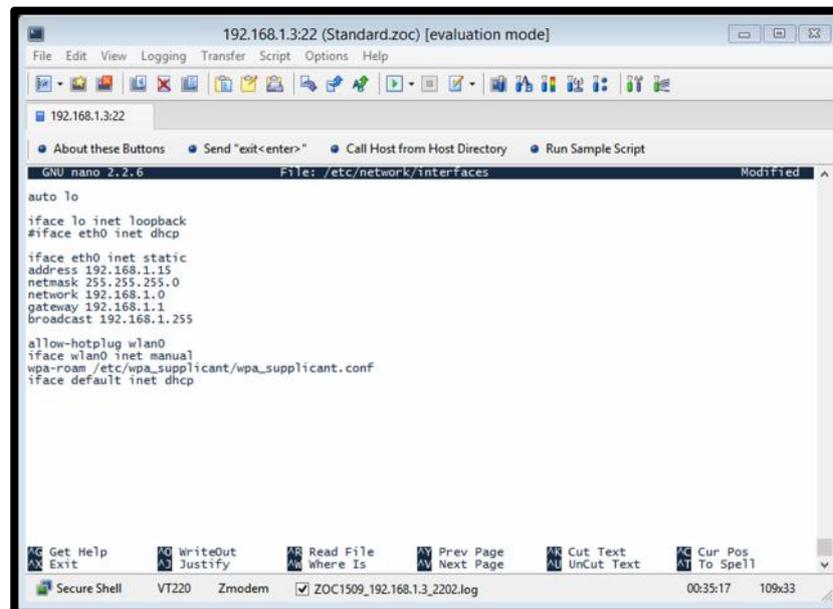


Figura 10 Asignación de parámetros para configurar la tarjeta de red.

Fuente: Autor

Para verificar que se tenga el recurso de Internet en la minicomputadora, se ingresa la dirección IP de un servidor DNS público. Para lo cual es necesario modificar el fichero `/etc/resolv.conf`.

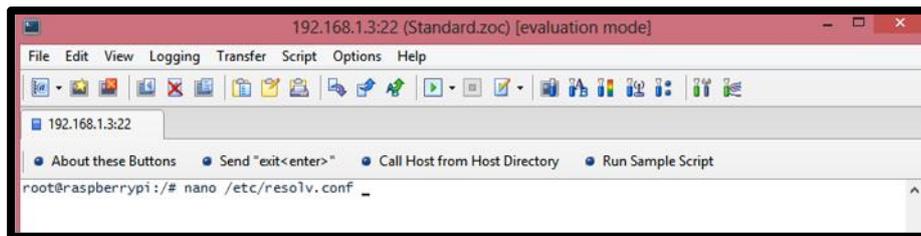


Figura 11 Fichero de configuración del DNS.

Fuente: Autor

Se agrega la siguiente línea nameserver seguido de la dirección Ip del servidor DNS, se guardan los cambios efectuados.

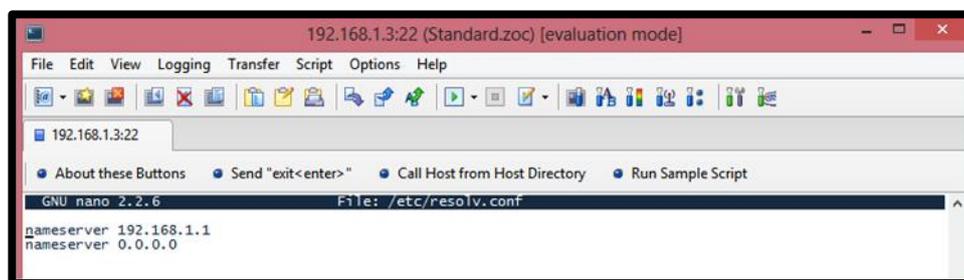


Figura 12 Asignación de la dirección Ip del DNS.

Fuente: Autor

Para efectuar los cambios realizados se debe reiniciar el servicio o demonio de configuración de la tarjeta de red, con el comando /etc/init.d/networking restart o bien reiniciar la raspberry Pi B+.

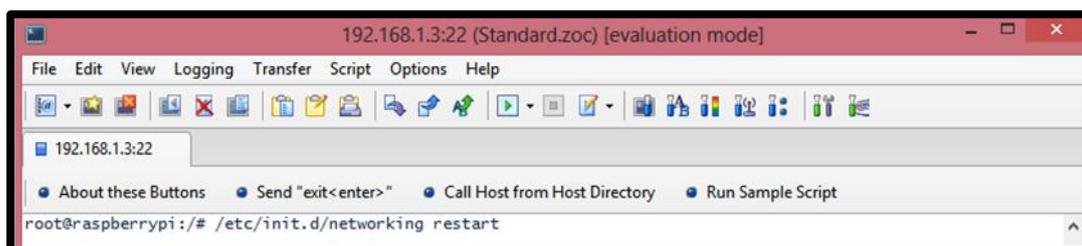


Figura 13 Reinicio del servicio de configuraciones de la tarjeta de red

Fuente: Autor

Verificamos que los cambios se hayan realizado con éxito a través del comando ifconfig.

```

root@raspberrypi:~# ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:3c:99:85
          inet addr:192.168.1.15  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:255  errors:0  dropped:1  overruns:0  frame:0
          TX packets:109  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:19372 (18.9 KiB)  TX bytes:13362 (13.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0  errors:0  dropped:0  overruns:0  frame:0
          TX packets:0  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@raspberrypi:~#

```

Figura 14 Verificación de las configuraciones de la tarjeta de red

Fuente: Autor

Realizamos un ping a una dirección IP pública para verificar que la conexión es exitosa.

```

root@raspberrypi:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_req=1 ttl=51 time=96.2 ms
64 bytes from 8.8.8.8: icmp_req=2 ttl=51 time=96.7 ms
64 bytes from 8.8.8.8: icmp_req=3 ttl=51 time=96.1 ms
64 bytes from 8.8.8.8: icmp_req=4 ttl=51 time=96.2 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 96.150/96.349/96.757/0.449 ms
root@raspberrypi:~# ping www.gogole.com
PING www.gogole.com (216.58.192.67) 56(84) bytes of data:
64 bytes from mia07s34-in-f3.1e100.net (216.58.192.67): icmp_req=1 ttl=53 time=106 ms
64 bytes from mia07s34-in-f3.1e100.net (216.58.192.67): icmp_req=2 ttl=53 time=94.7 ms
64 bytes from mia07s34-in-f3.1e100.net (216.58.192.67): icmp_req=3 ttl=53 time=88.8 ms
64 bytes from mia07s34-in-f3.1e100.net (216.58.192.67): icmp_req=4 ttl=53 time=94.2 ms
^C
--- www.gogole.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 88.880/96.027/106.271/6.347 ms
root@raspberrypi:~# _

```

Figura 15 Ejecución de un ping

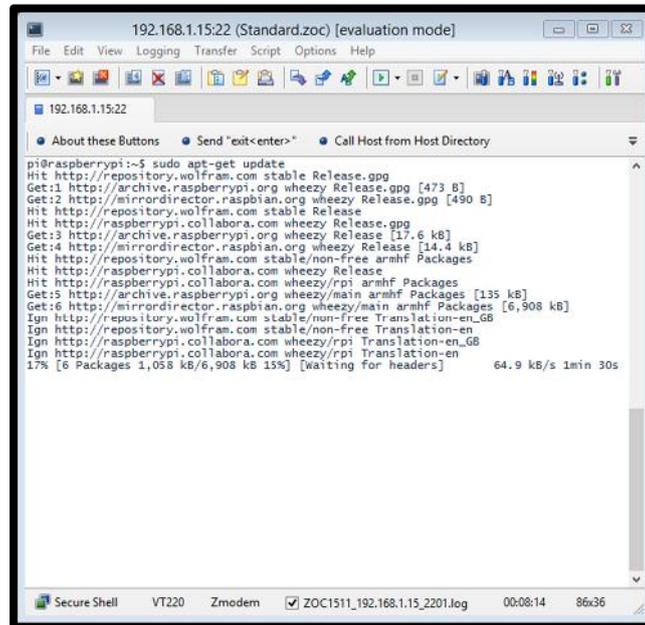
Fuente: Autor

4. Instalación del servidor Web y Php.

Instalación y configuración del servidor LAMP. Usando el software Zoc nos conectamos remotamente por SSH al Raspberry PI B+.

4.1 Actualización de los repositorios.

En la consola del terminal escribimos el siguiente comando `sudo apt-get update`, que nos permita actualizar la lista de los repositorios de raspbian.



```

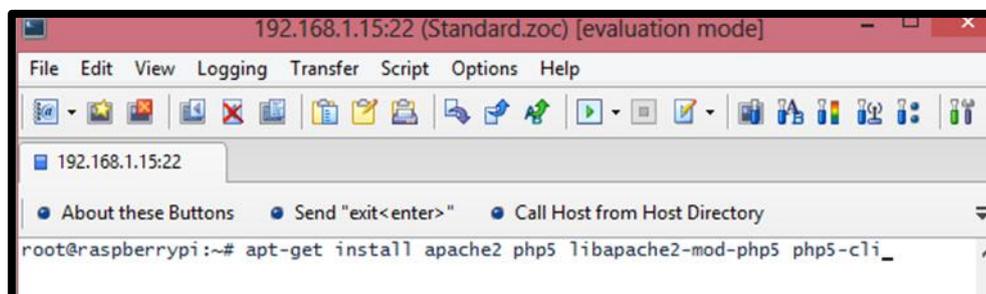
pi@raspberrypi:~$ sudo apt-get update
Hit http://repository.wolfram.com stable Release.gpg
Get:1 http://archive.raspberrypi.org wheezy Release.gpg [473 B]
Get:2 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]
Hit http://repository.wolfram.com stable Release
Hit http://repository.wolfram.com stable Release.gpg
Get:3 http://archive.raspberrypi.org wheezy Release [17.6 kB]
Get:4 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
Hit http://repository.wolfram.com stable/non-free armhf Packages
Hit http://raspberrypi.collabora.com wheezy Release
Hit http://raspberrypi.collabora.com wheezy/rpi armhf Packages
Get:5 http://archive.raspberrypi.org wheezy/main armhf Packages [135 kB]
Get:6 http://mirrordirector.raspbian.org wheezy/main armhf Packages [6,908 kB]
Ign http://repository.wolfram.com stable/non-free Translation-en_GB
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en_GB
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en
17% [6 Packages 1,058 kB/6,908 kB 15%] [waiting for headers] 64.9 kB/s 1min 30s
  
```

Figura 16 Actualización de los repositorios para Raspbian

Fuente: Autor

4.2 Instalación del servidor web Apache y PHP.

Para poder instalar el servidor web apache y los módulos que permiten integrar tanto PHP como Python, en la consola escribimos el siguiente comando.



```

root@raspberrypi:~# apt-get install apache2 php5 libapache2-mod-php5 php5-cli_
  
```

Figura 17 Instalación del servidor web apache y los respectivos módulos de Php

Fuente: Autor

Además también se instalará el siguiente módulo que nos permite integrar Python.

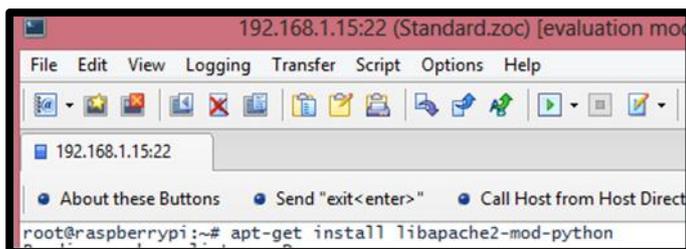


Figura 18 Instalación de los módulos de Python

Fuente: Autor

4.3 Comprobación del funcionamiento del servidor web apache

Reiniciamos el servidor web apache y verificamos el estado del servicio, a través de los siguientes comandos.

```

root@raspberrypi:~# service apache2 restart
Restarting web server: apache2 ... waiting .
root@raspberrypi:~# service apache2 status
Apache2 is running (pid 7638).
root@raspberrypi:~#

```

Figura 19 Verificación del servicio de apache

Fuente: Autor

Comprobamos su funcionamiento colocando nuestra dirección IP en un navegador web.



Figura 20 Comprobación del funcionamiento del servidor web apache

Fuente: Autor

4.4 Configuración de permisos al grupo de apache

Creamos y damos permisos al grupo que usa apache por defecto. Los directorios típicamente utilizados por un servidor web en linux se sitúan en /var/www, y el usuario

típico para dicho entorno suele ser www-data. Ahora vamos a crear el grupo y usuario estándar para nuestro servidor, a la par que otorgamos los permisos pertinentes y añadimos a nuestro usuario por defecto (pi) al grupo comentado. De esta forma no será preciso que el usuario root (su) sea siempre el que pueda operar en /var/www.

Cambiamos el usuario y grupo al directorio.

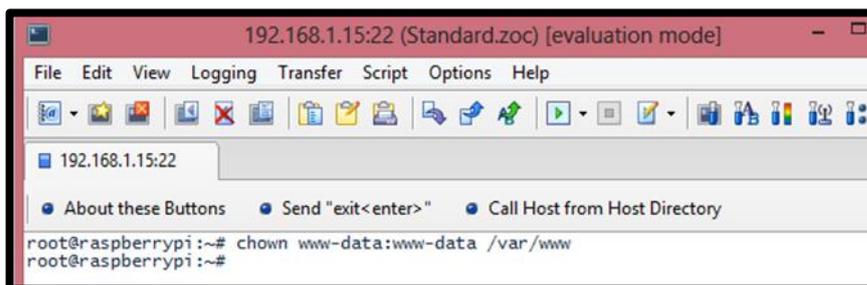


Figura 21 Cambio de usuario y grupo al directorio /var/www

Fuente: Autor

Damos los permisos necesarios tanto de lectura, escritura y ejecución a la carpeta /var/www.

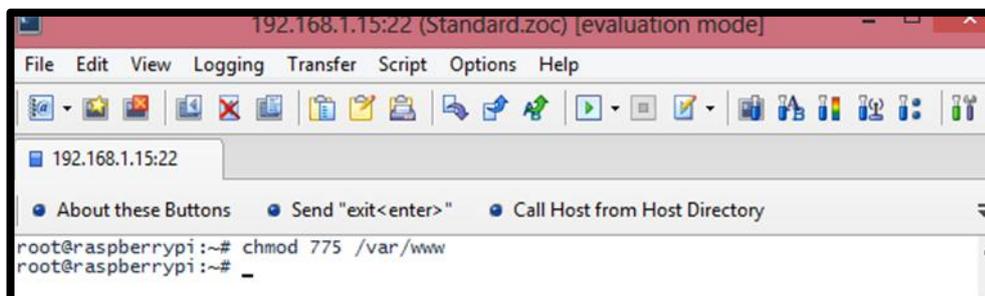
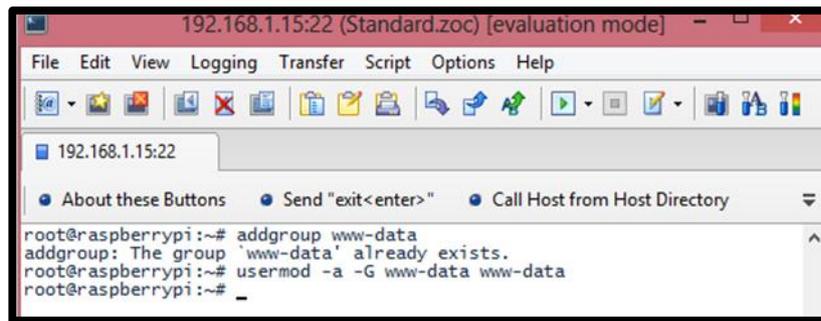


Figura 22 Ejecución de permisos de lectura, escrita y ejecución en carpeta

Fuente: Autor

Añadimos el usuario al grupo www-data.



```

192.168.1.15:22 (Standard.zoc) [evaluation mode]
File Edit View Logging Transfer Script Options Help
192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory
root@raspberrypi:~# addgroup www-data
addgroup: The group 'www-data' already exists.
root@raspberrypi:~# usermod -a -G www-data www-data
root@raspberrypi:~# _

```

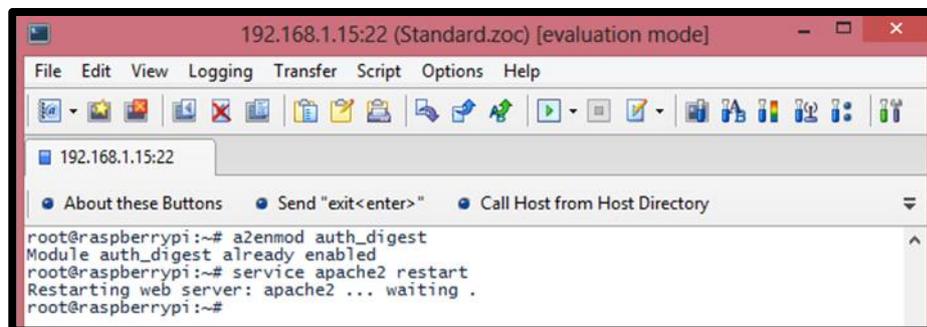
Figura 23 Añadiendo usuario al grupo

Fuente: Autor

Reiniciamos el servidor web apache.

4.5 Autenticación de usuarios en Apache con Digest

Implementamos privacidad en el servidor web a través de un módulo en Apache llamado Digest, el cual viene instalado en Apache. Activamos con el siguiente comando.



```

192.168.1.15:22 (Standard.zoc) [evaluation mode]
File Edit View Logging Transfer Script Options Help
192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory
root@raspberrypi:~# a2enmod auth_digest
Module auth_digest already enabled
root@raspberrypi:~# service apache2 restart
Restarting web server: apache2 ... waiting .
root@raspberrypi:~#

```

Figura 24 Activación de módulo Digest

Fuente: Autor

Editamos el fichero de configuración de Apache agregando las siguientes líneas donde se localiza nuestro VirtualHost, el cual está situado en el directorio `/etc/apache2/sites-available/default`.

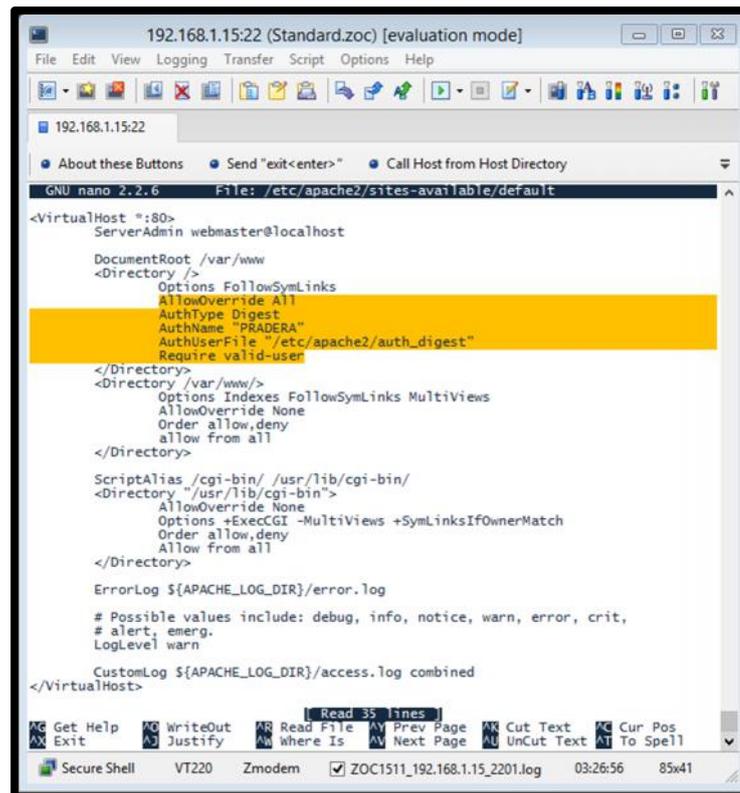


Figura 25 Edición de fichero de configuración de apache

Fuente: Autor

La directiva **AuthType** define el método en autenticación que vamos a usar, en nuestro caso digest. Con **AuthName** se especifica el dominio (o grupo de usuarios) al que pertenecen los usuarios, en mi caso he puesto PRADERA. **AuthUserFile** identifica el directorio donde se almacenan los usuarios y contraseñas. Por último en la directiva **Require** indicamos que necesitamos un usuario válido para mostrar el contenido de la web.

Creamos el usuario con su respectiva contraseña utilizando el comando htdigest.

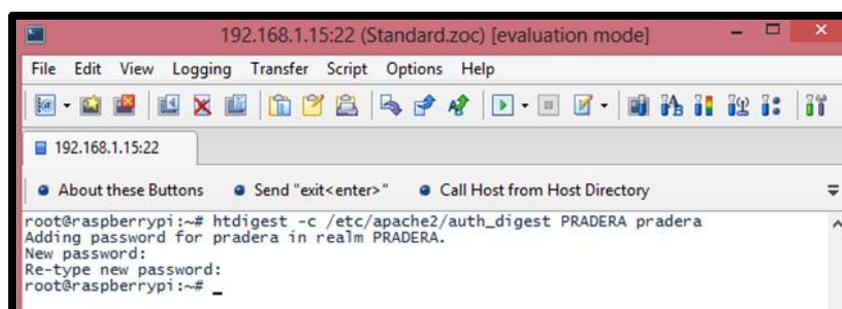


Figura 26 Creación de usuario y contraseña para el servidor web apache

Fuente: Autor

Donde el grupo de usuarios es: PRADERA

El usuario es: **pradera**

Contraseña: **pradera**

Verificamos que nuestro servidor web contenga privacidad.

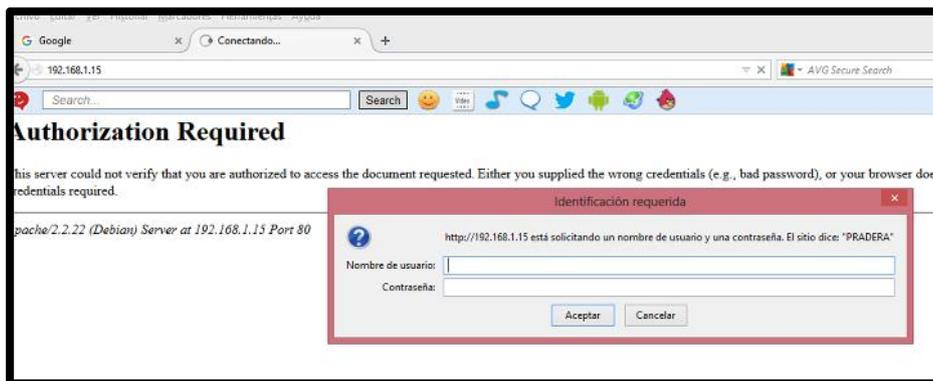


Figura 27 Verificación de privacidad en el servidor web

Fuente: Autor

Colocamos el nombre de usuario con su respectiva contraseña, y damos clic en Aceptar.

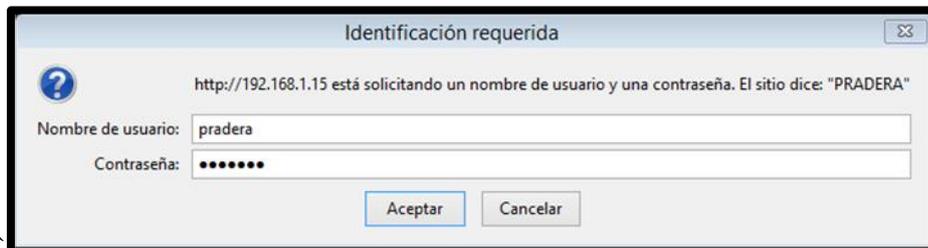


Figura 28 Colocación de nombre de usuario y su respectiva contraseña

Fuente: Autor

Podemos comprobar que se ingresó correctamente.



Figura 29 Comprobación de usuario que ingreso correctamente al servidor web

Fuente: Autor

5. Instalación de MySQL y phpMyAdmin

5.1 Instalación de MySQL y phpMyAdmin

A continuación se instalará el sistema de gestión de Base de Datos y los respectivos módulos que permiten integrar PHP y Python. También se instala phpMyAdmin para administrar la base de datos a través de una interfaz web. En la consola del terminal escribimos lo siguiente.

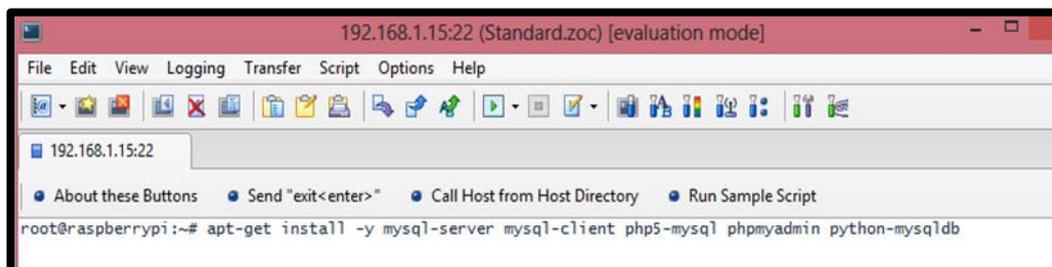


Figura 30 Instalación de MySQL, phpMyAdmin y sus respectivos módulos

Fuente: Autor

5.2 Configuración de contraseñas de los paquetes instalados.

A continuación se instalarán los paquetes y nos pedirá que ingresemos las contraseñas respectivas.

Seleccionamos la opción yes, para configurar la base de datos para phpMyAdmin.

```

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaa Configuring phpmyadmin aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
The phpmyadmin package must have a database installed and configured before it can be used. This can be
optionally handled with dbconfig-common.

If you are an advanced database administrator and know that you want to perform this configuration
manually, or if your database has already been installed and configured, you should refuse this option.
Details on what needs to be done should most likely be provided in /usr/share/doc/phpmyadmin.

Otherwise, you should probably choose this option.

Configure database for phpmyadmin with dbconfig-common?

                                     <Yes>                                     <No>
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

```

Figura 34 Configuración de la base de datos para phpMyAdmin

Fuente: Autor

Ingresamos el password de la base de datos para el usuario administrativo, en este caso hemos escrito el **Password=pradera**.

```

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aa Please provide the password for the administrative account with which this package should create its
aa MySQL database and user.aaaaaa
Password of the database's administrative user:
*****
                                     <Ok>|<>                                     <Cancel>
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

```

Figura 35 Ingreso del password de la base de datos del usuario administrativo

Fuente: Autor

Ingresamos el password para phpMyAdmin, también se ha escrito el **Password=pradera**.

```

aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aa Please provide a password for phpmyadmin to register with the database server. If left blank, a random
a password will be generated.aaaaaa
MySQL application password for phpmyadmin:
*****
                                     <Ok>|<>                                     <Cancel>
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

```

Figura 36 Ingreso de password para phpMyAdmin

Fuente: Autor

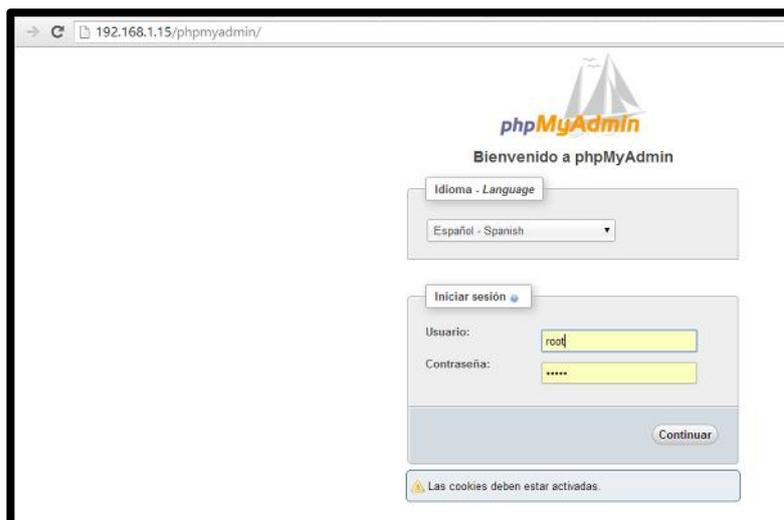


Figura 39 Ingreso a phpMyAdmin

Fuente: Autor

Ingresamos el usuario y la contraseña que ingresamos en la instalación.

Usuario: root

Contraseña: pradera

5.3 Comprobación del funcionamiento de la base de datos MySQL.

En la consola del terminal escribimos el siguiente comando:

```
Root@raspberrypi:~# mysql -u root -p
```

Seguidamente proporcionamos la contraseña que ingresamos en la instalación.

```
root@raspberrypi:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 47
Server version: 5.5.46-0+deb7u1 (Debian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES
```

Figura 40 Comando para ingresar a MySQL

Fuente: Autor

Como se puede observar en el prompt del terminal ha cambiado a **mysql>**, lo que significa que ingresamos correctamente a MySQL.

Con el comando **SHOW DATABASES** se muestra las bases de datos que vienen instaladas por defecto:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
+-----+
4 rows in set (0.02 sec)

mysql> _
```

Figura 41 Bases de datos por defecto

Fuente: Autor

5.4 Creación de la base de datos del sistema.

Con el comando **CREATE DATABASE Ganado;** creamos la base de datos donde van a estar almacenados las tablas de cada animal con su respectiva información.

```
root@raspberrypi:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 5.5.46-0+deb7u1 (Debian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE Ganado;
Query OK, 1 row affected (0.00 sec)

mysql>
```

Figura 42 Crear base de datos denominada Ganado

Fuente: Autor

Comprobamos que la base de datos Ganado se haya creado satisfactoriamente, a través del comando **SHOW DATABASES**.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| Ganado    |
| mysql     |
| performance_schema |
| phpmyadmin |
+-----+
5 rows in set (0.00 sec)

mysql> _
```

Figura 43 Comprobación de la base de datos Ganado

Fuente: Autor

5.5 Creación de usuario y contraseña.

Creamos un usuario con su respectiva contraseña, a través del comando **CREATE USER**.

Donde:

Usuario= pradera

Contraseña= pradera

```
mysql> CREATE USER 'pradera'@'localhost' IDENTIFIED BY 'pradera';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Figura 44 Usuario y contraseña nueva creada

Fuente: Autor

A través del comando **USE Ganado**, estamos utilizando la base de datos que hemos creado en este caso Ganado. Otorgamos los permisos respectivos con el comando **GRANT ALL ON Ganado**.

```
mysql> CREATE USER 'pradera'@'localhost' IDENTIFIED BY 'pradera';
Query OK, 0 rows affected (0.00 sec)

mysql> USE Ganado;
Database changed
mysql> GRANT ALL ON Ganado.* TO 'pradera'@'localhost';
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Figura 45 Otorgamiento de permisos

Fuente: Autor

Cabe recalcar que para la creación de tablas y para insertar los datos de cada tabla se lo realizará con el script de Python.

6. Instalación de Python

Python en la raspberry Pi B+ ya viene instalado por defecto, por lo cual solo debemos instalar los módulos que vayamos a requerir. En caso de no tener instalado escribir el comando **sudo apt-get install python-dev**.

6.1 Instalación del módulo python-rpi.gpio.

Con este módulo podemos acceder a los pines GPIO de la raspberry Pi B+, estos pines pueden ser utilizados como entradas o salidas dependiendo de las necesidades que se tenga. En nuestro caso vamos a utilizar una entrada conectada a un pulsador para poder apagar automáticamente todo el sistema.

Para su instalación escribimos el comando **sudo python-rpi.gpio**.

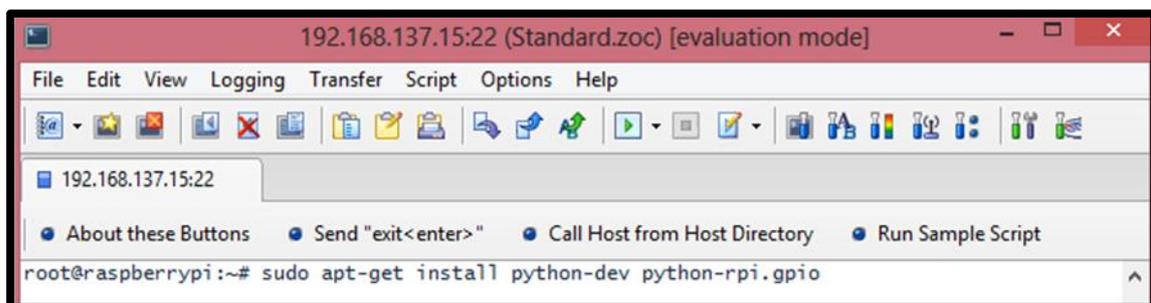


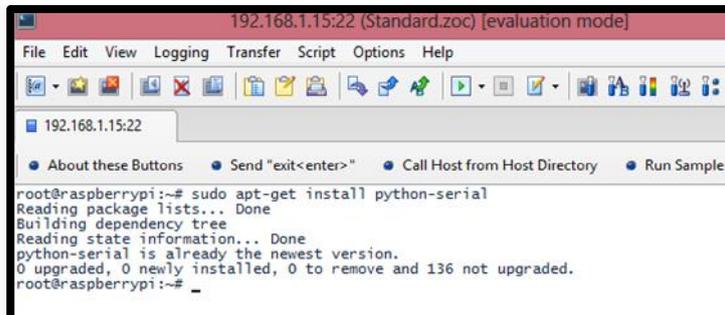
Figura 46 Instalación del módulo Python-rpi.gpio

Fuente: Autor

6.2 Instalación del módulo Python-serial.

Este módulo es muy importante ya que nos permite comunicar serialmente entre el microcontrolador Arduino Mega 2560 y la raspberry Pi B+.

En la consola del terminal escribimos el comando **sudo apt-get install python-serial**.



```

192.168.1.15:22 (Standard.zoc) [evaluation mode]
File Edit View Logging Transfer Script Options Help
192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory Run Sample Scripts
root@raspberrypi:~# sudo apt-get install python-serial
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-serial is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 136 not upgraded.
root@raspberrypi:~# _

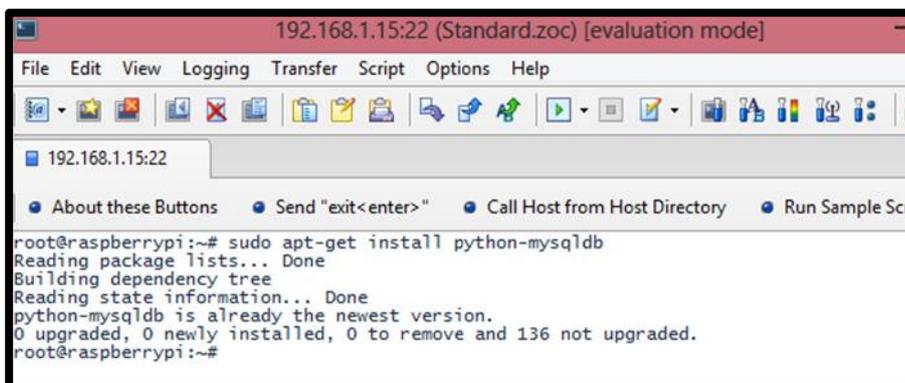
```

Figura 47 Instalación del módulo Python-serial

Fuente: Autor

6.3 Instalación del módulo python-mysqldb.

Este módulo es el que nos permite integrar la base de datos Mysql con Python. Para su instalación escribimos el comando **sudo apt-get install python-mysqldb**.



```

192.168.1.15:22 (Standard.zoc) [evaluation mode]
File Edit View Logging Transfer Script Options Help
192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory Run Sample Scripts
root@raspberrypi:~# sudo apt-get install python-mysqldb
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-mysqldb is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 136 not upgraded.
root@raspberrypi:~#

```

Figura 48 Instalación del módulo python-mysqldb

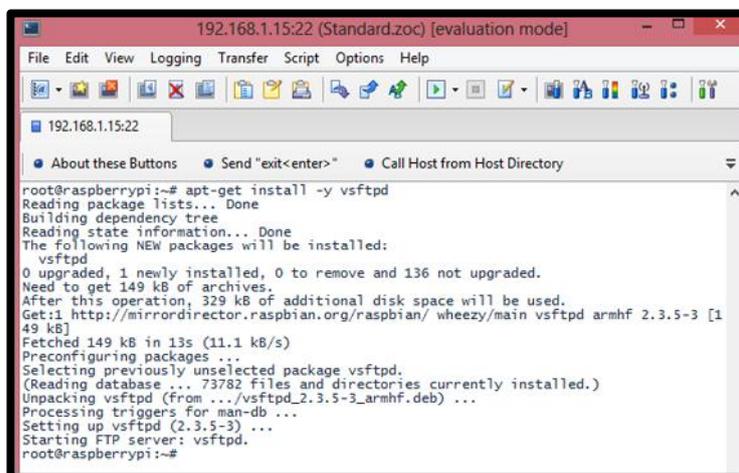
Fuente: Autor

7. El Servidor FTP

El servidor FTP trabaja de capa de aplicación en el modelo cliente-servidor permiten a los usuarios transferir archivos a un sistema remoto, ejecutando un programa de servidor FTP y desde este.

7.1 Instalación del servidor FTP

En la consola escribimos el comando `sudo apt-get install -y vsftpd` para instalar el servidor vsftpd.



```

root@raspberrypi:~# apt-get install -y vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
 vsftpd
0 upgraded, 1 newly installed, 0 to remove and 136 not upgraded.
Need to get 149 kB of archives.
After this operation, 329 kB of additional disk space will be used.
Get:1 http://mirrordirector.raspbian.org/raspbian/ wheezy/main vsftpd armhf 2.3.5-3 [149 kB]
Fetched 149 kB in 13s (11.1 kB/s)
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 73782 files and directories currently installed.)
Unpacking vsftpd (from ../vsftpd_2.3.5-3_armhf.deb) ...
Processing triggers for man-db ...
Setting up vsftpd (2.3.5-3) ...
Starting FTP server: vsftpd.
root@raspberrypi:~#

```

Figura 49 Instalación del servidor vsftpd

Fuente: Autor

7.2 Configuración del servidor FTP.

Editamos el archivo de configuración del servicio FTP.



```

root@raspberrypi:~# nano /etc/vsftpd.conf

```

Figura 50 Archivo de configuración del servidor vsftpd

Fuente: Autor

Descomentamos las siguientes líneas para permitir la escritura de archivos a los usuarios de la Raspberry Pi B+.

```
local_enable=YES
```

```
write_enable=YES
```

```

192.168.1.15:22 (Standard.zoc) [evaluation mode]
File Edit View Logging Transfer Script Options Help
192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory
GNU nano 2.2.6 File: /etc/vsftpd.conf Modified
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=YES
#
# Run standalone with IPv6?
# Like the listen parameter, except vsftpd will listen on an IPv6 socket
# instead of an IPv4 one. This parameter and the listen parameter are mutually
# exclusive.
#listen_ipv6=YES
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=NO
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will

```

Figura 51 Edición del archivo de configuración vsftpd para escritura

Fuente: Autor

Si deseamos que nos aparezca un banner o mensaje de bienvenida al loguearnos por Ftp, descomentamos y modificamos esta línea:

```

192.168.1.15:22 (Standard.zoc) [evaluation mode]
File Edit View Logging Transfer Script Options Help
192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory
GNU nano 2.2.6 File: /etc/vsftpd.conf Modified
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsuare
#
# Enable this and the server will recognise asynchronous ABOR requests. Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that on some FTP servers, ASCII support allows a denial of service
# attack (DoS) via the command "SIZE /big/file" in ASCII mode. vsftpd
# predicted this attack and has always been safe, reporting the size of the
# raw file.
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
ftpd_banner=BIENVENIDOS A LA GRANJA PRADERA-FTP.
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
# You may restrict local users to their home directories. See the FAQ for
# the possible risks in this before using chroot_local_user or
# chroot_list_enable below.
#chroot_local_user=NO

```

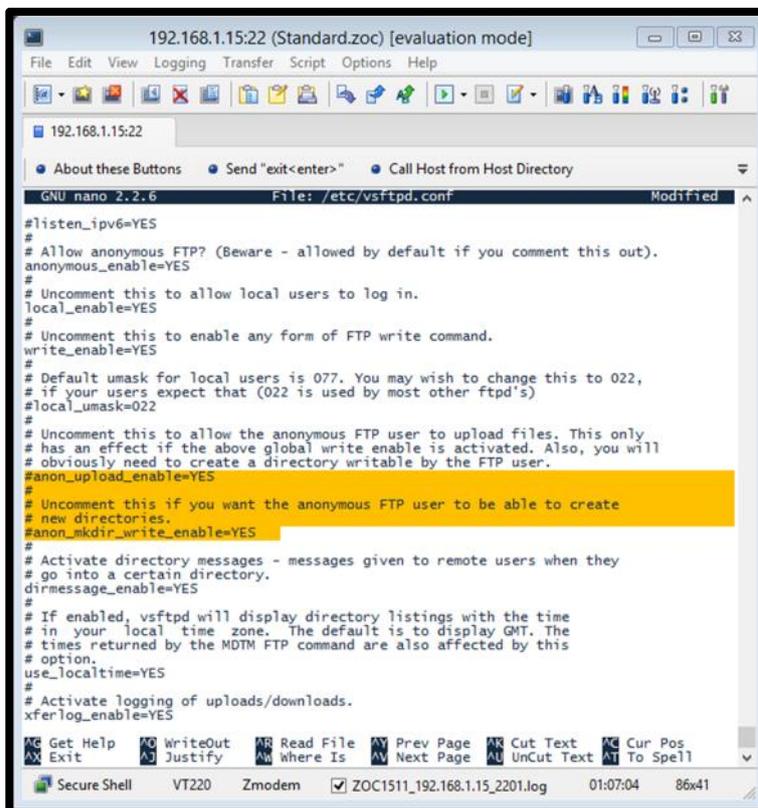
Figura 52 Banner de bienvenida

Fuente: Autor

Si deseamos que el usuario anonymous pueda editar y crear nuevos directorios, descomentamos estas 2 líneas:

```
anon_upload_enable=YES
```

```
anon_mkdir_write_enable=YES
```



```

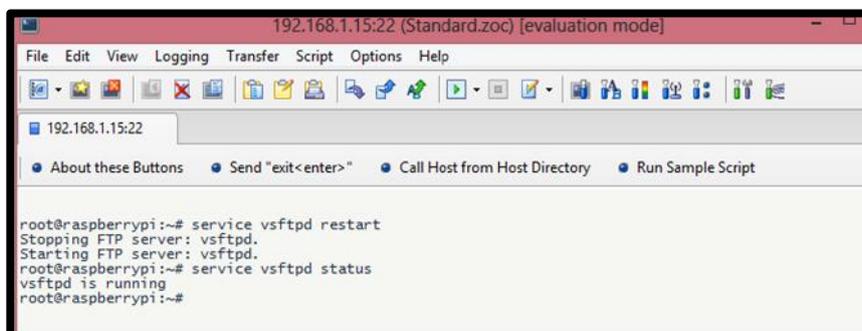
192.168.1.15:22 (Standard.zoc) [evaluation mode]
File Edit View Logging Transfer Script Options Help
192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory
GNU nano 2.2.6 File: /etc/vsftpd.conf Modified
#listen_ipv6=YES
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# If enabled, vsftpd will display directory listings with the time
# in your local time zone. The default is to display GMT. The
# times returned by the MDTM FTP command are also affected by this
# option.
use_localtime=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
Secure Shell VT220 Zmodem ZOC1511_192.168.1.15_2201.log 01:07:04 86x41

```

Figura 53 Edición del archivo de configuración vsftpd para usuario anonymous

Fuente: Autor

Guardamos el fichero con los cambios realizados y reiniciamos el servicio.



```

192.168.1.15:22 (Standard.zoc) [evaluation mode]
File Edit View Logging Transfer Script Options Help
192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory Run Sample Script
root@raspberrypi:~# service vsftpd restart
Stopping FTP server: vsftpd.
Starting FTP server: vsftpd.
root@raspberrypi:~# service vsftpd status
vsftpd is running
root@raspberrypi:~#

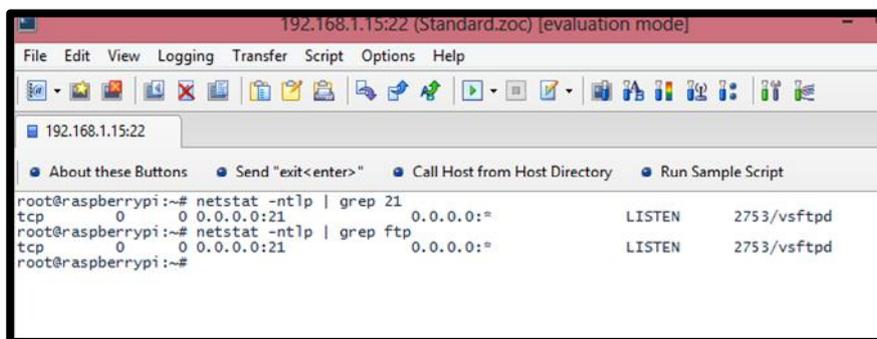
```

Figura 54 Reinicio del servicio vsftpd

Fuente: Autor

7.3 Verificación del puerto del servidor FTP.

Escribimos en el terminal el comando `netstat -ntlp | grep 21`, y verificamos que el puerto FTP este siendo escuchado:



```
192.168.1.15:22 (Standard.zoc) [evaluation mode]
File Edit View Logging Transfer Script Options Help
192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory Run Sample Script
root@raspberrypi:~# netstat -ntlp | grep 21
tcp        0      0 0.0.0.0:21          0.0.0.0:*          LISTEN    2753/vsftpd
root@raspberrypi:~# netstat -ntlp | grep ftp
tcp        0      0 0.0.0.0:21          0.0.0.0:*          LISTEN    2753/vsftpd
root@raspberrypi:~#
```

Figura 55 Verificación del puerto del servidor vsftpd

Fuente: Autor

7.4 Instalación y comprobación del cliente FTP.

En el cliente nos descargamos el programa WinSCP, y comprobamos que se pueda acceder con el usuario anonymous. Para lo cual damos clic en Anonymous login y luego clic en Login.

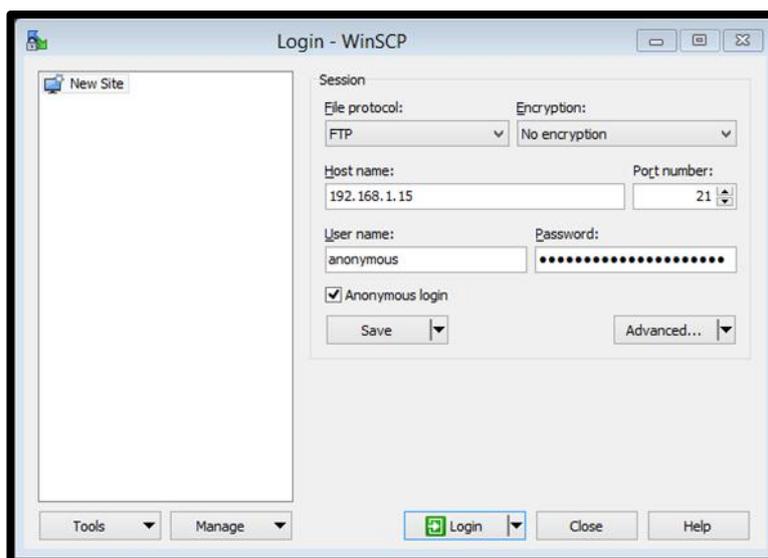


Figura 56 Ejecución del software WinSCP

Fuente: Autor

Podemos observar que ingreso correctamente con el usuario Anonymous.

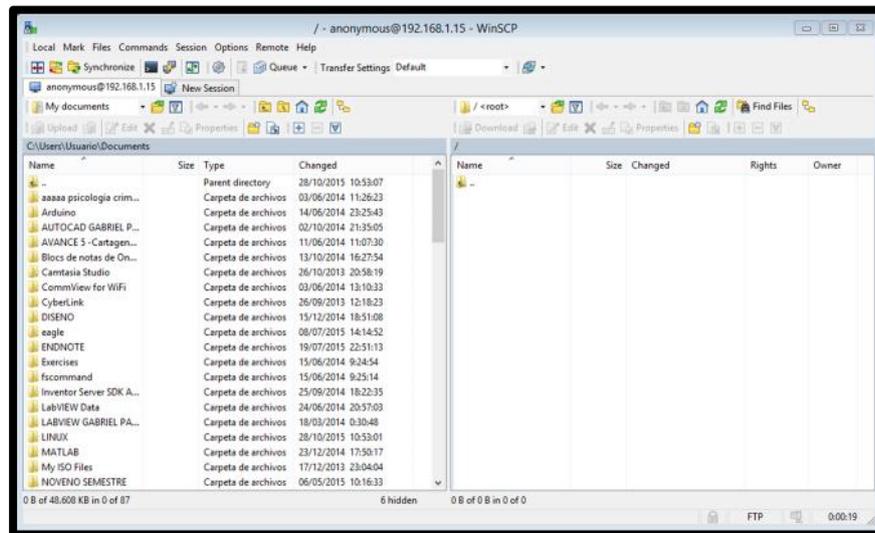


Figura 57 Ingreso correcto del usuario anonymous

Fuente: Autor

Comprobamos que ingrese con el usuario pi y su respectiva contraseña. Damos clic en Login.

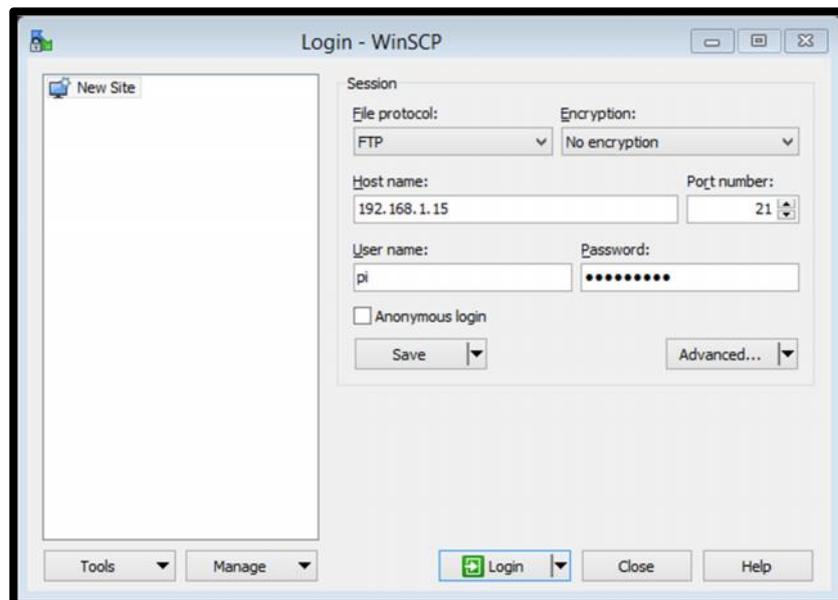


Figura 58 Ingreso al servidor vsftpd con el usuario pi

Fuente: Autor

Y podemos observar que ingreso correctamente al usuario pi, ya se puede recibir y transferir archivos.

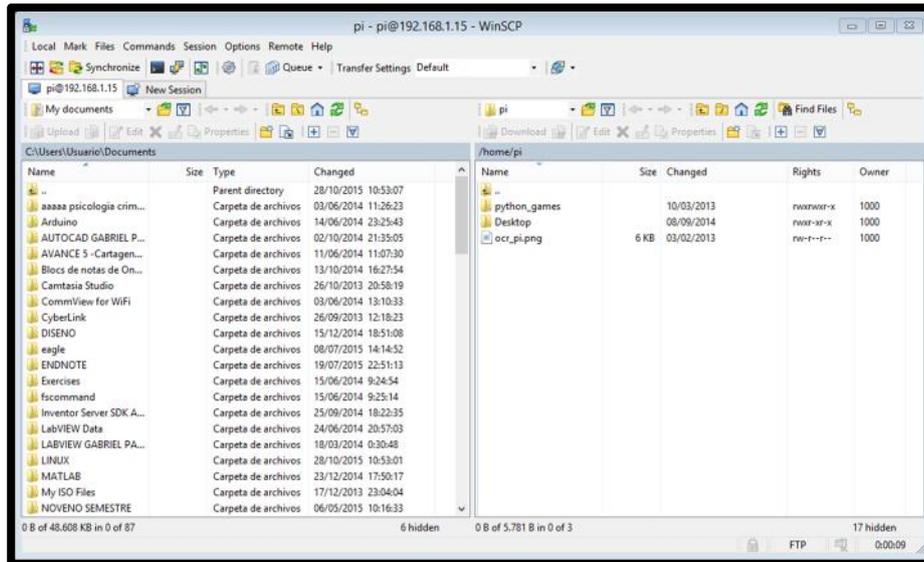


Figura 59 Ingreso correcto al usuario pi

Fuente: Autor

7.5 Agregando un nuevo usuario al servidor FTP.

Agregamos un nuevo usuario a través del comando `useradd pradera`. Creamos una carpeta donde va a contener todos los archivos del usuario `pradera` a través del comando `mkdir /home/pradera`, le otorgamos permisos a través del comando `chown pradera:users /home/pradera` y creamos una contraseña para el usuario `pradera` con el comando `passwd pradera`.

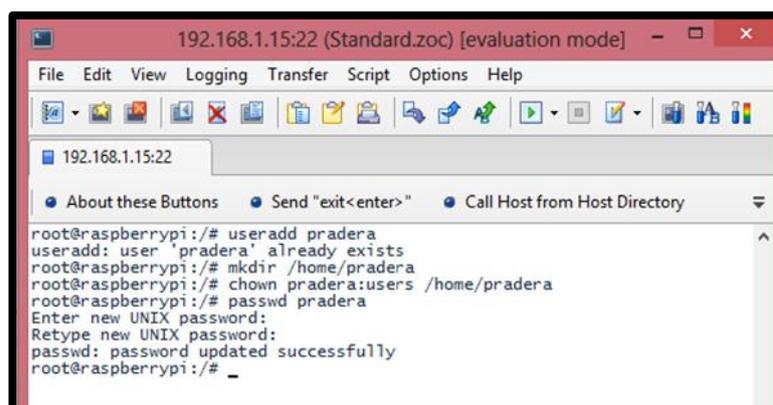


Figura 60 Creando un nuevo usuario al servidor vsftpd

Fuente: Autor

Usuario: pradera

Contraseña: pradera

Comprobamos que el nuevo usuario pueda acceder al servicio FTP, colocando el usuario pradera y la respectiva contraseña.

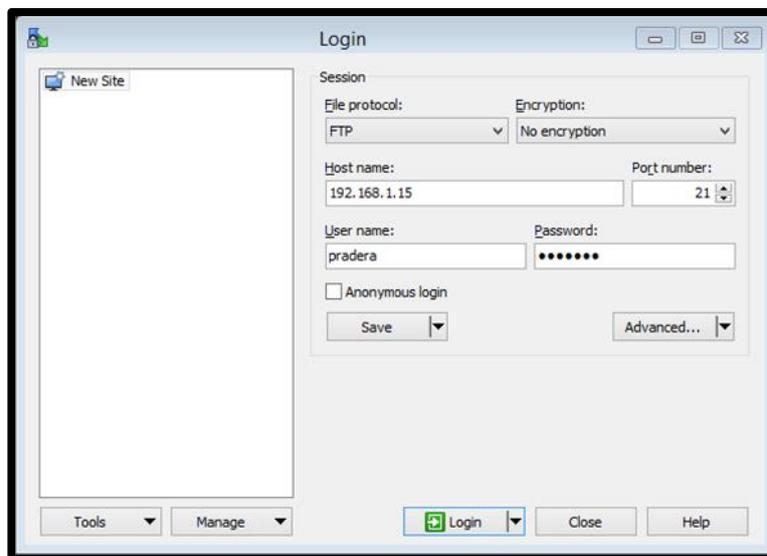


Figura 61 Ingreso al servidor vsftpd con el nuevo usuario

Fuente: Autor

Podemos observar que ingreso correctamente al usuario pradera, y ya se puede recibir y transferir archivos.

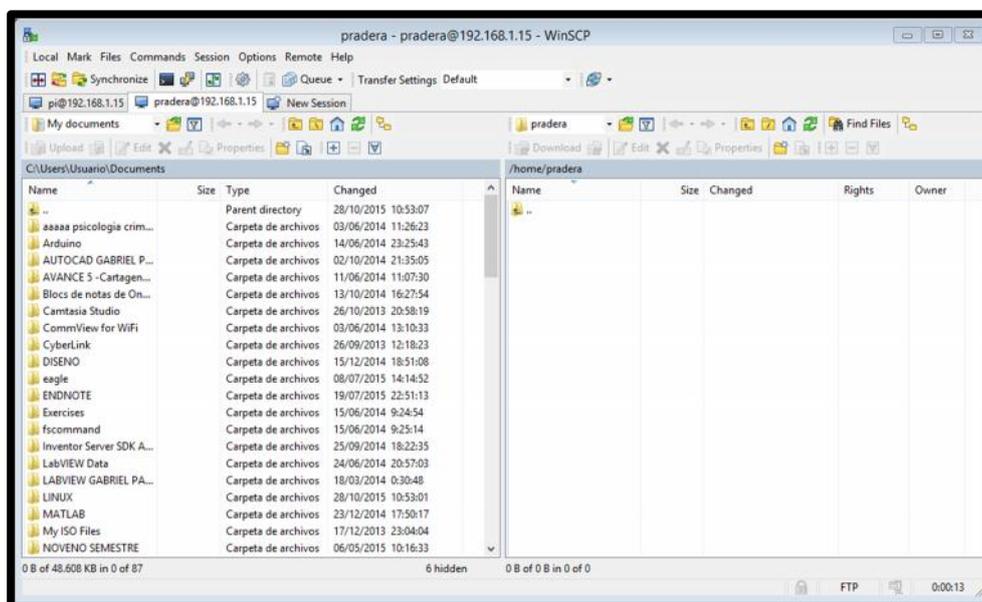


Figura 62 Ingreso correcto a la carpeta del nuevo usuario

Fuente: Autor

Para transferir cualquier fichero desde el host remoto hacia el raspberry PI, basta con seleccionar el fichero que se desea transferir y soltar sobre la carpeta /home/pradera. Si se desea transferir los archivos desde el raspberrry pi hacia el host remoto simplemente se realiza el proceso anterior inversamente.

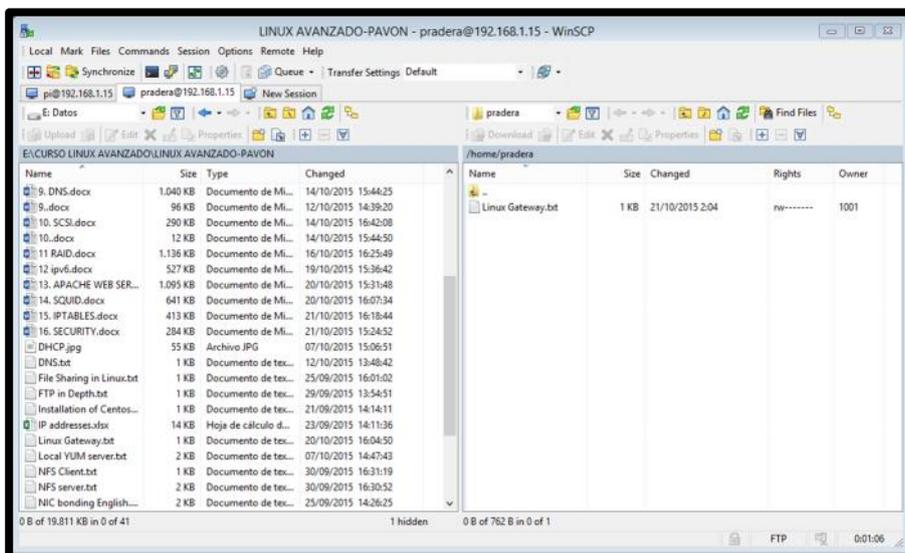


Figura 63 Transferencia de archivo hacia el nuevo usuario (pradera)

Fuente: Autor

Comprobamos que el fichero se transfirió exitosamente desde el host remoto hacia el raspberry pi.



Figura 64 Comprobación de que el archivo se transfirió exitosamente hacia el usuario pradera

Fuente: Autor

Se recomienda dar permisos de lectura, escritura y ejecución para los usuarios de todo el contenido del usuario pradera, a través del comando `chmod 775 /home/pradera/*`.

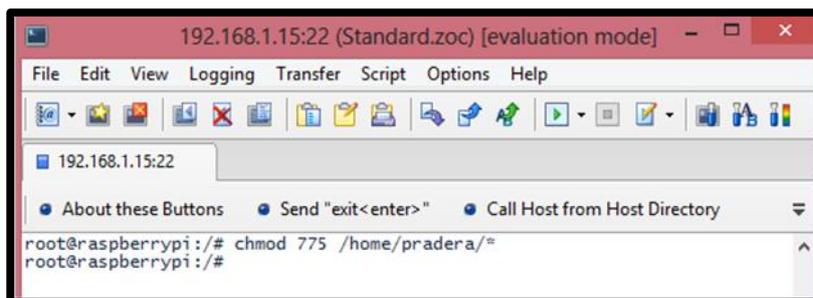


Figura 65 Otorgando permisos de lectura, escritura y ejecución a todo el contenido del usuario pradera

Fuente: Autor

8. VNC (Virtual Network Computing).

Virtual Network Computing es un sistema de intercambio de escritorio gráfico que le permite controlar de forma remota la interfaz de escritorio de un ordenador a otro. Transmite los de teclado y ratón eventos del controlador, y recibe actualizaciones a la pantalla por la red desde el host remoto.

8.1 Instalación del servidor VNC.

Ingresamos por acceso remoto a través de SSH al Raspberry Pi, en la consola escribimos el comando **sudo apt-get install tightvncserver** que nos permite instalar el paquete.

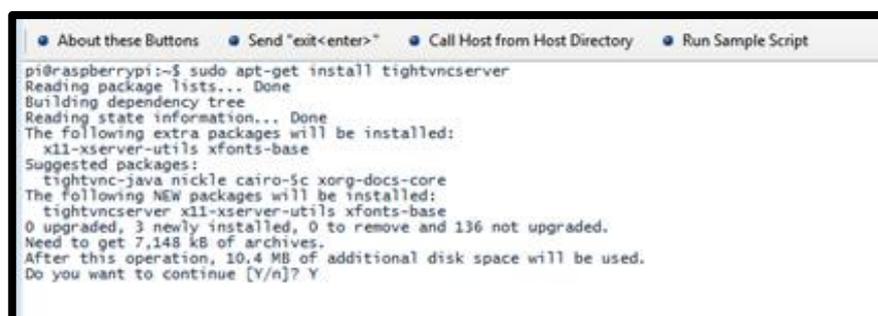


Figura 66 Instalación del servidor VNC

Fuente: Autor

8.2 Ejecución del servidor TightVNC.

A continuación ejecutamos el comando **tightvncserver**, en la cual nos pedirá que introduzcamos una contraseña para poder acceder al equipo.

Contraseña: pradera



```
pi@raspberrypi:~$ tightvncserver
You will require a password to access your desktops.
Password:
Verify:
Would you like to enter a view-only password (y/n)? y
Password:
Verify:
New 'X' desktop is raspberrypi:1
Creating default startup script /home/pi/.vnc/xstartup
Starting applications specified in /home/pi/.vnc/xstartup
Log file is /home/pi/.vnc/raspberrypi:1.log
pi@raspberrypi:~$
```

Figura 67 Ejecución del servidor TightVNC

Fuente: Autor

8.3 Inicialización en el sistema.

Nos logueamos como root con el comando **su -**, e introducimos la contraseña de root. Creamos un fichero de inicialización en el sistema en el directorio `/etc/init.d/`, en este caso se denomina como `vncboot`.



```
root@raspberrypi:/home/pi# cd /etc/init.d/
root@raspberrypi:/etc/init.d# nano vncboot
```

Figura 68 Creación de fichero vncboot

Fuente: Autor

Ingresamos los parámetros para iniciar y parar el demonio. Guardamos los cambios y salimos del fichero.



```

GNU nano 2.2.6 File: vncboot
### BEGIN INIT INFO
# Provides: vncboot
# Required-Start: $remote_fs $syslog
# Required-Stop: $remote_fs $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Start VNC Server at boot time
# Description: Start VNC Server at boot time.
### END INIT INFO

#!/bin/sh
# /etc/init.d/vncboot

USER=pi
HOME=/home/pi

export USER HOME

case "$1" in
start)
echo "Starting VNC Server"
#Insert your favoured settings for a VNC session
su - pi -c "/usr/bin/vncserver :1 -geometry 1280x800 -depth 16 -pixelformat rgb565"
;;
stop)
echo "Stopping VNC Server"
/usr/bin/vncserver -kill :1
;;
*)
echo "Usage: /etc/init.d/vncboot {start|stop}"
exit 1
;;
esac

exit 0

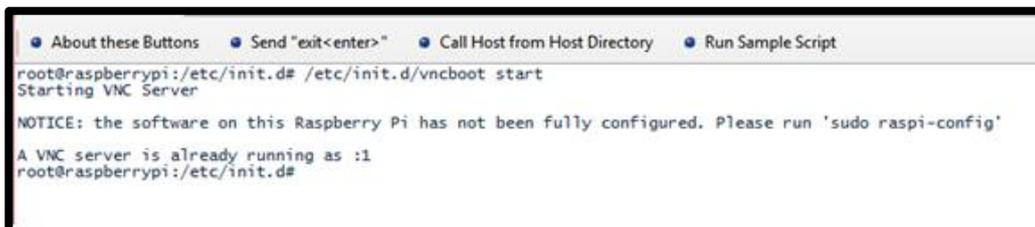
```

Figura 69 Parámetros para iniciar y parar el servicio VNC

Fuente: Autor

Para ejecutar el fichero manualmente se lo puede realizar de las siguientes maneras:

pi@raspberrypi:~\$ sudo /etc/init.d/vncboot start //Iniciar el demonio.



```

root@raspberrypi:/etc/init.d# /etc/init.d/vncboot start
Starting VNC Server

NOTICE: the software on this Raspberry Pi has not been fully configured. Please run 'sudo raspi-config'

A VNC server is already running as :1
root@raspberrypi:/etc/init.d#

```

Figura 70 Iniciar el servicio VNC.

Fuente: Autor

pi@raspberrypi:~\$ sudo /etc/init.d/vncboot stop //Parar el demonio.



```

root@raspberrypi:~# /etc/init.d/vncboot stop
Stopping VNC Server
Killing Xtightvnc process ID 2569
root@raspberrypi:~# _

```

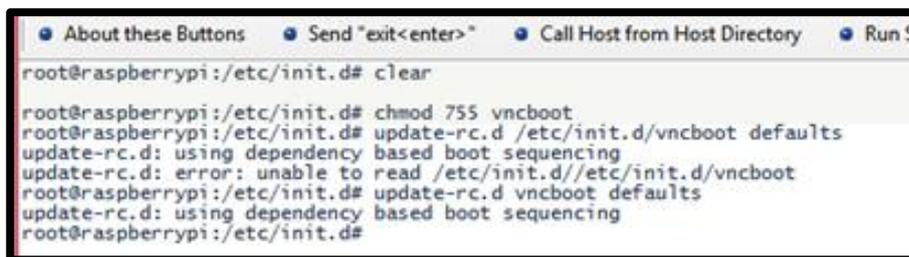
Figura 71 Parar el servicio VNC

Fuente: Autor

Podemos observar que el servicio VNC está corriendo en el sistema, el :1 significa que está escuchando en el puerto 5901.

8.4 Automatización y ejecución en el arranque del sistema.

Otorgamos los permisos necesarios tanto de lectura, escritura y ejecución al fichero, a través del comando **chmod 775 vncboot**. Y encolamos el fichero para la ejecución al iniciar el sistema con el comando **update-rc.d vncboot defaults**.



```

root@raspberrypi:/etc/init.d# clear

root@raspberrypi:/etc/init.d# chmod 775 vncboot
root@raspberrypi:/etc/init.d# update-rc.d /etc/init.d/vncboot defaults
update-rc.d: using dependency based boot sequencing
update-rc.d: error: unable to read /etc/init.d/etc/init.d/vncboot
root@raspberrypi:/etc/init.d# update-rc.d vncboot defaults
update-rc.d: using dependency based boot sequencing
root@raspberrypi:/etc/init.d#

```

Figura 72 Automatización y ejecución del fichero VNC en el arranque del sistema

Fuente: Autor

Reiniciamos el sistema, a través del comando `reboot` o `shutdown -r now`.



```

root@raspberrypi:/etc/init.d# reboot_

```

Figura 73 Reinicio del sistema

Fuente: Autor

8.5 Instalación y comprobación del cliente VNC.

En la maquina cliente instalamos el programa VNC Viewer, ejecutamos el programa. Y escribimos la dirección Ip del servidor seguido del número del puerto :1, tal y como se muestra en la imagen siguiente.

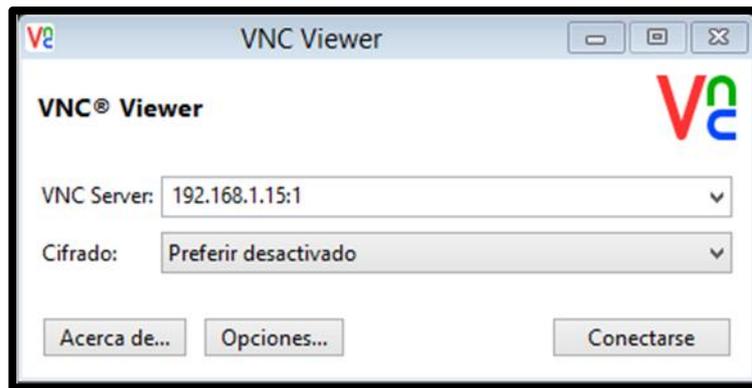


Figura 74 Ejecución del cliente VNC

Fuente: Autor

A continuación damos clic en Continuar.

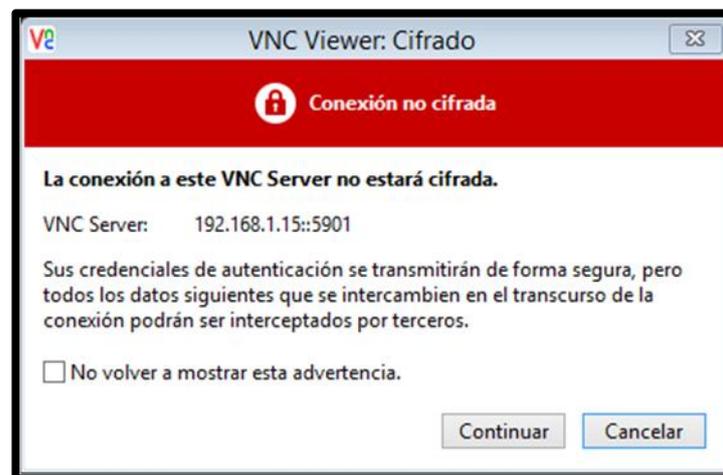


Figura 75 Mensaje de VNC Viewer

Fuente: Autor

Colocamos la contraseña que pusimos en la ejecución del servidor TightVNC, en este caso es **pradera**. Y damos clic en aceptar.

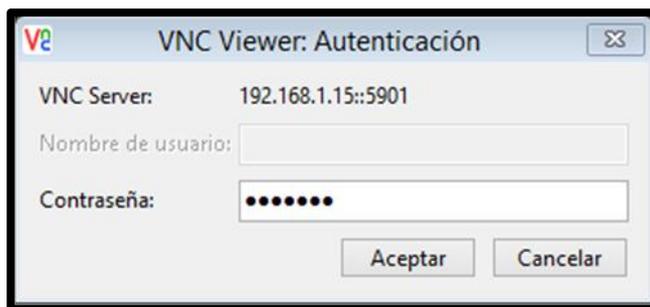


Figura 76 Colocación de la contraseña para ingresar por VNC

Fuente: Autor

Comprobamos que se puede acceder al sistema correctamente.

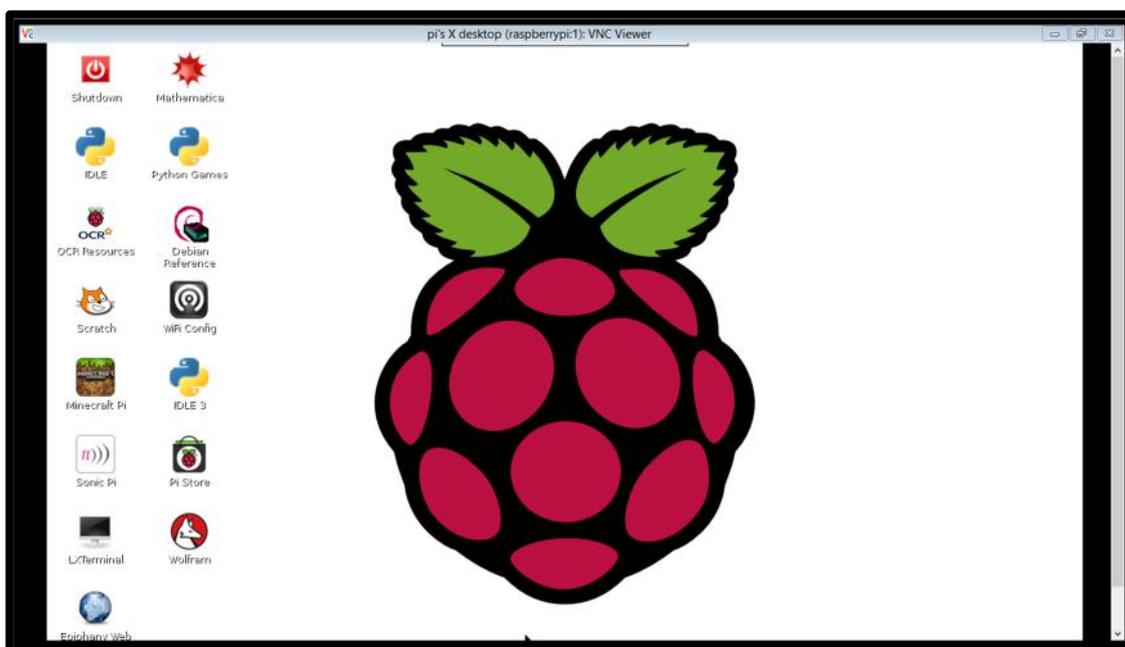


Figura 77 Comprobación del ingreso correcto por VNC a la Raspberry Pi B+

Fuente: Autor

9. MODULO RTC (Real Time Clock)

La raspberry Pi B+ mientras esté conectada a la red con Internet establece una conexión con el servidor NTP (Network Time Protocol) para ajustar la hora del sistema. Pero en caso de no estar conectado y para evitar fallas en el tiempo y fecha se optó por instalar un módulo RTC (Real Time Clock). El módulo RTC resolverá el problema, así

cuando la raspberry Pi B+ se inicie utiliza la fecha y hora de este módulo para establecer correctamente dichos parámetros en su sistema.

Este módulo utiliza la comunicación I2C (Circuitos Inter-Integrados), y es utilizado para conectar periféricos de bajas velocidades.

9.1 Activación de I2C en la Raspberry Pi B+.

Desde la consola escribimos el comando **sudo raspi-config**, que es una utilidad que nos permite realizar las configuraciones del sistema.

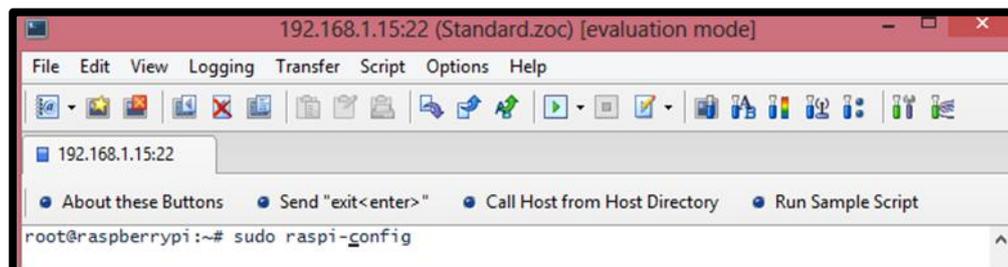


Figura 78 Comando para realizar configuraciones del sistema

Fuente: Autor

Nos abrirá la utilidad raspi-config.

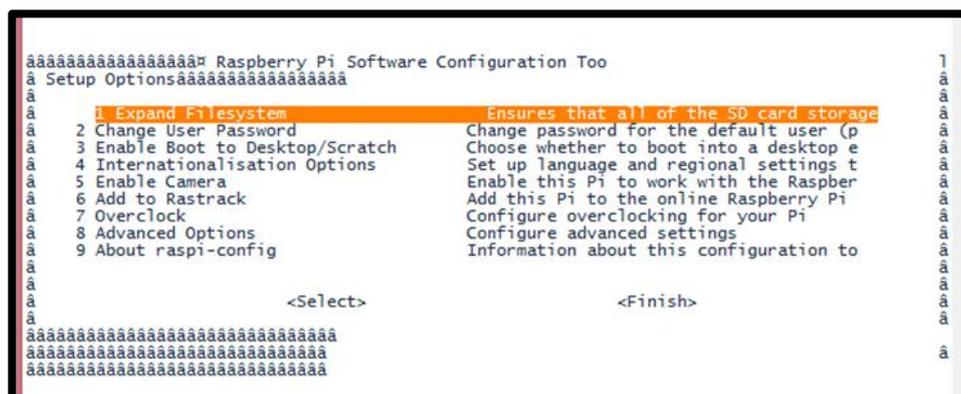


Figura 79 Opciones de configuraciones del sistema

Fuente: Autor

Seleccionamos la opción 8 “Advanced Options”.

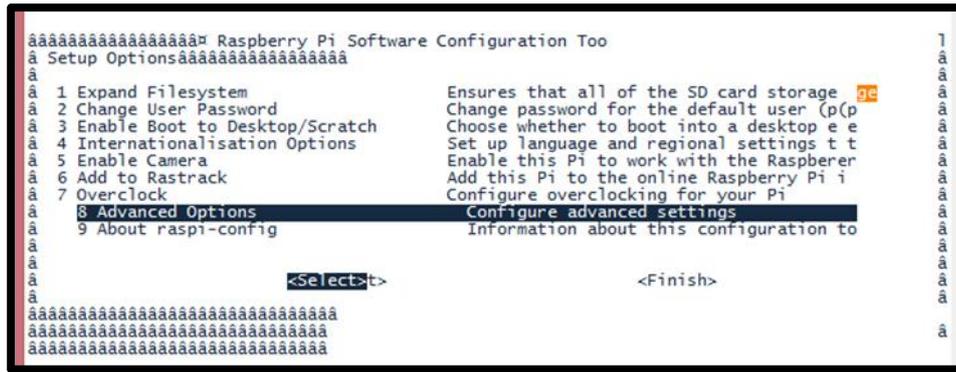


Figura 80 Selección de opciones avanzadas

Fuente: Autor

Seleccionamos A6, la comunicación I2C.

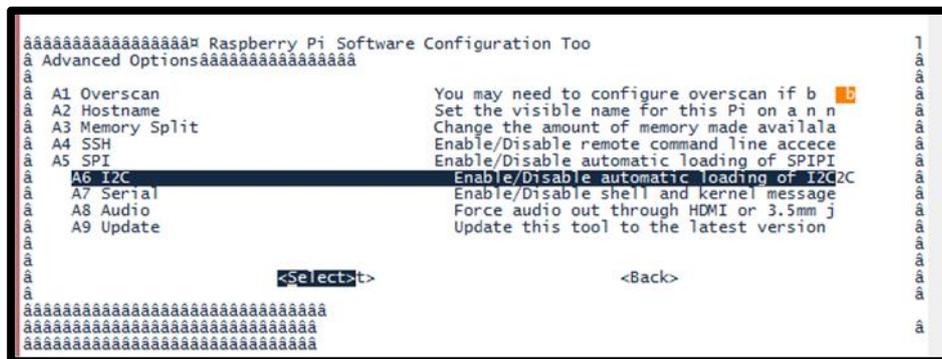


Figura 81 Selección de configuración de la comunicación I2C

Fuente: Autor

Nos preguntará si nos gustaría que el módulo del kernel i2c que se cargue por defecto. Seleccionamos Yes.

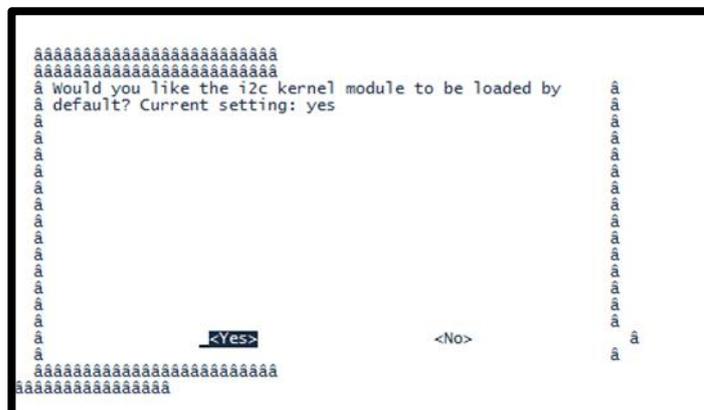
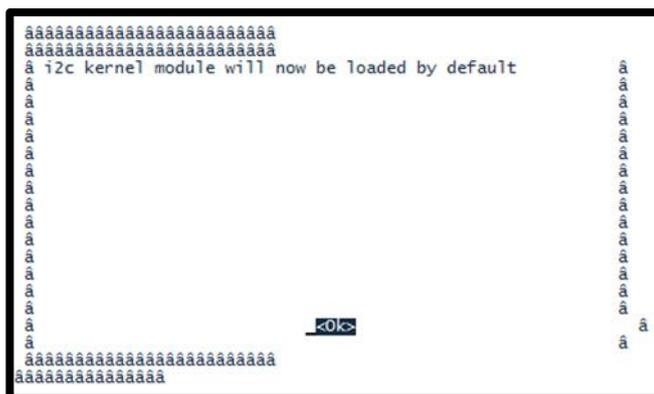


Figura 82 Modulo I2C cargando por defecto

Fuente: Autor

La pantalla nos indicará que el módulo del kernel I2c ahora se carga por defecto.



```

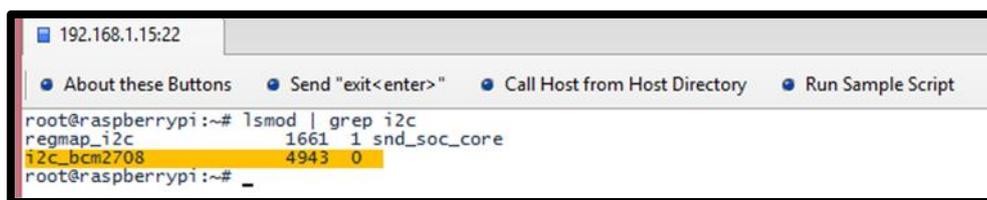
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
i2c kernel module will now be loaded by default
                                     <OK>
aaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaa

```

Figura 83 Modulo I2C cargado correctamente por defecto

Fuente: Autor

Reiniciamos y comprobamos que el módulo I2c está activado.



```

192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory Run Sample Script
root@raspberrypi:~# lsmod | grep i2c
regmap_i2c          1661 1 snd_soc_core
i2c_bcm2708        4943 0
root@raspberrypi:~# _

```

Figura 84 Comprobación del módulo I2C este activado

Fuente: Autor

9.2 Configuración de la zona horaria.

Para obtener la hora exacta del servidor NTP se recomienda estar conectado a Internet. Escribimos nuevamente el comando **sudo raspi-config**, Y seleccionamos la opción 4 Internationalisation Options.

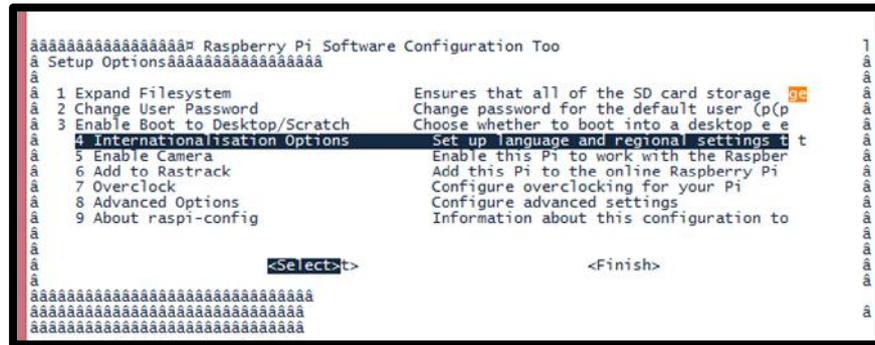


Figura 85 Selección de opciones internacionales

Fuente: Autor

A continuación seleccionamos I2, Change Timezone.

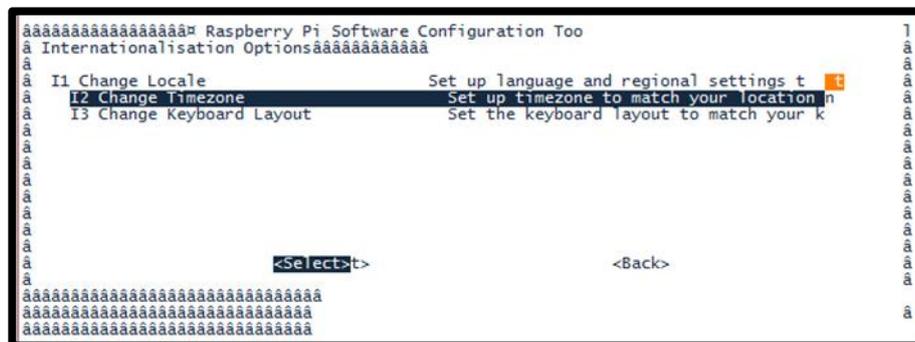


Figura 86 Cambiamos la zona horaria

Fuente: Autor

Seleccionamos América.



Figura 87 Seleccionanos el área geográfica

Fuente: Autor

Y buscamos la opción de Guayaquil y la seleccionamos.

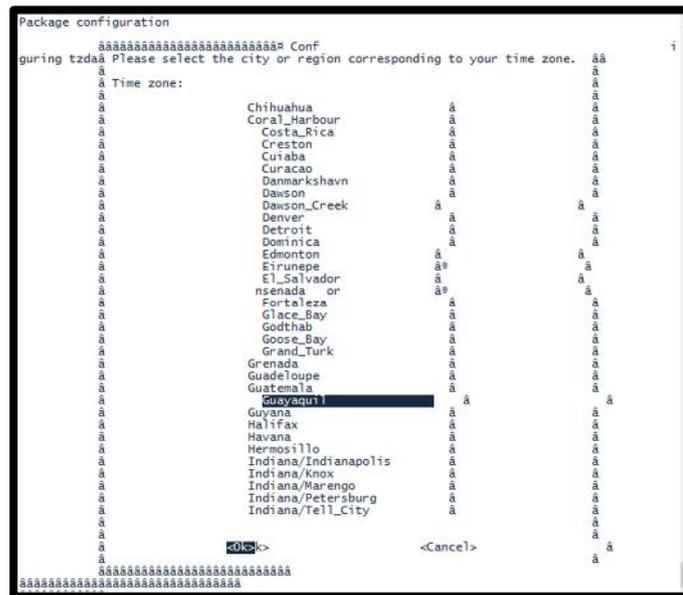


Figura 88 Seleccionamos la región más cercana a la que nos encontremos

Fuente: Autor

Damos enter en la opción de Finish.

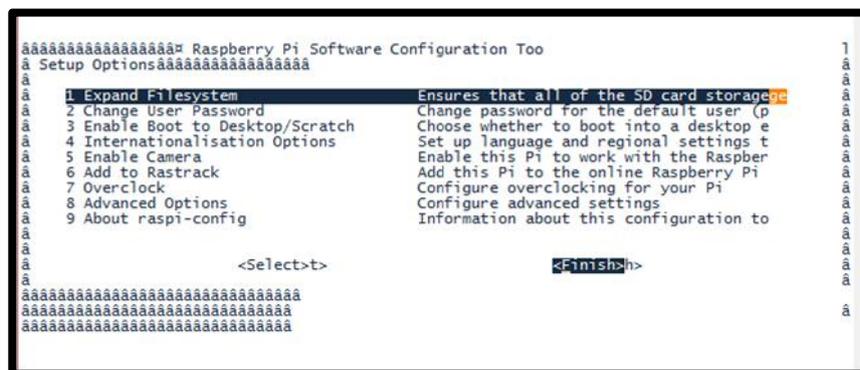


Figura 89 Final de configuración de la zona horaria

Fuente: Autor

Escribimos el comando **date** para comprobar la fecha y hora actual.

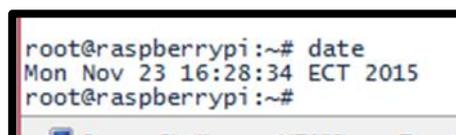


Figura 90 Comprobación de la fecha y hora actualizadas

Fuente: Autor

9.3 Edición de los módulos.

Editamos el fichero de los módulos con el comando `nano /etc/modules`.

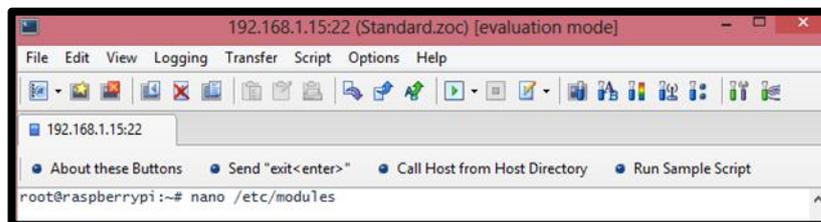


Figura 91 Fichero de configuración de los módulos

Fuente: Autor

Debemos agregar los siguientes módulos `i2c-bcm2708`, `i2c-dev`, `rtc-ds1307`.

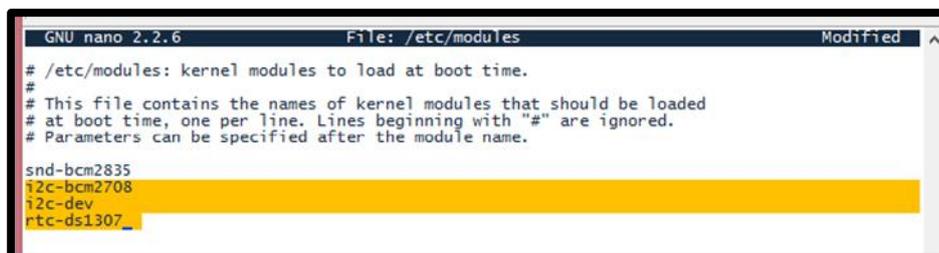


Figura 92 Agregando los módulos que son necesarios

Fuente: Autor

9.4 Edición de la lista negra del Raspberry Pi B+.

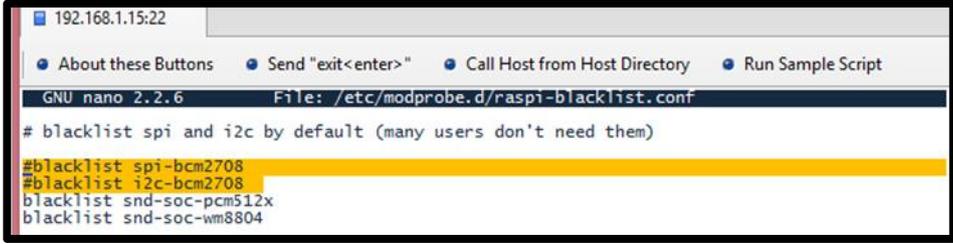
Para evitar que el Raspberry nos bloquee las comunicaciones I2C y SPI, editamos el archivo `/etc/modprobe.d/raspi-blacklist.conf`.



Figura 93 Fichero de configuración de las listas negras del Raspberry.

Fuente: Autor

Comentamos las siguientes líneas, guardamos el fichero y salimos.



```

192.168.1.15:22
GNU nano 2.2.6 File: /etc/modprobe.d/raspi-blacklist.conf
# blacklist spi and i2c by default (many users don't need them)
#blacklist spi-bcm2708
#blacklist i2c-bcm2708
blacklist snd-soc-pcm512x
blacklist snd-soc-wm8804

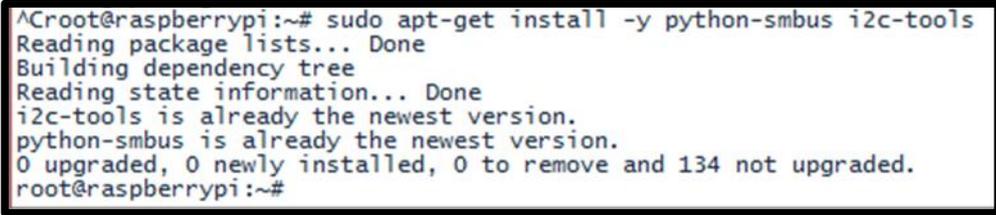
```

Figura 94 Comentando los módulos SPI e I2C

Fuente: Autor

9.5 Instalación de utilidades.

Instalamos algunas utilidades que son necesarias para que funcione correctamente.



```

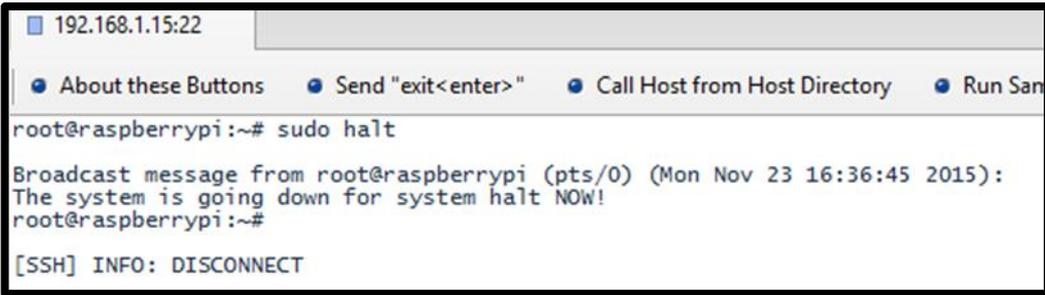
^Croot@raspberrypi:~# sudo apt-get install -y python-smbus i2c-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version.
python-smbus is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 134 not upgraded.
root@raspberrypi:~#

```

Figura 95 Instalación de utilidades

Fuente: Autor

Apagamos la raspberry y conectamos el módulo RTC.



```

192.168.1.15:22
root@raspberrypi:~# sudo halt
Broadcast message from root@raspberrypi (pts/0) (Mon Nov 23 16:36:45 2015):
The system is going down for system halt NOW!
root@raspberrypi:~#
[SSH] INFO: DISCONNECT

```

Figura 96 Apagado del sistema

Fuente: Autor

9.6 Comprobación de I2C y RTC estén habilitados.

Con el comando `lsmod | grep i2c`, verificamos que están activados los módulos i2c.

```

root@raspberrypi:~# lsmod | grep i2c
i2c_dev                5769  0
regmap_i2c             1661  1 snd_soc_core
i2c_bcm2708            4943  0
root@raspberrypi:~#

```

Figura 97 Verificación del módulo I2C

Fuente: Autor

Con el comando **lsmod | grep rtc**, verificamos que está activado el módulo rtc.

```

root@raspberrypi:~# lsmod | grep rtc
rtc_ds1307              7494  0
root@raspberrypi:~# _

```

Figura 98 Verificación del módulo RTC

Fuente: Autor

9.7 Pruebas de hardware.

Una vez conectados el hardware del módulo RTC con la raspberry Pi B+, en la consola del terminal escribimos el comando **sudo i2cdetect -y 1**, el cual nos sirve para detectar que tenemos conectado un dispositivo o en este caso es el módulo RTC en la interfaz I2C y su dirección es 0x68.

```

root@raspberrypi:~# sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  57  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  68  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
root@raspberrypi:~# _

```

Figura 99 Dirección del dispositivo RTC

Fuente: Autor

9.8 Carga de los módulos al kernel.

Lo primero es cargar el módulo RTC e I2C en el Kernel para ello escribimos en la consola los siguientes comandos:

```

192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host Directory Run Sample Script
root@raspberrypi:~# sudo modprobe i2c-bcm2708
root@raspberrypi:~# sudo modprobe rtc-ds1307
root@raspberrypi:~# _

```

Figura 100 Carga de los módulos al kernel

Fuente: Autor

9.9 Configuración del dispositivo I2C.

El siguiente paso es añadir el dispositivo DS1307 como reloj del sistema en el archivo rc.local, para ello ejecutamos `sudo nano /etc/rc.local`. Y añadimos las siguientes líneas justo antes de “exit 0” :

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
```

```
sudo hwclock -s
```

```

GNU nano 2.2.6 File: /etc/rc.local Modified
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
#sudo python /home/pradera/shutdown.py
ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
sudo hwclock -s
exit 0

```

Figura 101 Añadiendo el dispositivo DS1307 como reloj del sistema

Fuente: Autor

Guardamos y reiniciamos con el comando `sudo reboot`.

Y comprobamos, ahora se repite el comando **sudo i2cdetect -y 1**, la dirección 0x68 se convierte en UU.

```

root@raspberrypi:~# sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  57  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  UU  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
root@raspberrypi:~# _

```

Figura 102 Comprobación que el dispositivo DS1307 esta como reloj del sistema

Fuente: Autor

Con el comando **sudo hwclock -r**, obtenemos la hora y fecha del sistema.

```

192.168.1.15:22
About these Buttons Send "exit<enter>" Call Host from Host
root@raspberrypi:~# sudo hwclock -r
Mon 23 Nov 2015 17:28:32 ECT -0.068327 seconds
root@raspberrypi:~#

```

Figura 103 Comando para verificar la hora del sistema

Fuente: Autor

Para escribir la hora y fecha en el módulo RTC, se utiliza los siguientes comandos:

```
#sudo date -s "29 AUG 2015 13:00:00"
```

```
#sudo hwclock --set --date="2015-10-06 13:00:00"
```

Para transferir la información al chip se utiliza el comando:

```
#sudo hwclock -w
```

Para comprobar que se ha escrito correctamente la hora en el módulo RTC, se escribe el comando:

```
#sudo hwclock -r
```

10. Apagado de todo el sistema.

Para apagar la Raspberry PI B+ de una manera segura se hace uso de un botón para activar un puerto GPIO (General Purpose Input Output) y así ejecutar un comando del sistema que permita apagar el sistema.

Se hace uso del módulo **python-rpi.gpio** que anteriormente ya fue instalado, seguidamente se creará un script en el lenguaje Python y se denomina shutdown.py.

10.1 Creación del script de apagado del sistema en Python.

En consola se escribe el comando `nano /home/pradera/shutdown.py`.

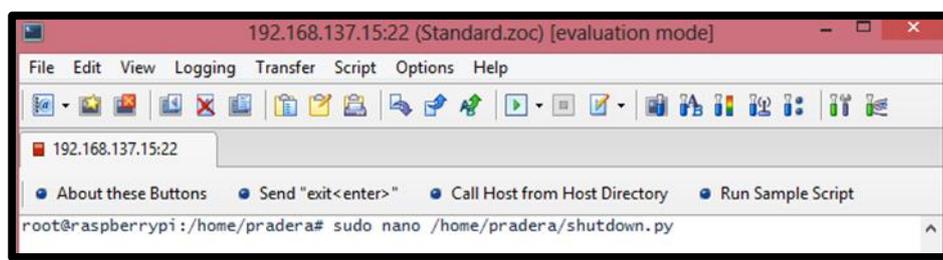
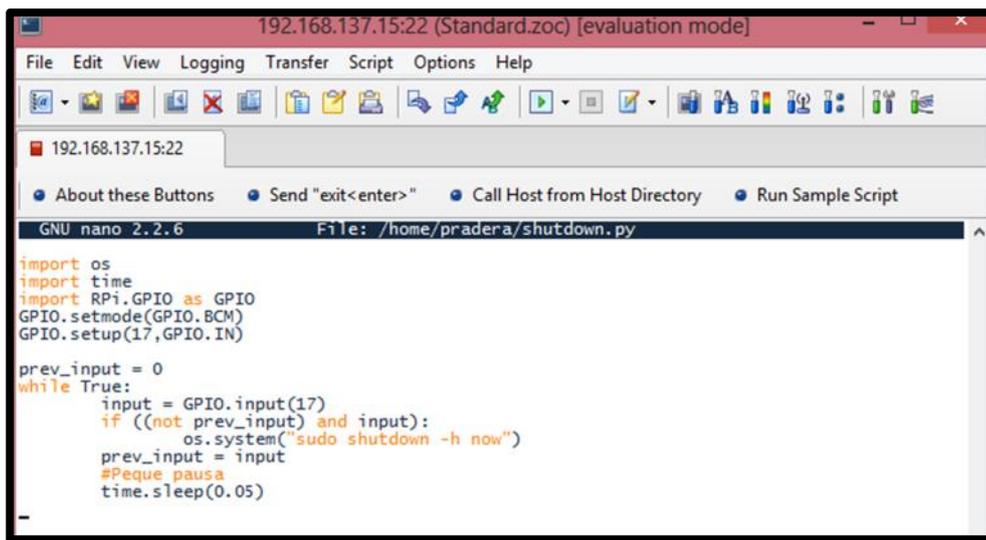


Figura 104 Creación del fichero de apagado del sistema

Fuente: Autor

Agregamos las siguientes líneas necesarias para que funcione correctamente el script.



```

GNU nano 2.2.6 File: /home/pradera/shutdown.py
import os
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(17,GPIO.IN)

prev_input = 0
while True:
    input = GPIO.input(17)
    if ((not prev_input) and input):
        os.system("sudo shutdown -h now")
    prev_input = input
    #Peque pausa
    time.sleep(0.05)

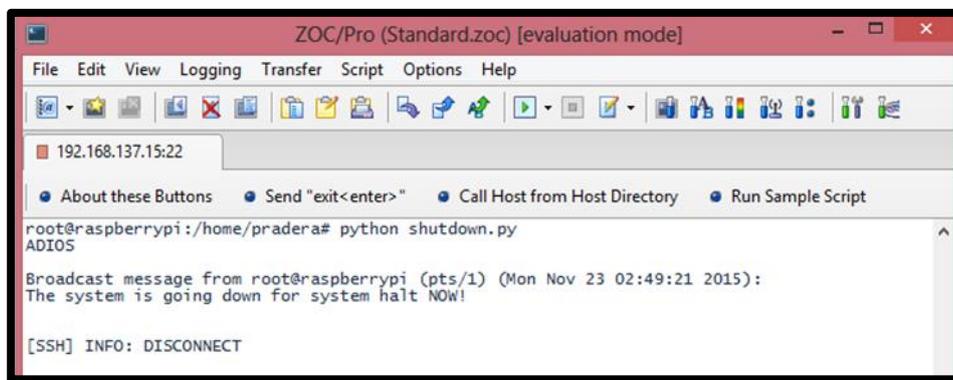
```

Figura 105 Script de apagado del sistema

Fuente: Autor

Donde se puede observar que el puerto GPIO que se utiliza es el número 17 y se lo define como entrada para el pulsador. Seguidamente entra la sentencia en el ciclo infinito while True, cuando el pulsador es presionado y es igual a un 1L (lógico) o 5V se ejecuta el comando `sudo shutdown -h now` que es el encargado de apagar todo el sistema, este comando puede ser reemplazado por cualquier otro como reboot, etc.

Ahora para comprobar que el script funciona perfectamente ejecutamos manualmente el scrip, y presionamos el botón y podemos observar un mensaje de ADIOS. El cual hace referencia a que el sistema se apagó correctamente.



```

ZOC/Pro (Standard.zoc) [evaluation mode]
192.168.137.15:22
root@raspberrypi:/home/pradera# python shutdown.py
ADIOS

Broadcast message from root@raspberrypi (pts/1) (Mon Nov 23 02:49:21 2015):
The system is going down for system halt NOW!

[SSH] INFO: DISCONNECT

```

Figura 106 Ejecución manual del script de apagado del sistema

Fuente: Autor

10.2 Automatización del script de apagado.

Para que el script se ejecute automáticamente cuando la Raspberry Pi B+ se encienda, es necesario modificar el fichero `/etc/rc.local`.

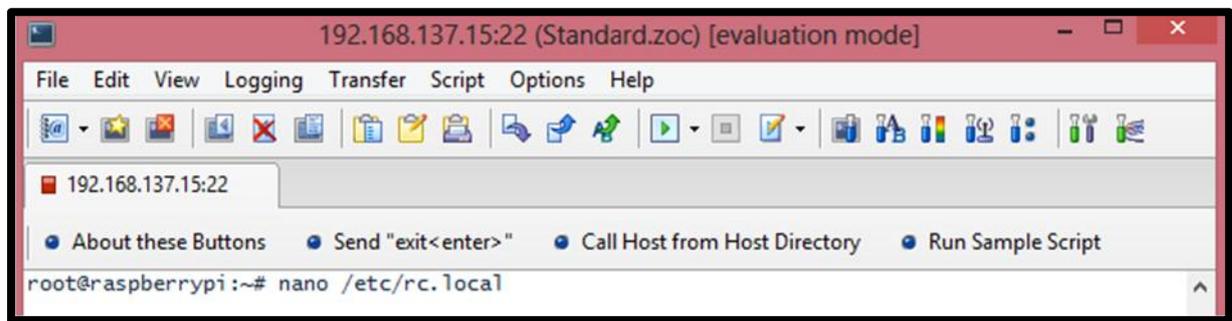


Figura 107 Fichero de ejecución automático del script de apagado del sistema

Fuente: Autor

En este fichero agregamos la siguiente línea, guardamos los cambios realizados y reiniciamos la Raspberry PI B+ para comprobar.

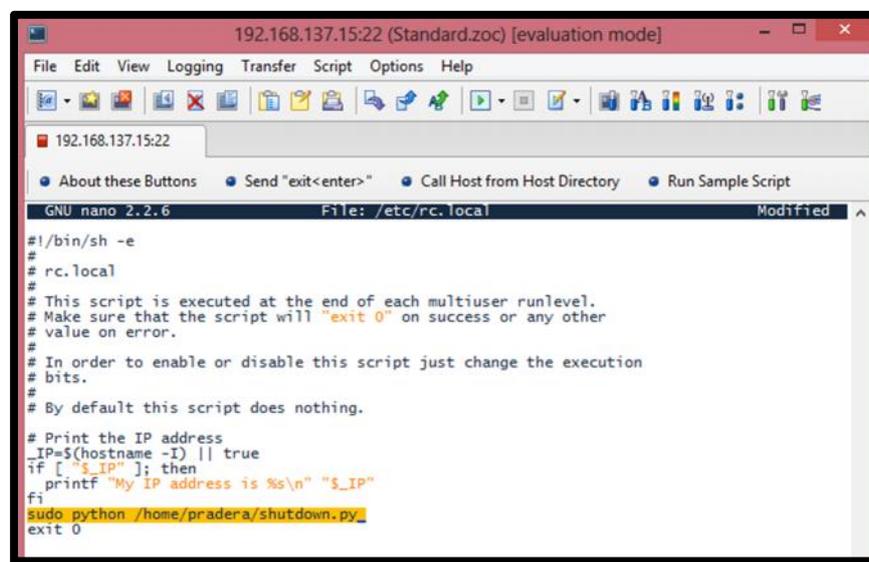
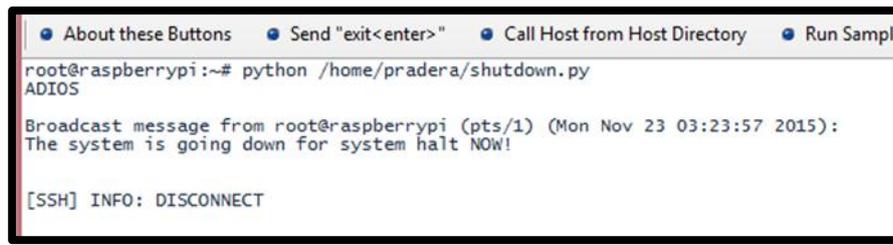


Figura 108 Agregando el comando para que se ejecute automáticamente el script de apagado

Fuente: Autor

Para comprobar presionamos el pulsador de apagado y se pudo observar que el sistema es apagado automáticamente sin necesidad de ejecutar ningún comando manualmente.



```
root@raspberrypi:~# python /home/pradera/shutdown.py
ADIOS
Broadcast message from root@raspberrypi (pts/1) (Mon Nov 23 03:23:57 2015):
The system is going down for system halt NOW!

[SSH] INFO: DISCONNECT
```

Figura 109 Comprobación de apagado de todo el sistema

Fuente: Autor

Para automatizar el script del ANEXO 02, se lo realiza de la misma manera que el script del apagado del sistema o se lo puede automatizar como el script de VNC.