

DESARROLLO DE UN SISTEMA DEMÓTICO BÁSICO USANDO RASPEBRY PI, PHP Y PYTHON



Desarrollado Por: Jefferson Rivera Patiño.

Licenciado en Electrónica.

Desarrollador de aplicaciones en internet.

TWITTER: *@riverajefer*

WEB: <http://jeffersonrivera.com/>

BLOG: <http://riverajefer.blogspot.com/>

YOUTUBE: <https://www.youtube.com/user/riverajefer>

Contenido

PROYECTO DEMÓTICA BÁSICA CON RASPBERRY PI	3
1. INTRODUCCIÓN.....	3
2. OBEJTIVOS.....	3
3. CONOCIMIENTOS PREVIOS.....	3
4. DIAGRAMA ESQUEMÁTICO DEL PROYECTO.....	4
5. CONOCIENDO LA RASPBERRY PI	5
6. INSTALANDO RASPBIAN EN RASPBERRY PI.....	7
7. CONECTANDO CON LA RASPBERRY PI A TRAVÉS DE SSH	14
8. INSTALANDO Y CONFIGURANDO SERVIDOR WEB EN RASPBERRY PI	17
9. CONOCIENDO EL GPIO DE LA RASPBERRY PI	20
10. CONTROLANDO EL GPIO DE LA RASPBERRY PI CON PYTHON.....	22
11. ACCEDIENDO AL GPIO DE LA RASPBERRY PI DESDE PHP	23
12. CONFIGURANDO EL ROUTER, PARA TENER ACCESO DESDE CUALQUIER PUNTO.....	24
13. INSTALANDO Y CONFIGURANDO NO-IP EN LA RASPBERRY PI.....	25
14. APLICACIÓN WEB PARA CONTROLAR EL GPIO	26
15. PROBANDO EL SISTEMA COMPLETO.....	29
16. POSIBLES APLICACIONES	30
17. CONCLUSIONES Y RECOMENDACIONES.....	30
18. BIBLIOGRAFÍA	31
19. ANEXOS.....	32

PROYECTO DEMÓTICA BÁSICA CON RASPBERRY PI

1. INTRODUCCIÓN

En el presente documento se dará a conocer el procedimiento para desarrollar un proyecto de demótica básico, que consiste en controlar el estado de un led, remotamente por medio de una aplicación Web montada en la Raspberry Pi.

2. OBEJTIVOS

Objetivo General

Desarrollar un sistema de control remoto, para un led usando la Raspberry Pi y PHP

Objetivos Específicos

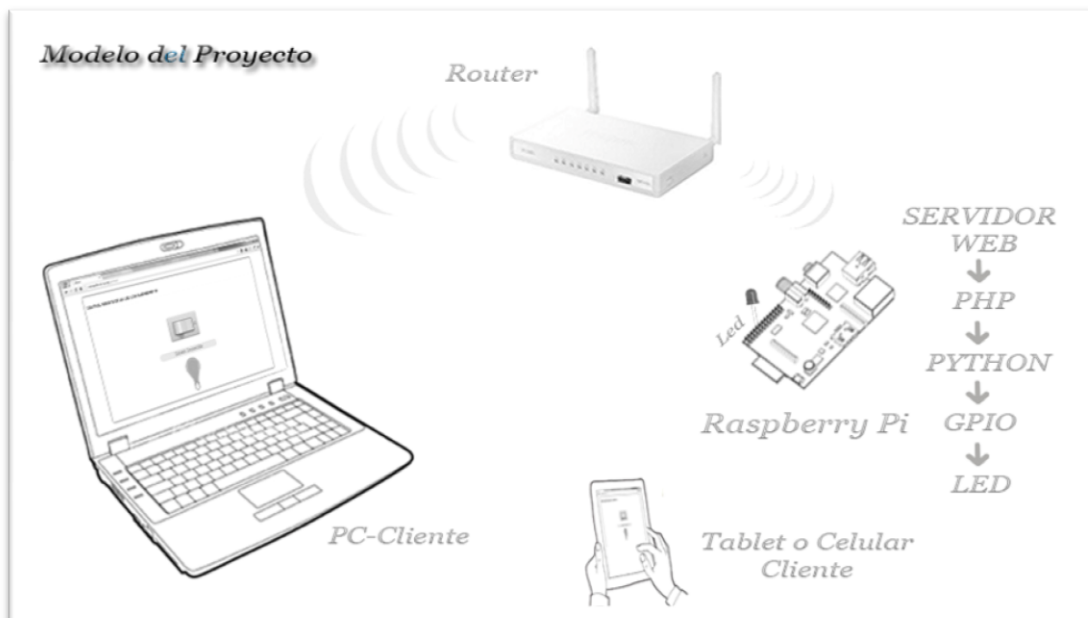
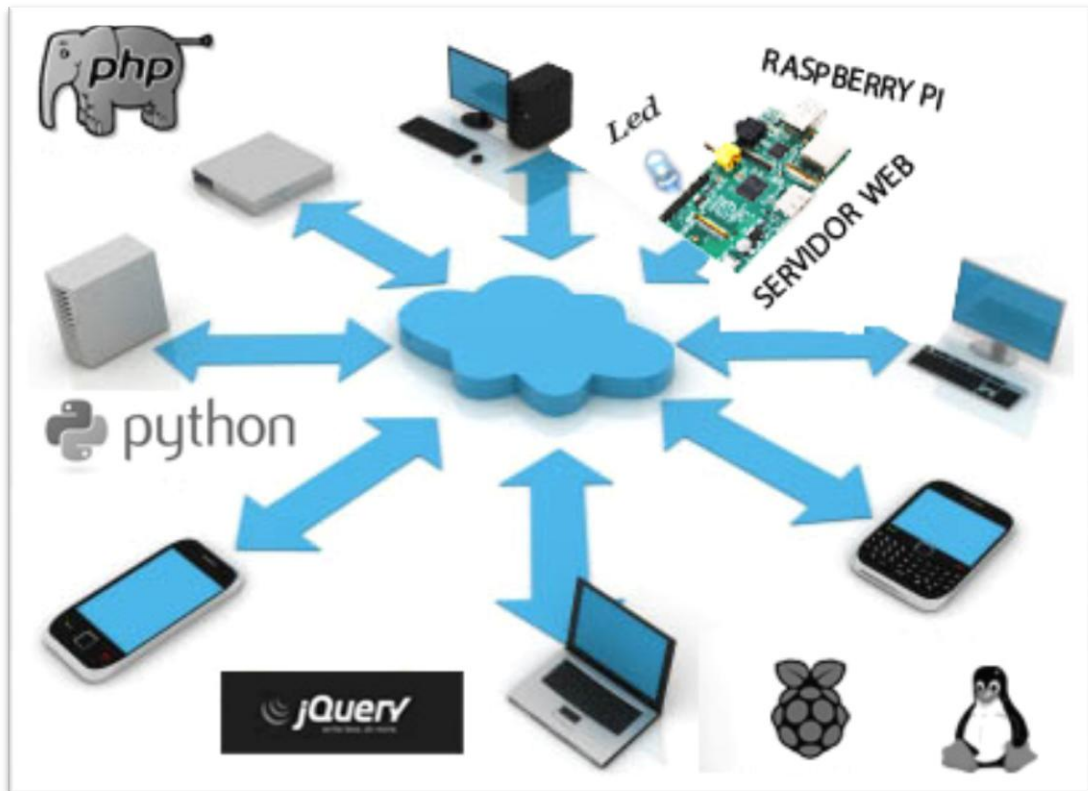
- Conocer las características técnicas de la Raspberry Pi Modelo B
- Instalar y configurar Raspbian en la Raspberry Pi
- Instalar y configurar un servidor web en Rasbian
- Controlar el puerto GPIO de la Raspberry Pi por medio de Python y PHP

3. CONOCIMIENTOS PREVIOS

Para que se te facilite el desarrollo y ejecución de proyecto, se requieren conocimientos previos:

- Redes de computadores.
- Comandos de Linux.
- Programación básica.
- HTML, PHP. Javascript.
- Python.
- Electrónica.

4. DIAGRAMA ESQUEMÁTICO DEL PROYECTO



Como se puede evidenciar en la imagen anterior, la Raspberry Pi hará de servidor, mientras que los demás dispositivos conectados a la nube de Internet, son los que corren la aplicación del lado del cliente, desarrollada en HTML y jQuery.

La cual consiste en un interruptor, o campo tipo checkbox, en donde cada vez que cambia de estado se envía una petición Ajax a PHP el cual ejecuta un script Python, que es el que controla el GPIO de la Raspberry Pi.

La siguiente imagen muestra un pantallazo de la aplicación del lado del cliente

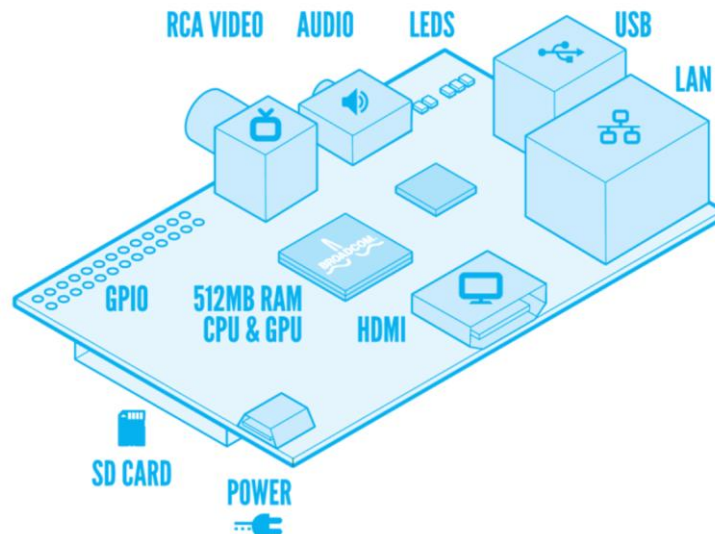


5. CONOCIENDO LA RASPBERRY PI

A continuación presentaremos una breve descripción sobre la tarjeta Raspberry Pi modelo B



RASPBERRY PI MODEL B

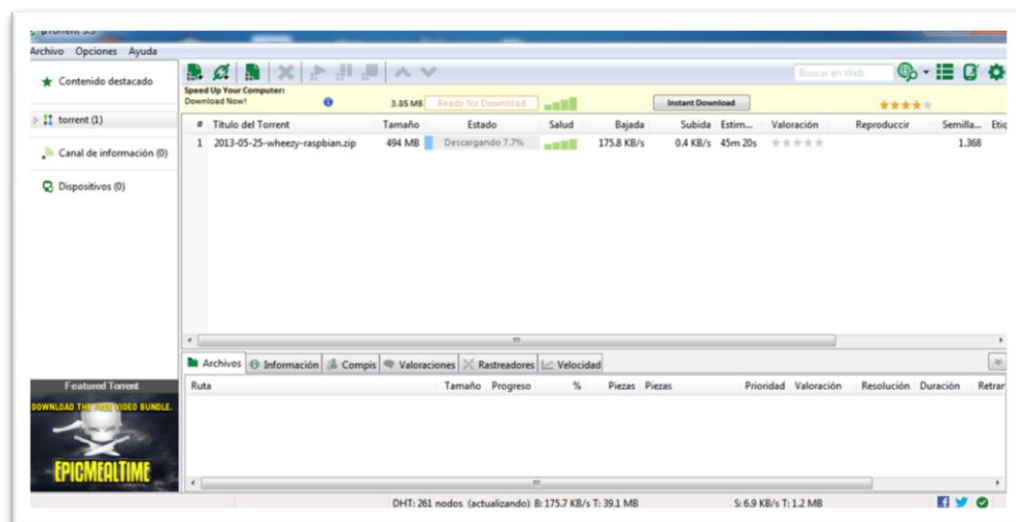
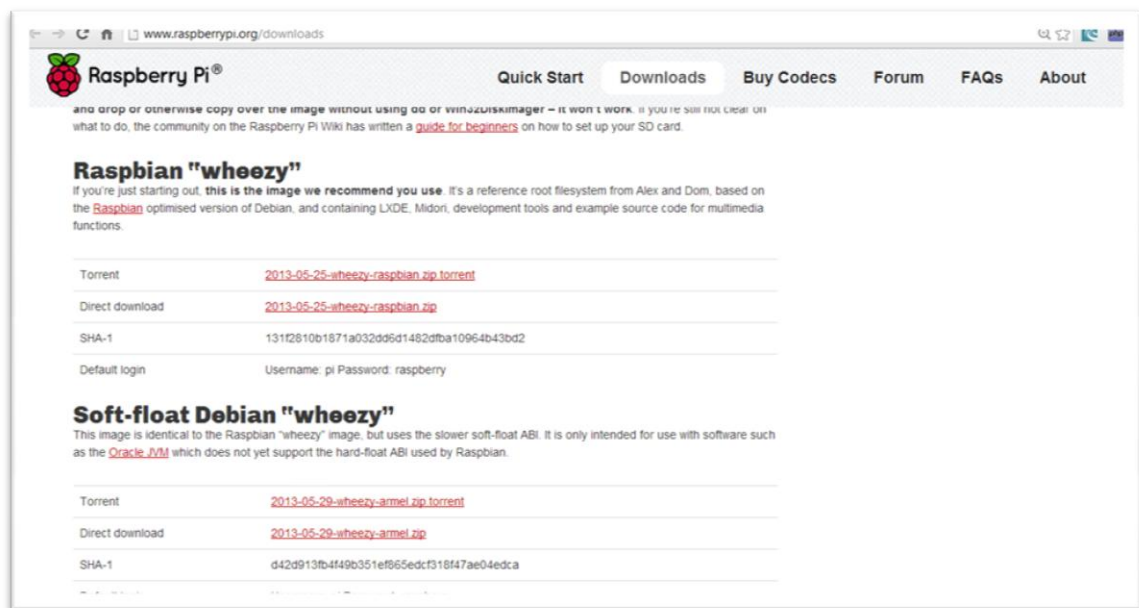


Especificaciones Técnicas

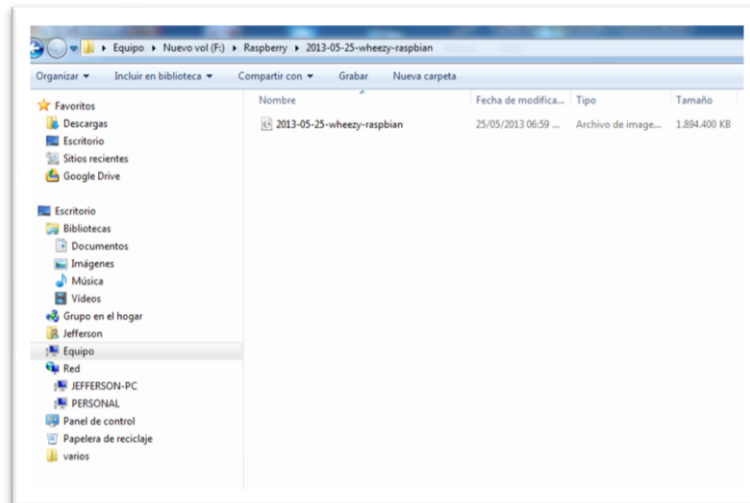
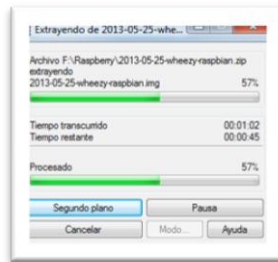
	Modelo A	Modelo B
Precio: ⁵	\$25	\$35
SoC: ⁵	Broadcom BCM2835 (CPU + GPU + DSP + SDRAM + puerto USB) ³	
CPU:	ARM1176JZF-S a 700 MHz (familia ARM11) ³	
GPU:	Broadcom VideoCore IV, ⁵⁹ OpenGL ES 2.0, -2 y VC-1 (con licencia), ⁵⁷ 1080p30 H.264/MPEG-4 AVC ³	
Memoria (SDRAM):	256 MB (compartidos con la GPU)	512 MB (compartidos con la GPU) ⁴ desde el 15 de octubre de 2012
Puertos USB 2.0: ⁵³	1	2 (vía hub USB integrado) ⁵²
Entradas de video: ⁶⁰	Conector [[MIPI] CSI que permite instalar un módulo de cámara desarrollado por la RPF	
Salidas de video: ⁵	Conector RCA (PAL y NTSC), HDMI (rev1.3 y 1.4), ⁶¹ Interfaz DSI para panel LCD ^{62 63}	
Salidas de audio: ⁵	Conector de 3.5 mm, HDMI	
Almacenamiento integrado:	SD / MMC / ranura para SDIO	
Conectividad de red: ⁵	Ninguna	10/100 Ethernet (RJ-45) via hub USB ⁵²
Periféricos de bajo nivel:	8 x GPIO, SPI, I ² C, UART ⁵⁹	
Reloj en tiempo real: ⁵	Ninguno	
Consumo energético:	500 mA, (2.5 W) ⁵	700 mA, (3.5 W)
Fuente de alimentación: ⁵	5 V vía Micro USB o GPIO header	
Dimensiones:	85.60mm × 53.98mm ⁶⁴ (3.370 × 2.125 inch)	
Sistemas operativos soportados:	GNU/Linux: Debian (Raspbian), Fedora (Pidora), Arch Linux (Arch Linux ARM), Slackware Linux. RISC OS ²	

6. INSTALANDO RASPBIAN EN RASPBERRY PI

Requisitos: Memoria SD mínimo 2GB



Descargamos la última actualización, que para esta fecha es la del 25 de mayo de 2013 y lo descomprimos, queda como imagen, la cual posteriormente se cargara en la tarjeta SD

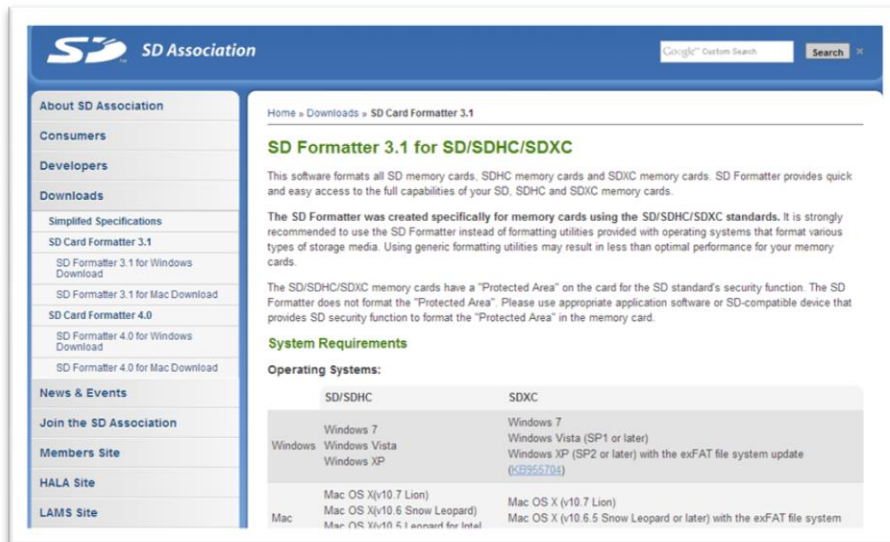


Preparamos la tarjeta SD para la instalación

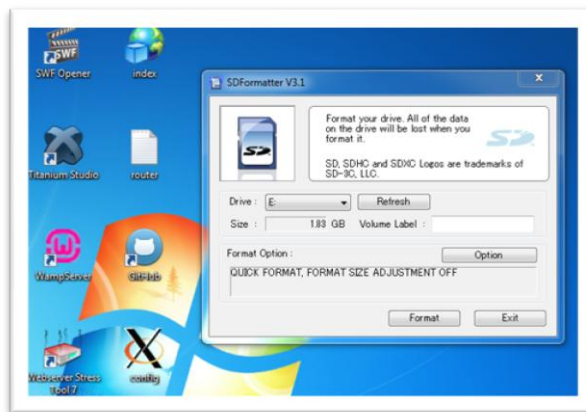


1. Descargamos e instalamos SD FORMATER

https://www.sdcard.org/downloads/formatter_3/



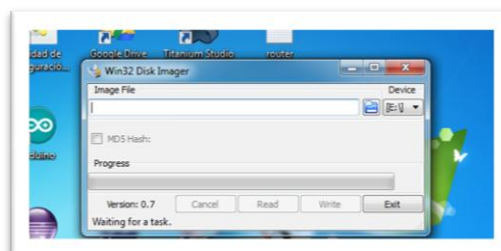
2. Formateamos la memoria con SD FORMATER



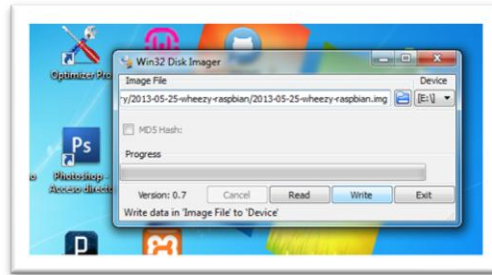
3. Descargamos e Instalamos Win32DiskImager

<http://sourceforge.net/projects/win32diskimager/>

Tarda un poco en abrir



4. Cargamos la imagen donde está Raspbian y le damos en Write

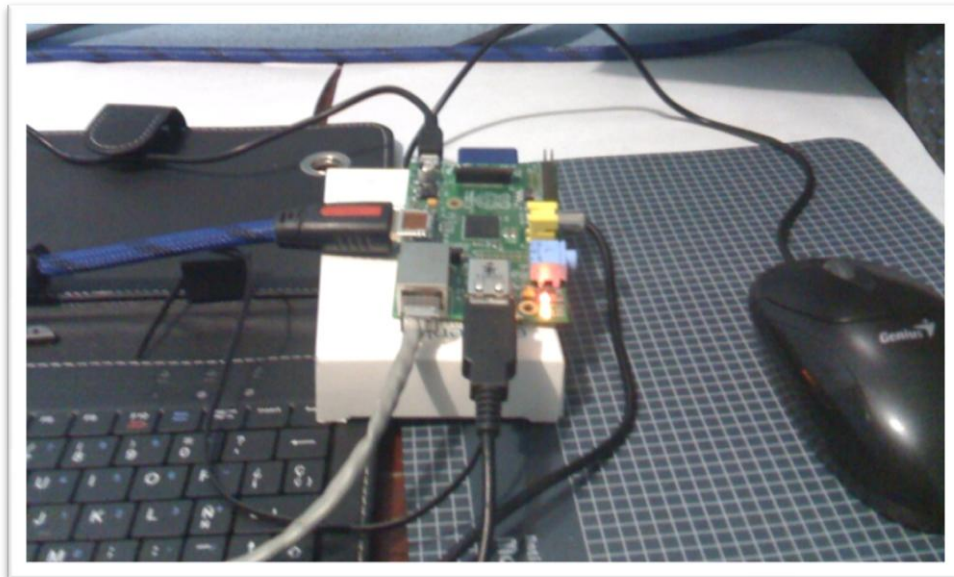


Se demora alrededor de 10 minutos copiando el sistema operativo en la memoria

Cuando haya terminado, retiramos la memoria y la introducimos en la tarjeta Raspberry PI



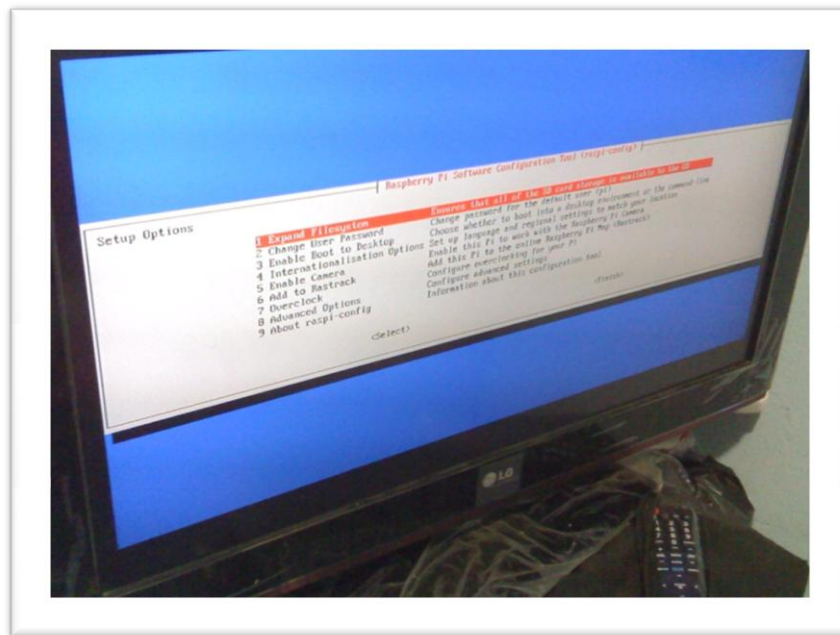
Luego conectamos la Raspberry Pi por HDMI a un TV o monitor, también conectamos un teclado y mouse usb, la fuente de voltaje 5 Voltios - 2 Amperios, y el cable de Ethernet.



Cargado controladores y demás archivos del Kernel

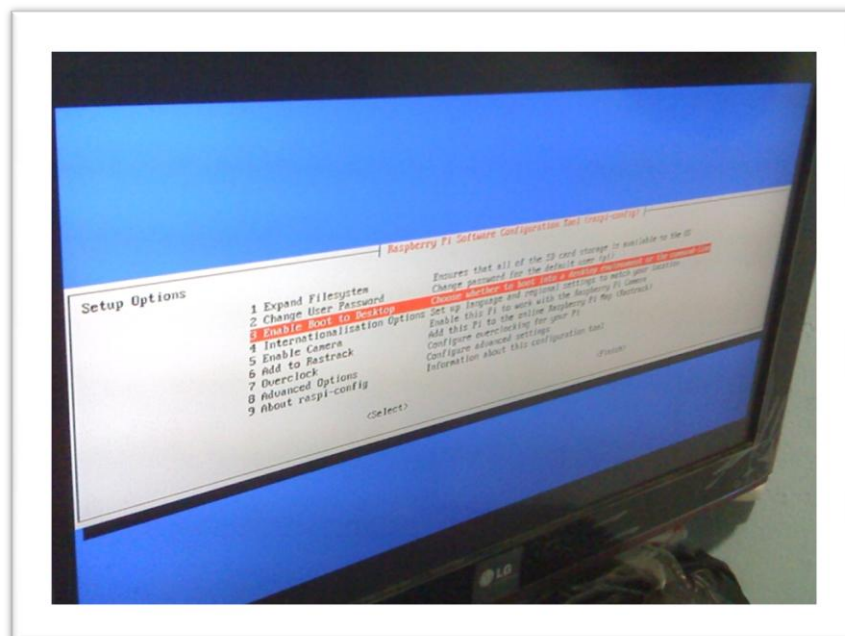


Configuración inicial



Damos en la primera opción que básicamente lo que hace es redimensionar la memoria SD, para darnos más espacio, para archivos y aplicaciones

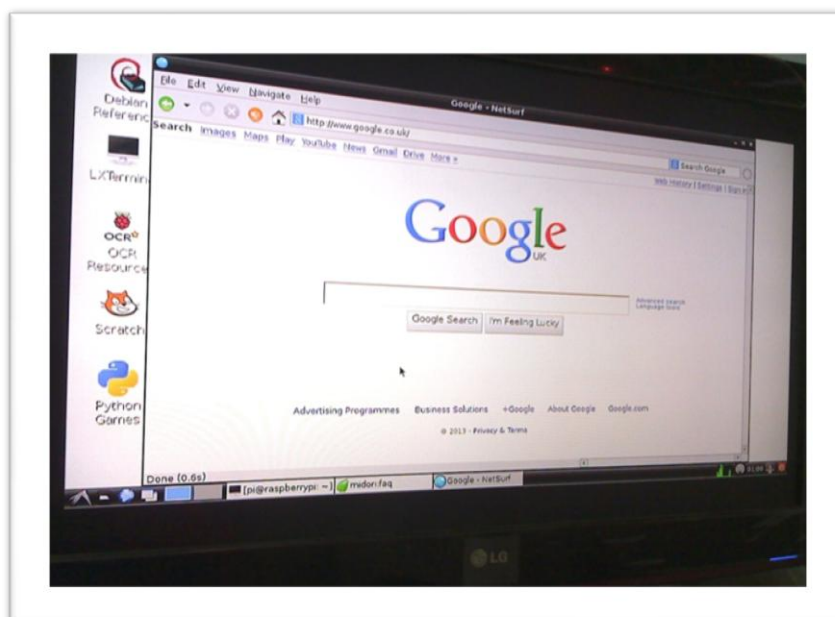
Ahora vamos a la opción 3, la cual nos habilita el inicio de escritorio, cada vez que arranquemos Raspbian.



Entorno grafico de Raspbian



Navegador Web de Raspbian



Abrimos la terminal y digitamos, lo siguientes comandos para actualizar los repositorios y el sistema, recuerda tener conectado el cable de internet a la Raspberry Pi

Sudo apt-get update.



Si tienes algún problema en la instalación de Raspbian, le recomiendo seguir , el siguiente vinculo, o también pueden googlear

<http://rasberryparatorpes.net/empezando/raspi-config-configuracion-inicial-de-raspbian/>

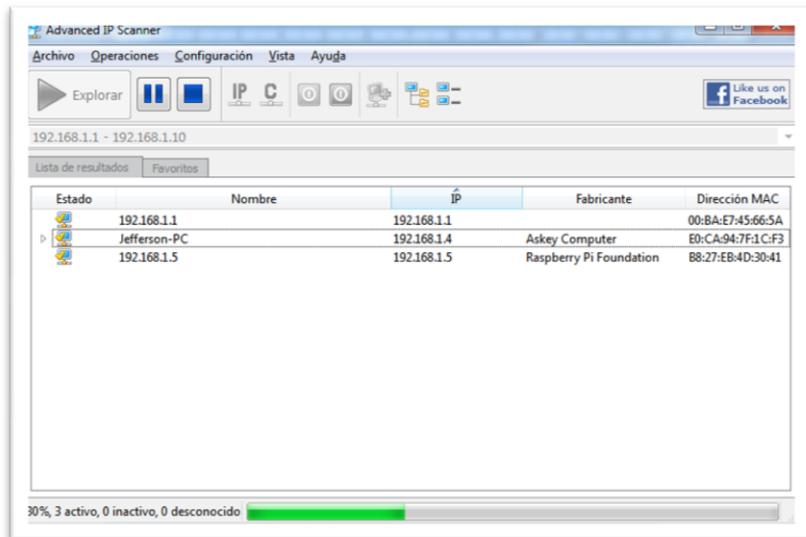
7. CONECTANDO CON LA RASPBERRY PI A TRAVÉS DE SSH

Poder controlar la Raspberry Pi, a través de SSH, nos facilita el trabajo, en el momento de instalar y configurar aplicaciones, sin necesidad de tener conectado un TV o monitor con entrada HDMI.

Para hacer esto primero identificamos la IP local, que tiene asignada la Raspberry Pi, esto lo podemos hacer usando un pequeño programa llamado Advance IP Sacan,

Se puede descargar desde:

<http://www.advanced-ip-scanner.com/es/>



Como se puede ver en la tala, la ip asignada a la Raspberry es la 192.168.1.5

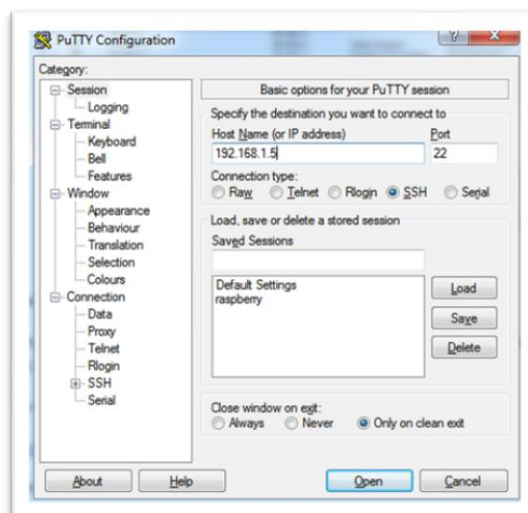
Ahora descargamos el programa PuTTY, que nos servirá para establecer una conexión con la Raspberry Pi, a través de SSH.

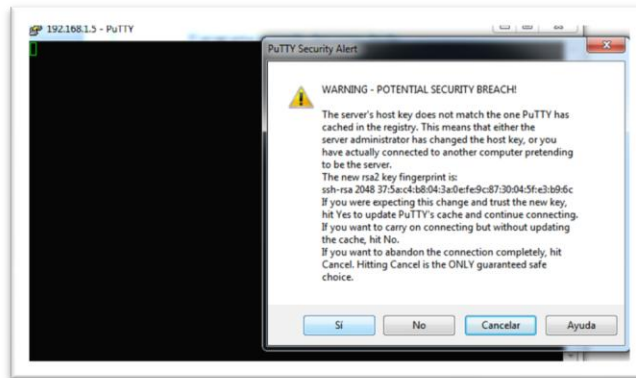
El programa se puede descargar desde:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

Descargamos e instalamos el programa.

Abrimos la conexión

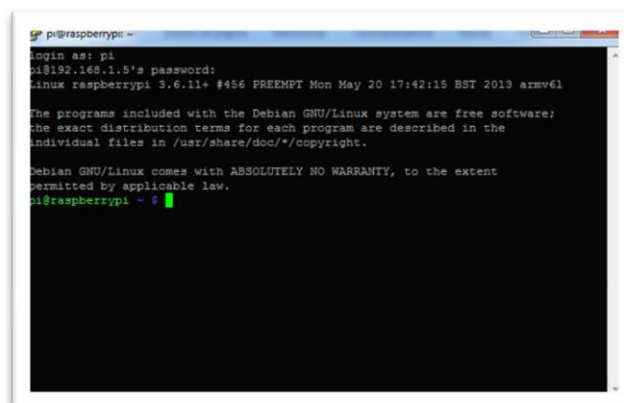
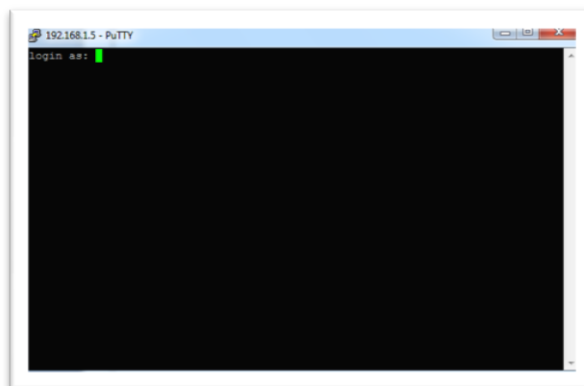




Damos en Si.

Usuario: pi

Password: raspberry



Desde aquí se puede trabajar en el Raspbian, sin necesidad de estar conectado a un TV

En el siguiente vínculo pueden consultar algunos comandos importantes usados en Linux

<http://www.esdebian.org/wiki/lista-comandos-gnulinux-i>

También pueden ver los videos de Youtube donde se explica más acerca de la conexión remota con Raspberry PI, incluso como establecer conexión remota de escritorio. Son 3 videos

<http://www.youtube.com/watch?v=NsfAGC4en6s>

8. INSTALANDO Y CONFIGURANDO SERVIDOR WEB EN RASPBERRY PI

Ahora vamos a instalar y configurar servidor APACHE, con PHP y MSQL, esto usando la conexión remota SSH

Primero

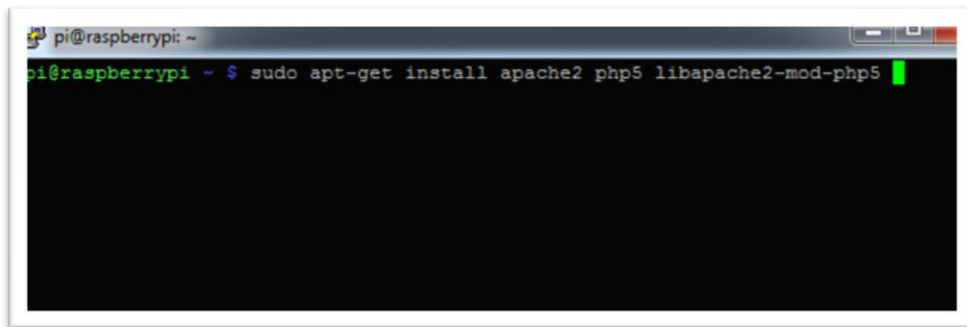
```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Luego

Instalar Apache y PHP

```
sudo apt-get install apache2 php5 libapache2-mod-php5
```



Reiniciamos apache

```
sudo /etc/init.d/apache2 restart
```



Otorgamiento de permisos

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. Vamos allá¹

Cambiamos el usuario y grupo al directorio

¹ Tomado de <http://www.diverteka.com/?p=1136>

```
sudo chown www-data:www-data /var/www/
```

Damos los permisos a la carpeta www/

```
sudo chmod 775 /var/www/
```

Añadimos el usuario pi al grupo www-data

```
sudo usermod -a -G www-data pi
```

```
sudo visudo
```

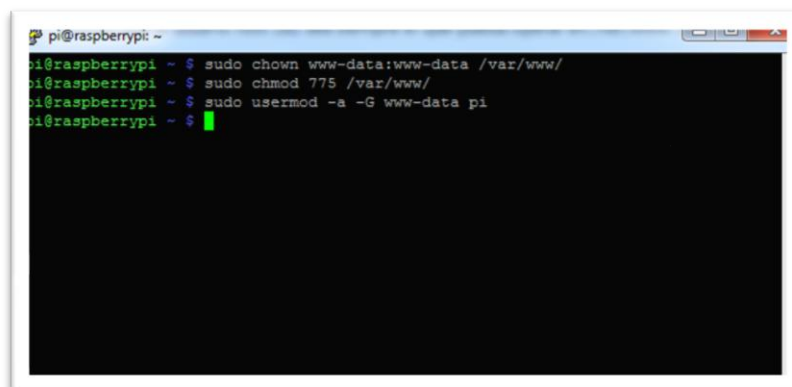
Agregamos el siguiente código al final.

```
www-data ALL=(root) NOPASSWD:ALL
```

Reiniciamos el Servidor

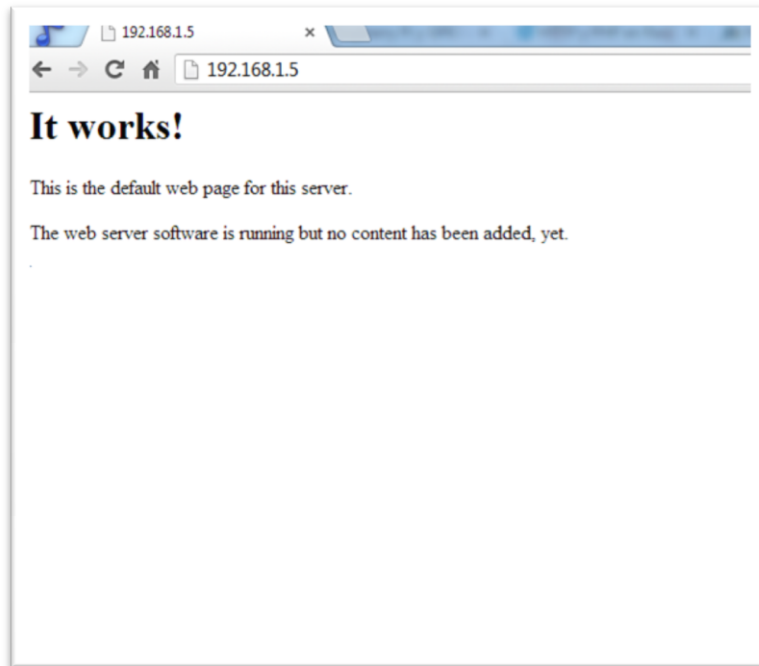
```
sudo /etc/init.d/apache2 restart
```

http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m1/permisos_de_archivos_y_carpetas.html buen post sobre permisos



Prueba desde mi navegador Web. Colocando en la URL la IP de la Raspberry PI

<http://192.168.1.5/>

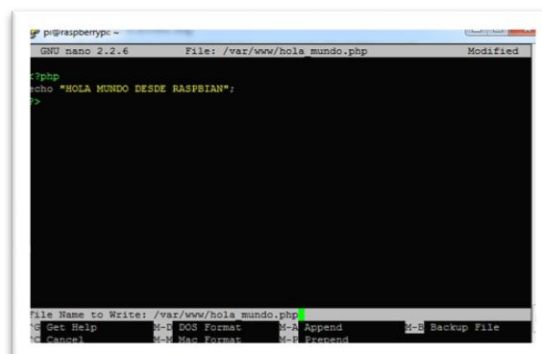


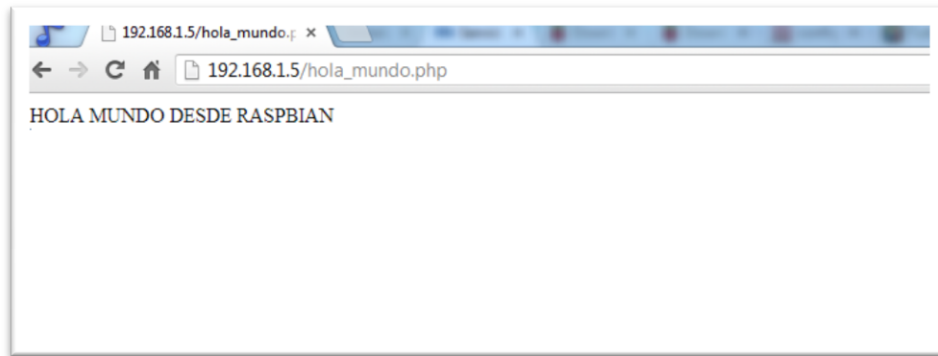
Creando un archivo de prueba, Hola mundo en PHP

```
sudo nano /var/www/hola_mundo.php
```



```
<?php  
echo "HOLA MUNDO DESDE RASPBIAN";  
?>
```






9. CONOCIENDO EL GPIO DE LA RASPBERRY PI

La Raspberry Pi, además de contar con puertos USB, HDMI, ETHERNET, tiene a su disposición una serie de pines de propósito general, entre los cuales está entradas y salidas digitales, UART, etc

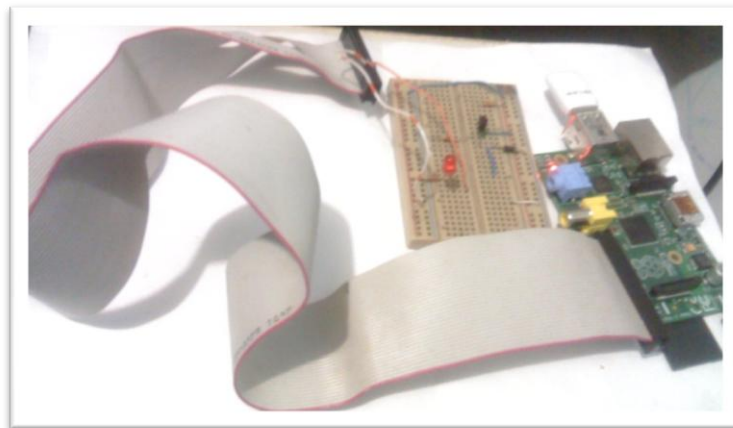
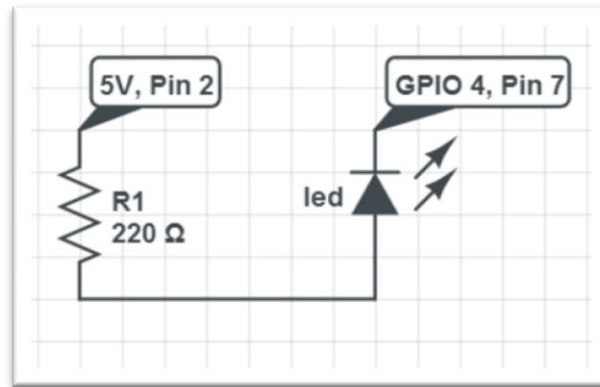
3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2C0 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

Pin 1	Pin 2	Pin 1	Pin 2	Pin 1	Pin 2	
3.3 V	5 V	3.3 V	5 V			
GPI0 0	5 V	GPI0 2	5 V			
GPI0 1	GND	GPI0 3	GND			
GPI0 4	GPI0 14	GPI0 4	GPI0 14			
GND	GPI0 15	GND	GPI0 15			
GPI0 17	GPI0 18	GPI0 17	GPI0 18			
GPI0 21	GND	GPI0 27	GND			
GPI0 22	GPI0 23	GPI0 22	GPI0 23			
3.3 V	GPI0 24	3.3 V	GPI0 24			
GPI0 10	GND	GPI0 10	GND			
GPI0 9	GPI0 25	GPI0 9	GPI0 25			
GPI0 11	GPI0 8	GPI0 11	GPI0 8			
GND	GPI0 7	GND	GPI0 7			
Pin 25	Pin 26	Pin 25	Pin 26		Pin 25	Pin 26
<div> Raspberry Pi (Rev. 1) </div>					<div> Raspberry Pi (Rev. 2) </div>	

http://elinux.org/RPi_Low-level_peripherals

- **Ficheros de Control de Interface.** Inicialmente **GPIO** está sólo disponible en el **espacio del núcleo** y para poder usarlo, tenemos que pedir al núcleo que nos exporte al espacio de usuario el pin que queremos usar. Por lo tanto los ficheros de control nos sirven para pedir al núcleo el uso de cierto GPIO y también para liberar el mismo una vez hayamos terminado.
- **Los propios GPIOs.** Estos aparecen en forma de directorios. Veremos más adelante que contienen.
- **Controlador GPIO.** Este es también otro directorio con ciertos ficheros que proveen información sobre el chip GPIO de la placa.

Para trabajar con el puerto GPIO, podemos usar un cable de los que se usaban, para conectar la unidad de disquete de un PC, en un extremo conectamos el GPIO y en el otro extremo sacamos dos cables, uno sale del pin de 5V y va en serie con una resistencia de 220Ω y está a su vez va al ánodo del diodo led, y el otro del pin 7 GPIO4 va al cátodo del diodo led, como muestra el siguiente diagrama.



Controlando el led, desde la línea de comandos

Un ejemplo de uso sería este que pone el pin gpio 4 del bcm2835 en modo salida y luego le da el valor de 1(HIGH):

```
sudo -i  
echo "4" > /sys/class/gpio/export  
echo "out" > /sys/class/gpio/gpio4/direction  
echo "1" > /sys/class/gpio/gpio4/value
```

10. CONTROLANDO EL GPIO DE LA RASPBERRY PI CON PYTHON

Python nos proporciona una librería con la que podemos interactuar con los pines del puerto GPIO, para trabajar con esta librería, primero debemos instalarla.

A continuación, se muestra el procedimiento que se debe llevar para su correcta instalación.

```
$ wget http://pypi.python.org/packages/source/R/RPi.GPIO/RPi.GPIO-0.1.0.tar.gz
```



```
$ tar xzf RPi.GPIO-0.1.0.tar.gz
$ cd RPi.GPIO-0.1.0
$ sudo python setup.py install
```

Después de haber instalado la librería, podemos hacer una prueba de funcionamiento. En este caso controlando el GPIO4. Que se encuentra en el pin 7

```
$ sudo python
>>> import RPi.GPIO as GPIO
>>> GPIO.setup(7, GPIO.OUT)
>>> GPIO.output(7, False)
```

En el siguiente vínculo se explica cómo instalar y configurar un servidor web, que controla el GPIO, en este caso con Python y Javascript

<https://code.google.com/p/webiopi/> **GPIO WEB**

Librerías desarrolladas a partir del driver del kernel por usuarios:

<http://muyraspi.blogspot.com/2013/02/programar-gpio.html>

11. ACCEDIENDO AL GPIO DE LA RASPBERRY PI DESDE PHP

En PHP tenemos una función llamada [exec](#) la cual ejecuta un programa externo.

Ejemplo de uso.

```
<?php
echo exec('programa');
?>
```

Para nuestro caso, queremos ejecutar un script de Python

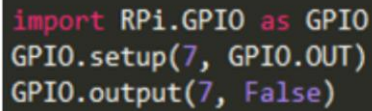
```
<?php
$a=exec('sudo python /var/www/pi/leds/prende.py');
echo $a;
?>
```

```
<?php
$a=exec('sudo python /var/www/pi/leds/prende.py');
echo $a;
?>
```

El script `prende.py` contiene:

```
import RPi.GPIO as GPIO
```

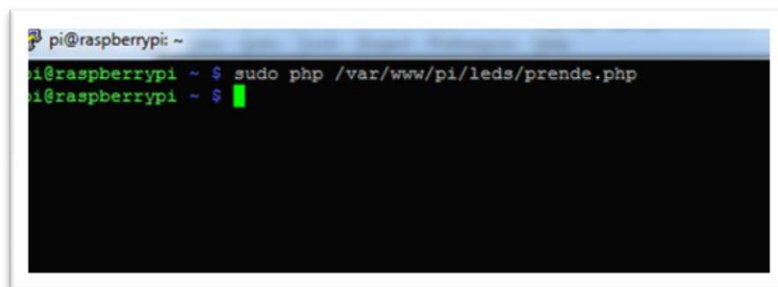
```
GPIO.setup(7, GPIO.OUT)
GPIO.output(7, False)
```



```
import RPi.GPIO as GPIO
GPIO.setup(7, GPIO.OUT)
GPIO.output(7, False)
```

Ahora lo podemos probar ejecutándolo por consola

```
$ sudo php /var/www/pi/leds/prende.php
```



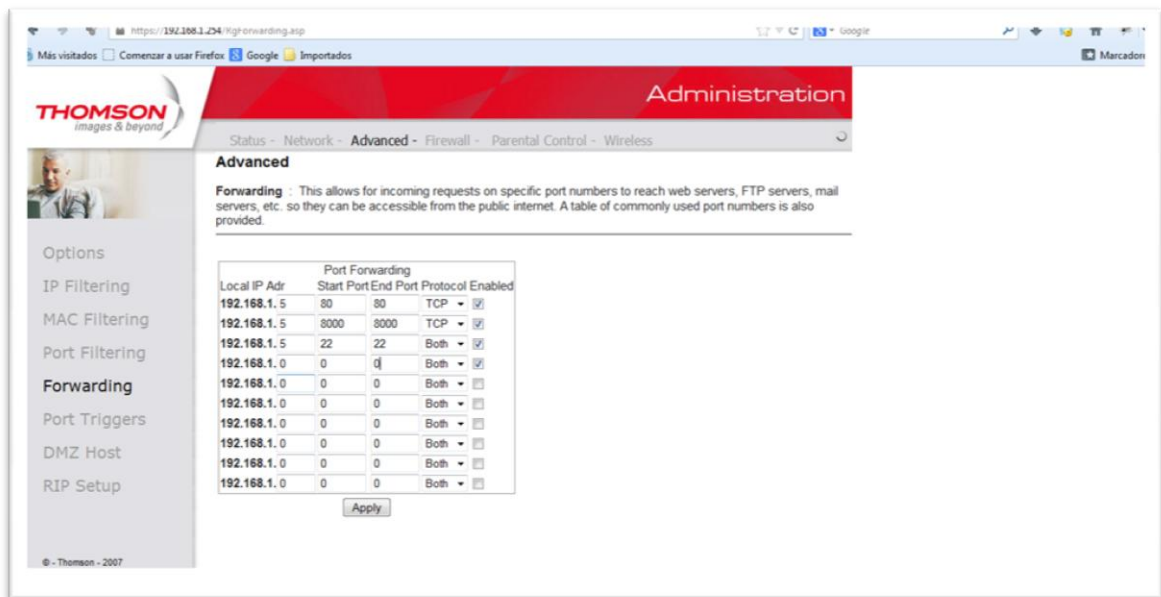
12. CONFIGURANDO EL ROUTER, PARA TENER ACCESO DESDE CUALQUIER PUNTO

Este punto es importante, para tener acceso a la Raspberry Pi, desde cualquier equipo que esté conectado a internet.

Entonces generalmente nuestro proveedor de servicios de Internet, nos deja inactivos o bloqueados los puertos del router, y por ejemplo si no está habilitado el puerto 80, no vamos a poder acceder a nuestras aplicaciones montadas en el servidor apache.

Entonces lo que debemos hacer, es abrir estos puertos, y apuntarlos a la IP local que tenga asignada nuestra Raspberry Pi.

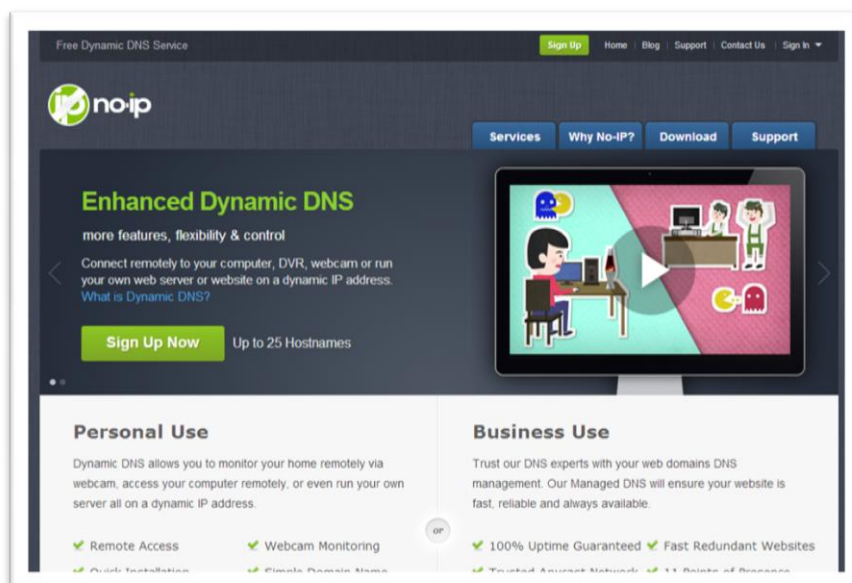
Para abrir los puertos necesitamos tener acceso a nuestro Router, para esto, debemos primero conseguir los datos de acceso, esto lo pueden hacer, buscando por Internet, con la marca del Router y el proveedor de servicio de internet que tengan.



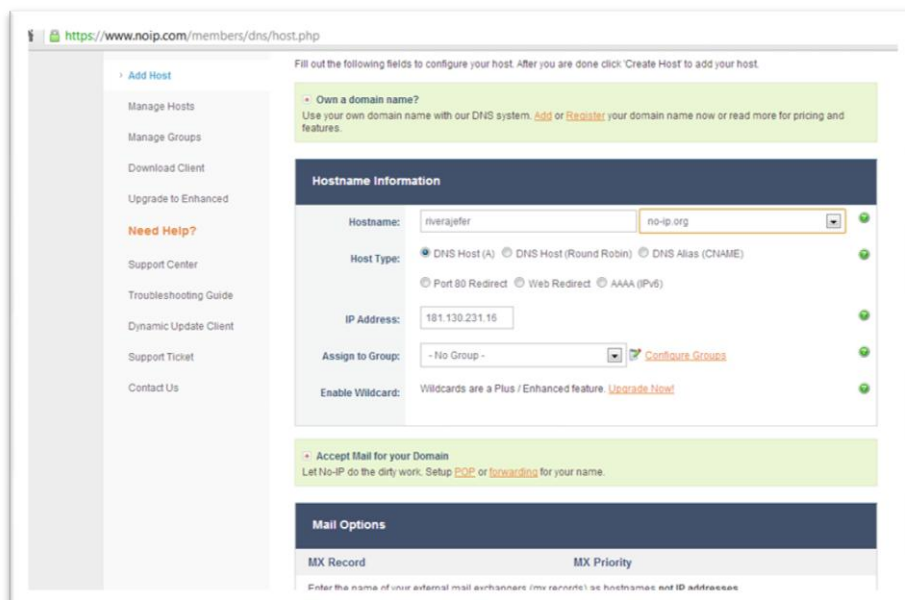
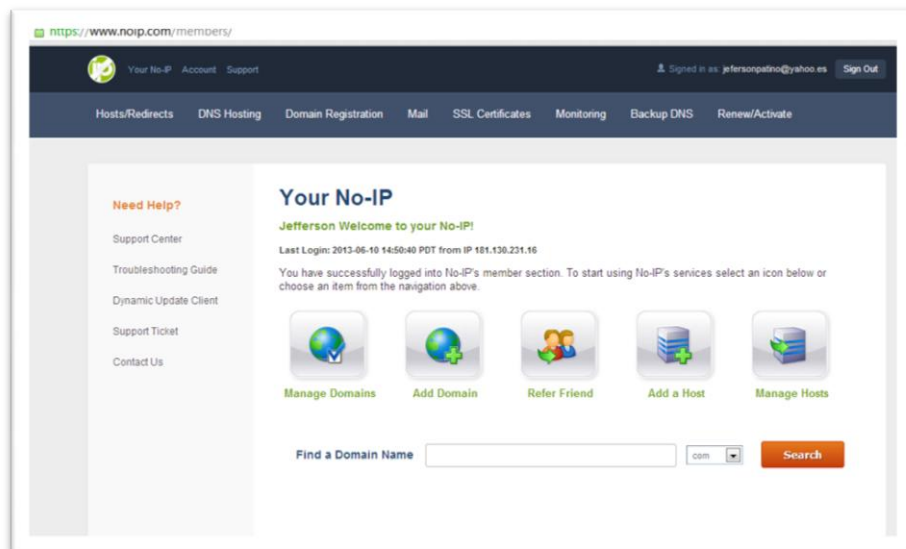
13. INSTALANDO Y CONFIGURANDO NO-IP EN LA RASPBERRY PI

Esta aplicación es de vital importancia, porque nos permite asignar un nombre de dominio a nuestra IP pública. Es una herramienta potente ya que para cargar nuestra aplicación Web no necesitamos colocar en la URL la dirección IP, si no solo el nombre de dominio, además de que no importa si está cambia, y lo mejor de todo es que es completamente gratis.

<http://www.noip.com/>



Creamos una cuenta,
importante recordar los datos, porque estos se solicitaran al momento de sincronizar
NO-IP en Raspbian



Para la instalación de NO-IP se recomienda seguir

<http://raspberrypihelp.net/tutorials/29-raspberry-pi-no-ip-tutorial>

14. APLICACIÓN WEB PARA CONTROLAR EL GPIO

La aplicación es muy básica, contiene los siguientes archivos.

Nombre	Fecha de modifica...	Tipo	Tamaño
css	23/06/2013 01:33 ...	Carpeta de archivos	
images	23/06/2013 01:27 ...	Carpeta de archivos	
js	23/06/2013 01:38 ...	Carpeta de archivos	
procesa	19/06/2013 09:57 ...	Archivo PHP	1 KB
apaga.py	19/06/2013 10:29 ...	Archivo PY	1 KB
prende.py	19/06/2013 10:29 ...	Archivo PY	1 KB
index	23/06/2013 01:38 ...	Chrome HTML Do...	2 KB

Carpeta css -> Contiene los archivos de estilos, entre estos, el que modelo el interruptor

Carpeta js -> Contiene el script de jQuery 1.9

Carpeta images -> Contiene la imagen de fondo

Index.html ->

Parte 1.

```

1 <!DOCTYPE html>
2 <html lang="es">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Boton</title>
8     <script src="js/jquery-1.9.1.js"></script>
9     <link rel="stylesheet" type="text/css" href="css/style.css" />
10    <link rel="stylesheet" type="text/css" href="css/estilo.css" />
11    <link href="http://fonts.googleapis.com/css?family=Open+Sans+Condensed:700,300" rel="stylesheet" type="text/css" />
12  </head>
13  <body>
14    <div class="container">
15      <header>
16        <h1><strong>CONTROL REMOTO </strong> DE UN LED CON RASPBERRY PI</h1>
17      </header>
18      <section class="main">
19        <br><br>
20        <input type="checkbox" id="switch" name="switch">
21        <label for="switch" class="switch"></label>
22        <div align="center" id="bombillo">
23        </div>
24      </section>
25    </div>
26  </body>

```

Parte2.

```

25 </div>
26
27
28 <script>
29     $("#switch").change(function()
30     {
31         var checkeado=$("#switch").attr("checked");
32         if(checkeado) {
33             $("#switch").removeAttr("checked");
34             console.log("estado 1");
35             var estado=1;
36             $("#bombligo").text("Estado Apagado");
37             $("#bombligo").css('background-color','#585858');
38
39             $.ajax({
40                 data:{valor_estado: estado},
41                 url:'procesa.php',
42                 type:'POST',
43                 success: function(response){
44                     //alert("Salida: "+response );
45                 }
46             });
47         } else {
48             $("#switch").attr("checked","checked");
49             console.log("estado 2");
50             var estado=2;
51             $("#bombligo").text("Estado Encendido");
52             $("#bombligo").css('background-color','#0083ca');
53             $.ajax({
54                 data:{valor_estado: estado},
55                 url:'procesa.php',
56                 type:'POST',
57                 success: function(response){
58                     //alert("Salida: "+response );
59                 }
60             });
61         }
62     });
63 </script>
64 </body>
65 </html>

```

Procesa.php->

```

1 <?php
2 $valor_estado=$_POST['valor_estado'];
3 if($valor_estado==1)
4 {
5     exec('sudo python /var/www/raspberry/leds/apaga.py');
6 }
7 else{
8     exec('sudo python /var/www/raspberry/leds/prende.py');
9 }
10 ?>

```

En la línea 2, recibimos la variable valor_estado, por el método POST, y la guardamos en \$valor_estado, luego preguntamos si es igual a 1, si es cierto, ejecutamos apaga.py, si no ejecutamos prende.py

prende.py->

```

1 import RPi.GPIO as GPIO
2 GPIO.setup(7, GPIO.OUT)
3 GPIO.output(7, False)
4

```

apaga.py->

```
1 import RPi.GPIO as GPIO
2 GPIO.setup(7, GPIO.OUT)
3 GPIO.output(7, True)
4
```

15. PROBANDO EL SISTEMA COMPLETO

En el siguiente vínculo se puede descargar la aplicación desarrollada

<http://jeffersonrivera.com/pi/led.zip>





16. POSIBLES APLICACIONES

Las aplicaciones son diversas.

- Aplicaciones de domótica.
- Apertura y cierre de puertas y ventanas remotamente.
- Monitoreo de estructuras.
- Sistemas de seguridad, control de cámaras remotamente.

17. CONCLUSIONES Y RECOMENDACIONES

Se esté desarrollo se puede concluir:

Se puede desarrollar un sistema domótico, con bajos recursos económicos y materiales.

Es importante contar con una alimentación de buen amperaje, para garantizar un buen funcionamiento de los dispositivos conectados a la Raspberry Pi.

Continua.....

18. BIBLIOGRAFÍA

<http://www.raspberrypi.org>

<http://commandcat.blogspot.com.es/2013/02/raspberry-pi-como-crear-una-sd-de.html>

http://elinux.org/RPi_Low-level_peripherals

<http://www.cyberhades.com/2013/01/18/raspberry-pi-interactuando-con-gpio-desde-la-linea-de-comandos/>

<http://www.esdebian.org/wiki/lista-comandos-gnulinix-i>

<https://www.noip.com>

<http://rasberryparatorpes.net>

<http://rasberrypihelp.net/tutorials/29-raspberry-pi-no-ip-tutorial>

<http://www.diverteka.com/?p=1370>

<http://www.youtube.com/watch?v=NsFAGC4en6s>

http://es.wikipedia.org/wiki/Arquitectura_ARM

http://es.wikipedia.org/wiki/Raspberry_Pi

<http://openmicros.org/index.php/articles/94-ciseco-product-documentation/raspberry-pi/217-getting-started-with-raspberry-pi-gpio-and-python>

19. ANEXOS

En el siguiente vínculo se puede descargar la aplicación desarrollada

<http://jeffersonrivera.com/pi/led.zip>

Desarrollado Por: Jefferson Rivera Patiño

Licenciado en Electrónica.

Desarrollador de aplicaciones en internet

TWITTER: *@riverajefer*

WEB: <http://jeffersonrivera.com/>

BLOG: <http://riverajefer.blogspot.com/>

YOUTUBE: <https://www.youtube.com/user/riverajefer>