

2015

Automatización de un Servidor de Descargas P2P

Grado Superior en administración de sistemas informáticos en red



Índice

Índice	1
Introducción y Justificación	2
Objetivos	2
Análisis del entorno	2
Desarrollo del Contenido	3
Mejoras	19
Conclusión y valoración personal	22
Biografía y fuentes de consultas	23
Fecha de terminación de proyecto	24
Anexo	25

Introducción y Justificación

En este proyecto intentamos crear una pequeña aplicación que permita la instalación y configuración de un servidor de descargas P2P. Hay que tener en cuenta que esta pequeña aplicación será creada para aquellas personas que no requieran de altos conocimientos de informática, así teniendo detalles en la aplicación para que el usuario sepa que hacer en todo momento y sea sencillo su utilización.

Objetivos

El objetivo principal de este proyecto es crear un script que automatice la instalación de un servidor de descargas utilizando los servicios Apache, OpenSSH, Pydio, SAMBA, NO-IP y un servicio de mensajería. Apache será la base de todo, OpenSSH lo utilizaremos para poder conectarnos remotamente a nuestra Pi, Pydio nuestra nube privada, SAMBA para compartir las descargas, NO-IP para conectarnos a nuestra Pi desde la calle y el servicio de mensajería para que nos avise de que la descarga ha sido completada.

Análisis del entorno

Tras investigar en la web, no he dado con ningún *SO* o aplicación completa para realizar un servidor de descargas p2p sobre una Raspberry PI, o al menos tan completa como la que se ha desarrollado.

El usuario medio de internet, tiene un conocimiento limitado y muy ligado al entorno gráfico, con lo cual encuentra ciertos hándicaps cuando busca una configuración más profunda de determinados servicios y aplicaciones, es por esto que se ha buscado la mayor simplicidad para la instalación, configuración y administración de los distintos servicios y aplicaciones aun siendo todo esto a través de un Shell de Linux.

Desarrollo del Contenido

Antes de empezar con la explicación del proyecto tenemos que tener ciertos conocimientos de los servicios utilizados en este proyecto, para ello daré una pequeña explicación de cada uno de ellos:

LAMP:



El término hace referencia al sistema creado por la conjunción de esas aplicaciones libres. Este grupo de aplicaciones generalmente son usados para crear servidores web.

LAMP provee a los desarrolladores con los cuatro elementos necesarios para un servidor web, LAMP hace referencia a Linux, Apache, MySQL y PHP/Perl/Python.

OpenSSH:



OpenSSH es un conjunto de herramientas de conectividad que se usan para acceder a sistemas remotos de forma segura.

Pydio:



Pydio es una plataforma libre donde compartir archivos de forma privada en el ámbito empresarial. La finalidad es ofrecer una plataforma de intercambio de archivos privada controlada por la empresa al mando a la que se pueda acceder desde cualquier parte, ya sea vía interfaz web o aplicación para smartphone.

SAMBA:



Samba son un conjunto de aplicaciones para Linux, que implementan el protocolo de comunicación SMB utilizado por los sistemas operativos Microsoft Windows para compartir carpetas e impresoras. Básicamente samba permite a PCs que utilizan Linux, conectarse a carpetas compartidas en PCs con Windows y compartir carpetas como si de un sistema Windows se tratara. Gracias a samba, en una red podemos tener PCs con Windows y PCs con Linux de forma que puedan intercambiar información en carpetas compartidas de la misma forma que se haría si todos los PCs fueran Windows.

NO-IP:



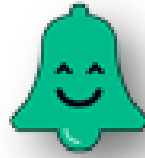
No-IP es un servicio gratuito que gestiona un dominio a una dirección IP dinámica. Si tienes un dominio pero no tienes una IP fija, si no que tienes una dirección IP dinámica, tendrías que andar modificando constantemente tu DNS de tu dominio a la IP que tengas en cada momento. Gracias a este servicio tendrás un nombre de dominio gratis.

Transmission:



Transmission es un cliente P2P liviano, gratuito y de código abierto para la descarga de torrents con una interfaz de usuario concisa, debido al rendimiento estable y a la compatibilidad para varias plataformas, se ha ganado el aprecio de los usuarios.

Pushetta:



Pushetta es un sistema hecho para notificaciones push en tiempo real a muchos dispositivos diferentes (teléfonos móviles, navegadores, TV inteligente,...).

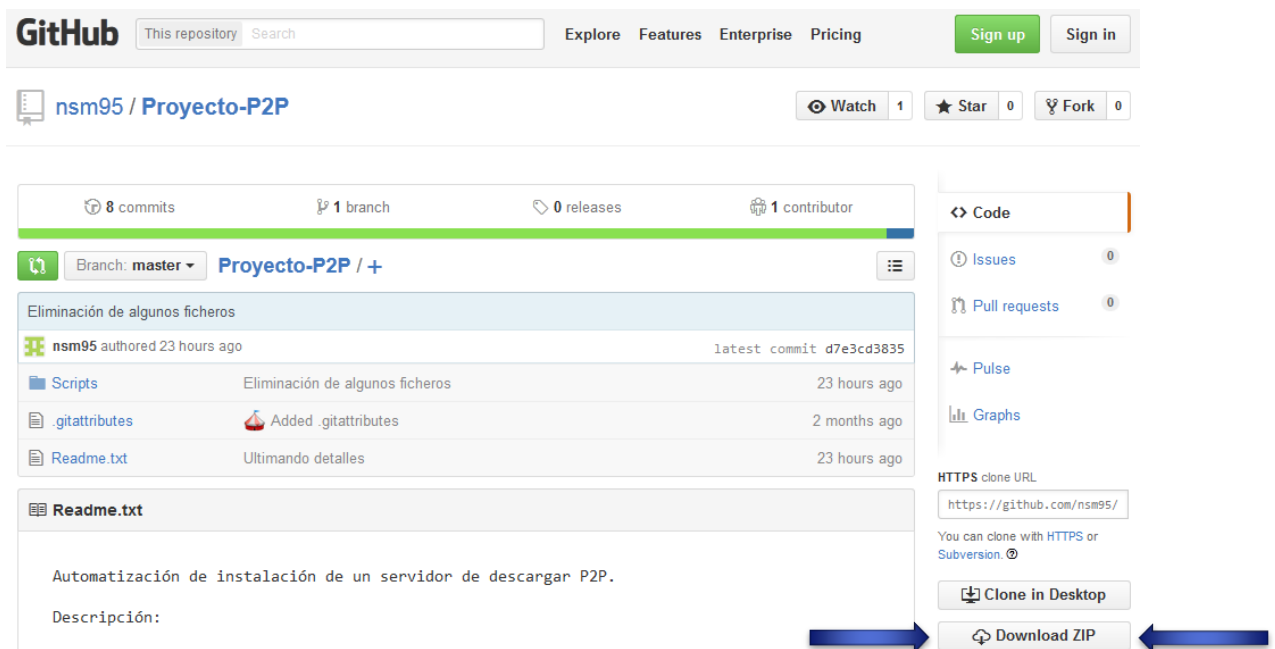
Es gratuita, no hay necesidades de los números de teléfono u otros datos personales, remitente obtiene estadísticas precisas sobre los abonados y los mensajes entregados, se puede ampliar para soportar dispositivos que no sean teléfonos (es decir. Smart TV, navegadores web, etc...).

Una vez explicado los servicios que consta la aplicación pasaremos a la explicación del funcionamiento de la aplicación:

¿Cómo podemos obtener esta aplicación? Pues de momento la podremos descargar de nuestro espacio en GitHub en el siguiente enlace, <https://github.com/nsm95/Proyecto-P2P> donde además podréis contribuir en la aplicación y si tenéis alguna idea poder mejorarla.

Requisitos para poder instalar por completo la aplicación, tener como usuario principal “pi”, tener cuenta en NO-IP, PYDIO y tener descargada la aplicación Pushetta en su Smartphone para recibir las notificaciones.

Accedemos a GitHub con el enlace mencionado anteriormente y abajo a la derecha pulsaremos Download ZIP.



Una vez descargada la aplicación en nuestra Raspberry Pi, lo primero que tenemos que hacer es descomprimir el paquete zip, para ello ejecutamos el comando `unzip + el nombre del archivo`, nos aparecerá una carpeta con los scripts, accedemos a ella y seguimos los pasos del documento `README.txt` para poder utilizar la aplicación.

```
1.- Ejecutar el script permisos de esta manera:
    - sh permisos.sh

2.- Ejecutar el script Install.sh para poder comenzar a instalar la aplicación.

3.- En caso de dudas ver manual en www.github.com/nsm95
```


Empecemos explicando la primera opción del menú principal, “*Configuración de red*”:

```
Pasos a seguir para la correcta instalación y configuración de su servidor P2P:

1) Configuración de red.
2) Instalación Necesaria.
3) Opciones de los servicios.
4) Status de los servicios.
5) Salir.

Indique una opcion: [ ]
```

Como bien dice la opción, nos permitirá configurar la red de nuestra Rapsberry Pi:

```
Configuración de red:

1) Configurar la red de su PI
2) Restaurar fichero Interfaces
3) Volver atras.

Indique una opcion: _
```

La primera opción “*Configurar la red de su PI*” cuando la ejecutemos lo primero que hará será una copia del fichero *interfaces* y *resolv.conf* por si ocurre cualquier error a la hora de configurarlo, nos enseñará las interfaces de red utilizadas e introduciremos los datos necesario para su configuración:

```
Configuración de red:

1) Configurar la red de su PI
2) Restaurar fichero Interfaces
3) Volver atras.

Indique una opcion: 1
-----
- Configuración de red -
-----

Realizando copias de seguridad...

El archivo interfaces ha sido salvaguardado con éxito.

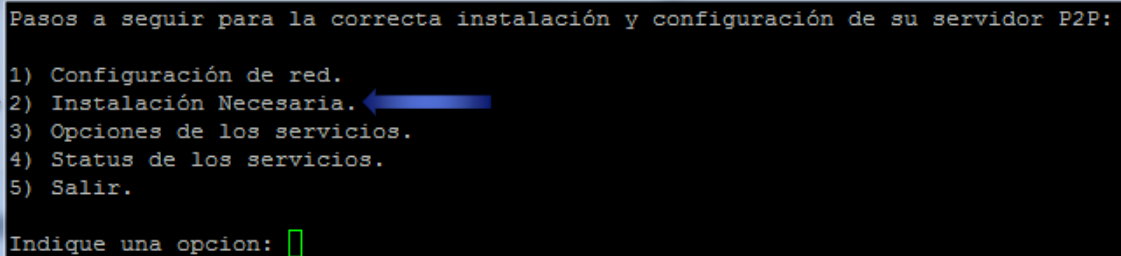
El archivo resolv.conf ha sido salvaguardado con éxito.

Mostrando interfaces de red actuales...
* Interfaz: eth0
-----
Elige la interfaz que deseas configurar: eth0
La interfaz elegida es: eth0
Introduce la nueva dirección ip estática: 192.168.1.22
Introduce la nueva máscara de red: 255.255.255.0
Introduce la puerta de enlace: 192.168.1.1
Introduce el DNS 1º: 8.8.8.8
Introduce el DNS 2º: 8.8.4.4

Pulse una tecla para continuar.
Desea grabar estos datos de configuración (S/N):
```

En caso de que no tengamos red observaremos si los datos insertados son los correctos, si no es así restauraremos el fichero interfaces con la segunda opción de este menú.

Ahora pasaremos a ver la segunda opción del menú principal, “*Instalación Necesaria*”:

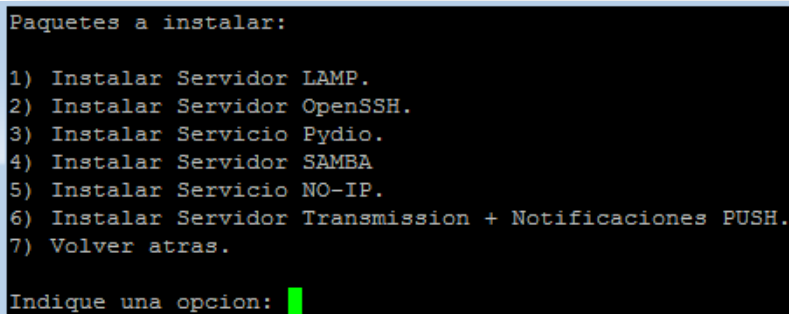


```
Pasos a seguir para la correcta instalación y configuración de su servidor P2P:

1) Configuración de red.
2) Instalación Necesaria.
3) Opciones de los servicios.
4) Status de los servicios.
5) Salir.

Indique una opcion: [ ]
```

En esta opción explicaremos la instalación de cada uno de los servicios, es obligatorio seguir el orden marcado e instalar todos los servicios para que la aplicación este completamente operativa:



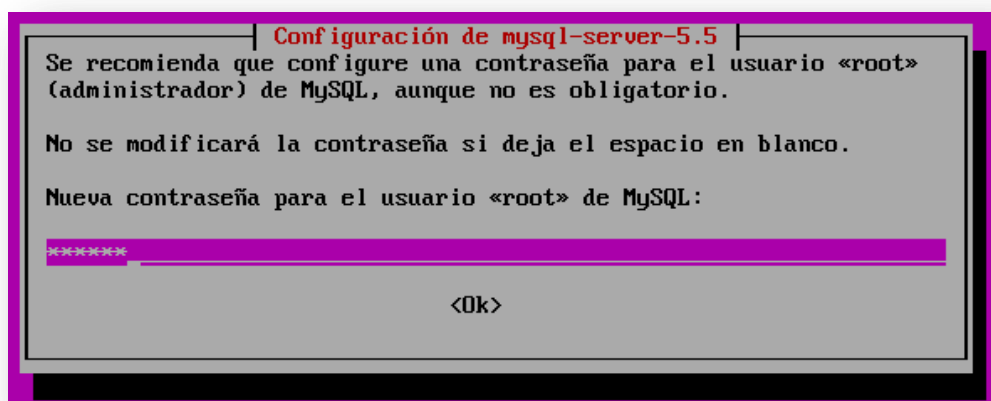
```
Paquetes a instalar:

1) Instalar Servidor LAMP.
2) Instalar Servidor OpenSSH.
3) Instalar Servicio Pydio.
4) Instalar Servidor SAMBA
5) Instalar Servicio NO-IP.
6) Instalar Servidor Transmission + Notificaciones PUSH.
7) Volver atras.

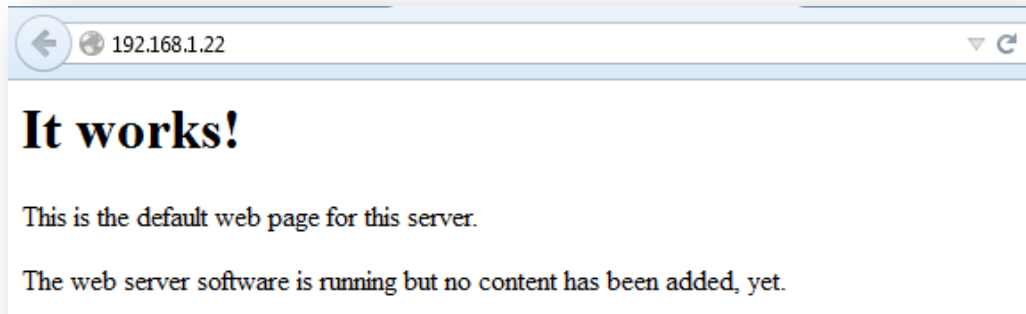
Indique una opcion: [ ]
```

LAMP:

Lamp como he explicado anteriormente es un paquete que instala, apache, mysql, php5 y algunos más servicios. En esta instalación lo único que tendremos que configurar será la contraseña de root de mysql, en la instalación aparecerá una ventana como la siguiente e introduciremos la contraseña:

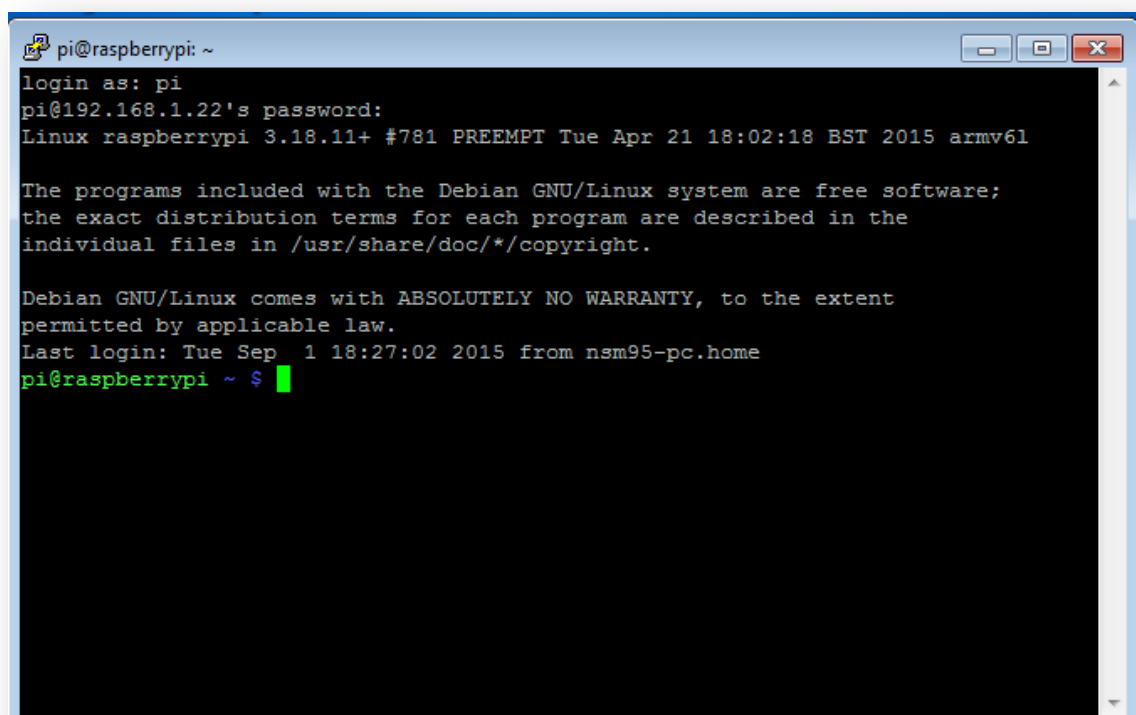


Para comprobar si el paquete instalado está funcionando una de las maneras para comprobarlos es dirigiendonos al navegador y escribiendo la dirección ip de nuestra pi y debería de aparecer esta pantalla:



Openssh:

Simplemente instala el paquete, para poder usar el servicio en mi caso instalo putty para poder acceder a la pi y así poder manejarla desde mi equipo, sin tener que conectarla a ninguna pantalla:



NO-IP:

En esta instalación se descargará no ip, una vez descargado nos pedirá que indiquemos nuestro correo que será la cuenta que tengamos creada en no-ip además de la contraseña, de esta manera asociará la pi con el host de no-ip:

```

pi@raspberrypi: ~/AplicacionP2P

make: 'noip2' is up to date.
if [ ! -d /usr/local/bin ]; then mkdir -p /usr/local/bin;fi
if [ ! -d /usr/local/etc ]; then mkdir -p /usr/local/etc;fi
cp noip2 /usr/local/bin/noip2
/usr/local/bin/noip2 -C -c /tmp/no-ip2.conf

Auto configuration for Linux client of no-ip.com.

Multiple network devices have been detected.

Please select the Internet interface from this list.

By typing the number associated with it.
0      eth0
1      eth0
0
Please enter the login/email string for no-ip.com nsm9595@gmail.com
Please enter the password for user 'nsm9595@gmail.com' *****

Only one host [nsm9595.no-ip.info] is registered to this account.
It will be used.
Please enter an update interval:[30]
Do you wish to run something at successful update?[N] (Y/N) █

```

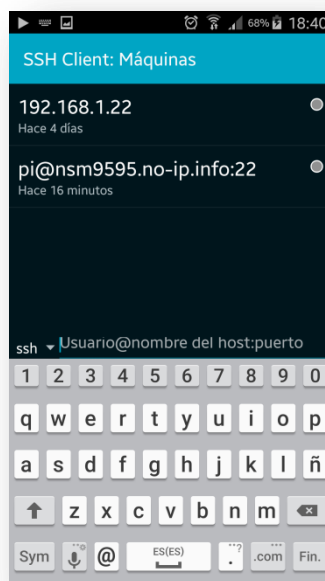
Una cosa a tener muy en cuenta es que si lo instalamos más de una vez es posible que en la página web de no-ip no se actualice la ip externa con lo que no te conectará, para ello tendrás que iniciar sesión en no-ip, dirigirte a hosts y en el hosts creado, modificarlo y cambiarle la ip por la nueva ip externa, ¿Cómo sabemos nuestra nuestra ip externa? Muy fácil, a través de esta página la sabremos <http://www.cualesmiip.com/> y la copiaremos en la configuración el host.

Hostname Information	
Hostname:	nsm9595.no-ip.info
Host Type:	<input checked="" type="radio"/> DNS Host (A) <input type="radio"/> DNS Host (Round Robin) <input type="radio"/> DNS Alias (CNAME) <input type="radio"/> Port 80 Redirect <input type="radio"/> Web Redirect
IP Address:	<input type="text" value="IP Externa"/> Last Update: 2015-09-02 09:21:56 PDT
Assign to Group:	<input type="text" value="- No Group -"/> Configure Groups
Enable Wildcard:	Wildcards are a Plus / Enhanced feature. Upgrade Now!
Advanced Records:	TXT, SPF, and SRV records and the use of some special clients are Plus / Enhanced features. Upgrade now to use them.

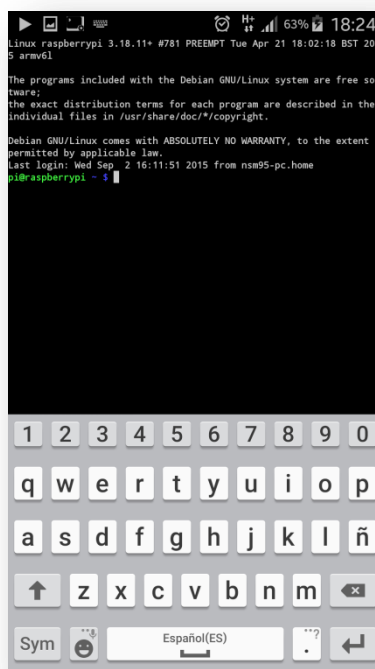
Esperaremos unos minutos y para comprobar que funciona hay dos maneras, una accediendo al navegador y escribiendo el nombre del host o la que yo utilizo que es descargándose una aplicación para tu Smartphone llamada SSH cliente, que es la aplicación que utilizaremos para poder manejar la pi desde la calle, para esto instalamos este servicio.

La aplicación es muy sencilla de utilizar:

Para poder iniciar la sesión con nuestra pi desde la calle con esta aplicación lo que tenemos que hacer es escribir el nombre de usuario + el nombre de host de no-ip + el puerto que es el 22:

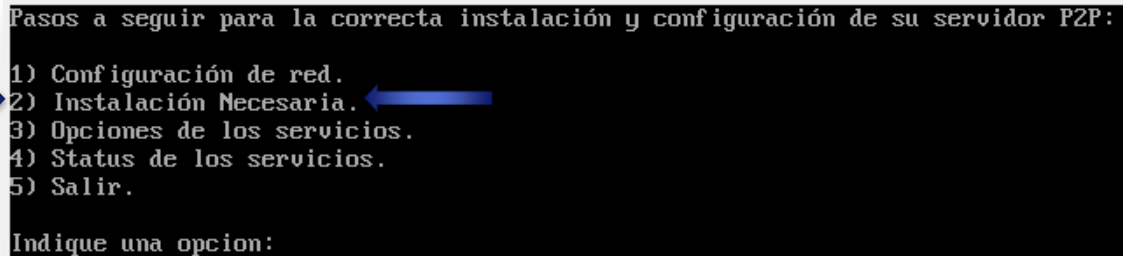


Accederemos y nos aparecerá una línea de comandos preguntándonos si deseamos conectarnos, le decimos que sí y ya estaremos dentro de nuestra pi:



Los demás servicios que quedan, samba y transmission + mensaje push, en este apartado solo se instalaran, en la siguiente opción del menú principal explicaremos la configuración de cada uno de ellos.

Ahora pasaremos con la tercera opción del menú principal, *opciones de los servicios*:

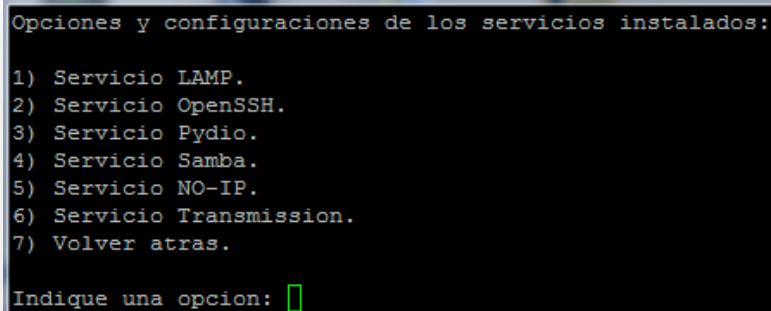


```
Pasos a seguir para la correcta instalación y configuración de su servidor P2P:
```

- 1) Configuración de red.
- 2) Instalación Necesaria.
- 3) Opciones de los servicios.
- 4) Status de los servicios.
- 5) Salir.

Indique una opcion:

En esta parte todos los servicios tienen las mismas opciones, Iniciar, parar, reiniciar y desinstalar el servicio, excepto pydio, samba y transmission que a continuación explicaré sus otras opciones de configuración:



```
Opciones y configuraciones de los servicios instalados:
```

- 1) Servicio LAMP.
- 2) Servicio OpenSSH.
- 3) Servicio Pydio.
- 4) Servicio Samba.
- 5) Servicio NO-IP.
- 6) Servicio Transmission.
- 7) Volver atras.

Indique una opcion: █

Pydio:

Una vez instalado pydio, nos dirigimos a esta opción y accederemos a ella:

```
Opciones del servicio Pydio:

1) Crear base de datos para Pydio.
2) Solucionar error de instalacion Pydio.
3) Información.
4) Volver atras.

Indica una opcion: █
```

Como vemos hay 3 opciones:

Primera opción *Crear base de datos para pydio*:

Como bien pone creamos una base de datos para que pydio esté completamente operativa:

```
Opciones del servicio Pydio:

1) Crear base de datos para Pydio.
2) Solucionar error de instalacion Pydio.
3) Información.
4) Volver atras.

Indica una opcion: 1
Indique el nombre de la base de datos: pydiobd
Indique el nombre de usuario para la base de datos: pi
Indique una contraseña para la dicho usuario: alumno
Enter password:
```

La segunda opción se tendrá que ejecutar siempre que se instale pydio por si tenemos problemas en la instalación.

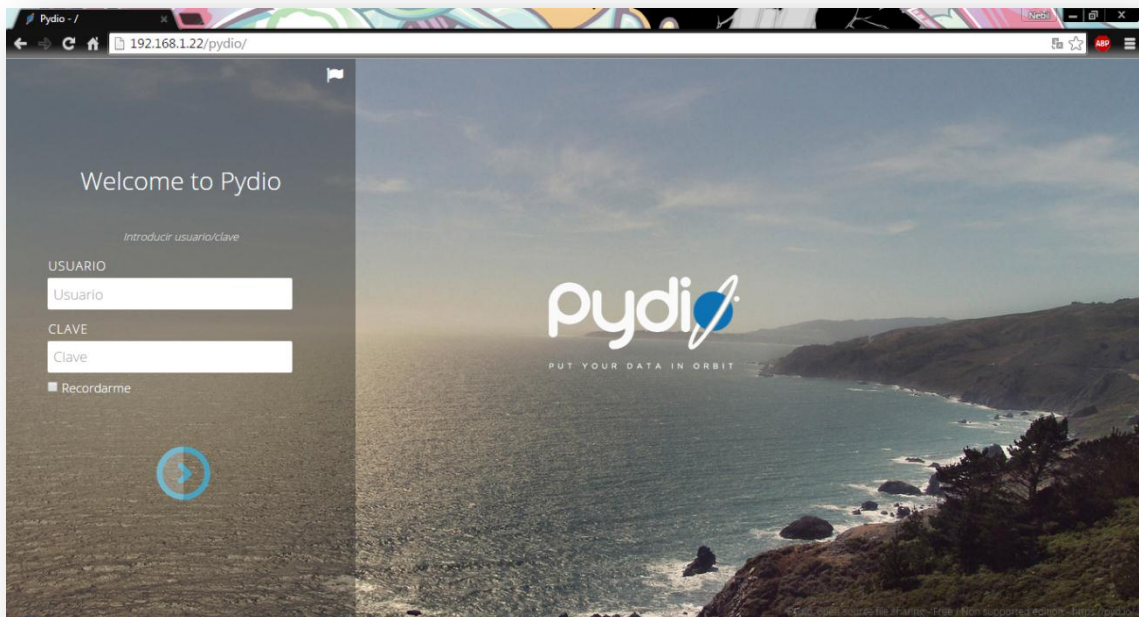
Y la tercera opción mostrara el siguiente paso para acceder a pydio:

```
Opciones del servicio Pydio:

1) Crear base de datos para Pydio.
2) Solucionar error de instalacion Pydio.
3) Información.
4) Volver atras.

Indica una opcion: 3

PARA PODER ACCEDER A PYDIO, TENDRÁ QUE DIRIGIRSE A UN NAVEGADOR Y ESCRIBIR LA DIRECCIÓN IP DE SU RASPBERRY MÁS PYDIO, ejemplo 192.168.1.x/pydio
```



Samba:

Accedemos a la configuración de Samba y nos aparecerá lo siguiente:

Únicamente tenemos que realizar la opción 3 ya que tenemos el usuario pi creado:

```
Opciones del servicio Samba:

1) Añadir Usuario a SAMBA.
2) Eliminar Usuario de SAMBA.
3) Añadir directorio compartido.
4) Iniciar Servicio Samba.
5) Parar Servicio Samba.
6) Reiniciar Servicio Samba.
7) Desinstalar Servidor SAMBA.
8) Volver atras.

Indica una opcion: 3
Indique el nombre de la carpeta: SAMBA
Ubicación en el servidor: /home/pi/AplicacionP2P/
mkdir: cannot create directory `/home/pi/AplicacionP2P/': File exists
Explorable? YES/NO: yes
Solo lectura? YES/NO: no
Usuarios validos (Separados por comas): pi
Permisos para crear archivos en codigo numerico: 777
Permisos para crear directorios en codigo numerico: 77

Carpeta Compartida añadida al fichero smb.conf. Reinicie el servicio para hacer e
fectivo el cambio
```



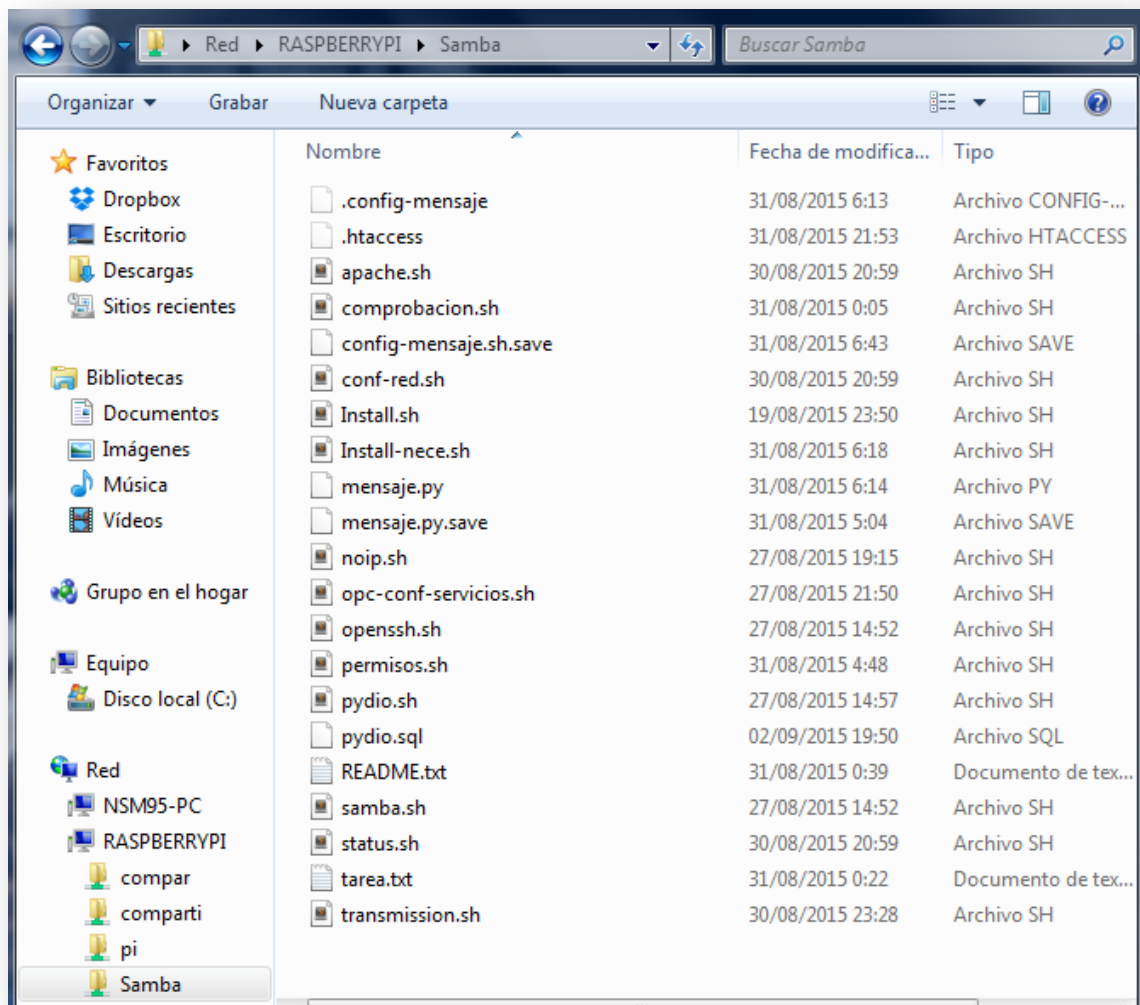
```

Opciones del servicio Samba:

1) Añadir Usuario a SAMBA.
2) Eliminar Usuario de SAMBA.
3) Añadir directorio compartido.
4) Iniciar Servicio Samba.
5) Parar Servicio Samba.
6) Reiniciar Servicio Samba.
7) Desinstalar Servidor SAMBA.
8) Volver atras.

Indica una opcion: 6
[ ok ] Stopping Samba daemons: nmbd smbd.
[ ok ] Starting Samba daemons: nmbd smbd.
Servicio reiniciado correctamente
    
```

Una vez realizado los pasos desde un equipo en mi caso windows 7 vamos a red y accederemos a RASPBERRYPI y ahí encontraremos todas las carpeta compartidas:



Por último la configuración de transmission, aquí simplemente añadimos dos opciones para ver el contenido de las carpetas donde se descargan los torrents:

```
Opciones y configuración del servicio Transmission:
```

- 1) Contenido de la carpeta de Torrents COMPLETOS.
- 2) Contenido de la carpeta de Torrents INCOMPLETOS.
- 3) Iniciar Servicio Transmission.
- 4) Parar Servicio Transmission.
- 5) Reiniciar Servicio Transmission.
- 6) Desinstalar Servicio Transmission.
- 7) Información.
- 8) Volver atras.

```
Indique una opcion: █
```

Por último queda explicar la cuarta y última opción del menú principal “Status de los servicios” donde nos mostrará si el servicio está en funcionamiento o está parado:

```
Estatus de los servicios instalados:
```

- 1) Estatus del Servicio LAMP.
- 2) Estatus del Servicio Samba.
- 3) Estatus del Servicio NO-IP.
- 4) Estatus del Servicio Transmission.
- 5) Volver atras.

```
Indique una opcion:
```

Haremos un ejemplo con el servicio LAMP:

```
Estatus de los servicios instalados:
```

- 1) Estatus del Servicio LAMP.
- 2) Estatus del Servicio Samba.
- 3) Estatus del Servicio NO-IP.
- 4) Estatus del Servicio Transmission.
- 5) Volver atras.

```
Indique una opcion: 4
```

```
-----  
-      Estatus Servicio Transmission      -  
-----
```

```
* transmission-daemon is running
```

En caso de que nosotros lo paremos utilizando la opción del menú anterior esta opción nos tendrá que mostrar que el servicio está parado:

```
Opciones del servicio Apache:
```

- 1) Iniciar Servicio Apache.
- 2) Parar Servicio Apache.
- 3) Reiniciar Servicio Apache.
- 4) Iniciar Servicio Mysql.
- 5) Parar Servicio Mysql.
- 6) Reiniciar Servicio Mysql.
- 7) Desinstalar Servicio LAMP.
- 8) Volver atras.

```
Indique una opcion: 2
```

```
* Stopping web server apache2  
*
```

```
Estatus de los servicios instalados:
```

- 1) Estatus del Servicio LAMP.
- 2) Estatus del Servicio Samba.
- 3) Estatus del Servicio NO-IP.
- 4) Estatus del Servicio Transmission.
- 5) Volver atras.

```
Indique una opcion: 1
```

```
-----  
-          Estatus Servicio LAMP          -  
-----
```

```
* apache2 is not running
```

Mejoras

Decir que este proyecto está basado en el de un compañero de clase.

Comencemos por el **menú**:

Nuevo menú es más intuitivo, ordenado y más fácil de moverte por él:

```
Pasos a seguir para la correcta instalación y configuración de su servidor P2P:

1) Configuración de red.
2) Instalación Necesaria.
3) Opciones de los servicios.
4) Status de los servicios.
5) Salir.

Indique una opción:
```

La siguiente mejora consiste en la **asignación de una IP estática**.

Otra de las mejoras realizadas ha sido el **correcto funcionamiento de los servicios**

El **servicio de mensajería**, el servicio que más problemas ha dado, buscando y buscando manuales siempre aparece la misma configuración, la configuración consiste en dirigirnos al fichero de configuración *settings.json* que se encuentra en */var/lib/transmission-daemon/info* y editar una línea del fichero para escribir la ruta de nuestro script con el mensaje, si realizamos ese cambio, al descargarse cualquier archivo esa línea lo que hace es ejecutar el script del mensaje y así poder mandarlo, pero no lo hace de ninguna manera, he probado con permisos, haciendo un puente al script, utilizando transmission en entorno gráfico y nada, sigue sin funcionar:

```
script-torrent-done-enabled: true
"script-torrent-done-enabled": false,
"script-torrent-done-filename": "",
```

La línea donde tenemos que añadir la ruta de nuestro script de mensajería es *"script-torrent-done-filename": ""*, y cuando finalice la descarga nos mandara un mensaje, a parte tener también la línea anterior en *true*, pero no es así.

Al no funcionar esta forma pensé en una forma un poco más complicada y consiste en crear un script que cada cierto tiempo compruebe que en la carpeta donde descarga se haya añadido un archivo, si es así que mande el mensaje y si no es así que siga comprobando hasta que encuentre un nuevo archivo en la carpeta, de esta manera funciona, pero hay un inconveniente que transmission añade el archivo mientras se está descargando, con lo que el mensaje llega antes de que termine la descarga.

Este es el script que realiza la comprobación:

```
#!/bin/bash

var1=`ls /var/lib/transmission-daemon/downloads | wc -l`

#echo "copiada var 1"

sleep 5

var2=`ls /var/lib/transmission-daemon/downloads | wc -l`

#echo "Variables copiadas"

if [ $var2 -gt $var1 ];
then
    #echo "mensaje enviado 2"
    sudo python /home/pi/AplicacionP2P/mensaje.py
    sleep 5
    cd /home/pi/AplicacionP2P
    ./comprobar.sh
else
    #echo "No hay nuevos ficheros"
    cd /home/pi/AplicacionP2P
    ./comprobar.sh
fi
```

Y este es el script que manda el mensaje, esta creado con python y el mensaje llegará a la aplicación que tendremos instalada en el móvil llamada pushetta:

```
#Importamos las librerias
import urllib2
import json
#Esta es la funcion que usara el script para enviar las notificaciones
def sendNotification(token, channel, message):
    data = {
        "body" : message,
        "message_type" : "text/plain"
    }

    req = urllib2.Request('http://api.pushetta.com/api/pushes/{0}/'.format(channel))
    req.add_header('Content-Type', 'application/json')
    req.add_header('Authorization', 'Token {0}'.format(token))

    response = urllib2.urlopen(req, json.dumps(data))

#Enviamos la notificacion en formato '<strong>Apikey+Canal+Mensaje</strong>'
sendNotification("fb232cf502fda3c39c9107cb32f0b6573877798a", "mensajes", "La descarga ha finalizado con exito!")

#Avisamos al usuario de que ya se ha enviado el mensaje
print "Mensaje enviado!"
```

La penúltima mejora es añadirle a la aplicación un apartado para comprobar en qué estado están los servicios, explicado en la hoja número 10.

En la última mejora explico cómo debemos configurar la aplicación Pushetta en el móvil:

Para ello deberemos ir a la página principal de pushetta <http://www.pushetta.com/>, crearte una cuenta, crear un canal de difusión.

Podemos monitorizar la actividad de los mensajes desde la web de pushetta logeándonos en la web de esta, o bajándonos la aplicación en el móvil y utilizando el buscador, localizamos nuestro canal creado previamente, y nos suscribimos a el, de esta manera llegará una notificación a nuestro Smartphone cuando se realice una descarga.

La instalación completa de la mensajería, consta de varias partes:

En primer lugar instalar las librerías de python para interpretar el script e insertar en crontab la tarea para que se ejecute el script de comprobación.sh al inicio del sistema.

En segundo lugar colocar los script en las carpetas correctas.

Y por último configurar a partir de la cuenta creada en pushetta el script, indicando el APIKEY cuando nos lo pida y el nombre del canal.

Conclusión y valoración personal

Para mí, este proyecto ha conseguido que le dé una oportunidad más a la programación ya que antes no la podía ni ver, era tal el odio que le tenía a los scripts que ahora hasta les he cogido el gusto hacerlos.

Ha sido un proyecto complicado ya que no parto desde cero, parto con una idea realizada por otra persona, quieras o no, es mucho más fácil empezar desde cero que ir buscando cada uno de los fallos, pero aun así estoy bastante satisfecho que el trabajo realizado, creo que cumple con las expectativas marcadas.

Biografía y fuentes de consultas

Script configuración de la red:

<http://profesorjlgarcia.blogspot.com.es/2014/01/script-para-configuracion-de-ip-estatica.html>

Configuración Transmission:

<http://raspberryparatorpes.net/proyectos/instalar-y-configurar-un-descargador-de-torrents/>

Servicio de mensajería (Pushetta + Script):

<http://www.pushetta.com/pushetta-docs/>

Configuración de Pydio:

<https://geekytheory.com/tutorial-raspberry-pi-crea-una-nube-privada-con-pydio/>

Configuración de mensajería en transmission que no funciona:

https://code.google.com/p/torrentexpander/wiki/How_to_setup_this_script_with_Transmission

Fecha de terminación de proyecto

31/08/2015

Anexo

Readme:

1.- Ejecutar el script permisos de esta manera:

- sh permisos.sh

2.- Ejecutar el script Install.sh para poder comenzar a instalar la aplicación.

3.- En caso de dudas ver manual en www.github.com/nsm95

Permisos:

```
#!/bin/bash
```

```
sudo chmod 777 /home/pi/AplicacionP2P/*
```

```
echo "Completado con exito"
```

Install.sh:

```

#!/bin/bash
_MENU()
{
    echo "Pasos a seguir para la correcta instalación y configuración de su servidor P2P: "
    echo
    echo "1) Configuración de red."
    echo "2) Instalación Necesaria."
    echo "3) Opciones de los servicios."
    echo "4) Status de los servicios."
    echo "5) Salir."
    echo
    echo -n "Indique una opcion: "
}
until [ "$opc" = "5" ];
do
    case $opc in
        1) sudo ./conf-red.sh
            clear
            _MENU ;;

        2) sudo ./Install-nece.sh
            clear
            _MENU ;;

        3) sudo ./opc-conf-servicios.sh
            clear
            _MENU ;;

        4) sudo ./status.sh
            clear
            _MENU ;;

        5) exit ;;

        *)
            clear
            _MENU
            ;;
    esac
    read opc
done

```

Conf-red.sh:

```
#!/bin/bash

_MENU()
{
    echo "Configuración de red: "
    echo
    echo "1) Configurar la red de su PI"
    echo "2) Restaurar fichero Interfaces"
    echo "3) Volver atras."
    echo
    echo -n "Indique una opcion: "
}
until [ "$opc" = "3" ];
do

case $opc in

    1)

echo "-----"
echo "-      Configuración de red      -"
echo "-----"
echo

if [ $LOGNAME != "root" ]
then
    read -p "Lo siento, este script debe ser ejecutado con privilegios de root."
    exit 1
fi

echo "Realizando copias de seguridad..."
echo
if [ -f /etc/network/interfaces ]
then
    cp /etc/network/interfaces /etc/network/interfaces.original
if [ $? -eq 0 ]
then
    echo "El archivo interfaces ha sido salvaguardado con éxito."
else
    echo "Error al salvaguardar el archivo interfaces."
fi
fi
echo
if [ -f /etc/resolv.conf ]
then
    cp /etc/resolv.conf /etc/resolv_original.conf
```

```

if [ $? -eq 0 ]
then
    echo "El archivo resolv.conf ha sido salvaguardado con éxito."
else
    echo "Error al salvaguardar el archivo resolv.conf."
fi
fi
echo
echo "Mostrando interfaces de red actuales..."
for nic in `lshw -short -class network | grep eth | tr -s " " | cut -f2 -d" "`
do
    echo "* Interfaz: $nic"
done
echo "-----"
interfaz=""
while [ "$interfaz" == "" ]
do
    read -p "Elige la interfaz que deseas configurar: " interfaz
done

interfaz=1
for nic in `lshw -short -class network | grep eth | tr -s " " | cut -f2 -d" "`
do
    if [ $nic == $interfaz ]
    then
        interfaz=0
    fi
done

echo "La interfaz elegida es: $nic"
read -p "Introduce la nueva dirección ip estática: " ip
read -p "Introduce la nueva máscara de red: " mascara
read -p "Introduce la puerta de enlace: " puerta
read -p "Introduce el DNS 1º: " dns1
read -p "Introduce el DNS 2º: " dns2
echo
read -p "Pulse una tecla para continuar."

respuesta=""
while [ "$respuesta" == "" ]
do
    read -p "Desea grabar estos datos de configuración (S/N): " respuesta
    case $respuesta in
        s/S)

            #Modificación de /etc/network/interfaces
            interfaces="/etc/network/interfaces"
            echo "auto lo" > $interfaces
    esac
done

```

```

echo "iface lo inet loopback" >> $interfaces
echo
echo "auto $nic" >> $interfaces
echo "allow-hotplug $nic" >> $interfaces
echo "iface $nic inet static" >> $interfaces
echo "address $ip" >> $interfaces
echo "netmask $mascara" >> $interfaces
echo "gateway $puerta" >> $interfaces

#Modificación de /etc/resolv.conf
resolv="/etc/resolv.conf"
echo "nameserver $dns1" >> $resolv
echo "nameserver $dns2" >> $resolv

#Reinicio de la red
/etc/init.d/networking restart

break;;
n|N)echo "Ok, no se modificarán los archivos de configuración"; read; break;;
*) echo "No ha introducido una respuesta válida, vuelva a intentarlo.";;
esac
done
_MENU
;;
2) echo "Restaurando copias de seguridad del fichero Interfaces..."
echo
cp /etc/network/interfaces.original /etc/network/interfaces

3) sudo ./Install.sh ;;
*)
clear
_MENU
;;
esac
read opc
done

```

Install-nece.sh:

```
#!/bin/bash

_MENU()
{
    echo "Paquetes a instalar: "
    echo
    echo "1) Instalar Servidor LAMP."
    echo "2) Instalar Servidor OpenSSH."
    echo "3) Instalar Servicio Pydio."
    echo "4) Instalar Servidor SAMBA"
    echo "5) Instalar Servicio NO-IP."
    echo "6) Instalar Servidor Transmission + Notificaciones PUSH."
    echo "7) Volver atras."
    echo
    echo -n "Indique una opcion: "
}
until [ "$opc" = "7" ];
do
    case $opc in
        1) sudo apt-get -y install apache2 mysql-server libapache2-mod-php5 php5 php5-
mcrypt
        echo
        echo "Servicio instalado correctamente"
        sleep 4

        _MENU
        ;;

        2) sudo apt-get install -y openssh-client openssh-server ssh
        echo
        echo "Servicio instalado correctamente"
        sleep 4

        _MENU
        ;;

        3) cd /home/pi
        wget http://kent.dl.sourceforge.net/project/ajaxplorer/pydio/stable-
channel/6.0.7/pydio_6.0.7_all.deb
        sudo dpkg -i pydio_6.0.7_all.deb
        sudo apt-get -f -y install
        sudo cp /usr/share/doc/pydio/apache2.sample.conf/etc/apache2/sites-
enabled/pydio.conf
        sudo service apache2 restart
        echo
        echo "Servicio instalado correctamente"
```

sleep 4

_MENU

::

```
4) sudo apt-get -y install samba samba-common samba-common-bin samba-doc
sudo sed -i "s/$(head -n 102 /etc/samba/smb.conf | tail -1)/ \security\ = user\ /g"
/etc/samba/smb.conf
echo
echo "Servicio instalado correctamente"
sleep 4
```

_MENU

::

```
5) cd /home/pi
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
tar -zxf noip-duc-linux.tar.gz
cd noip-2.1.9-1/
make
sudo make install
cd ..
rm -r noip-duc-linux.tar.gz
echo
echo
sleep 4
_MENU
::
```

```
6) sudo apt-get -y install transmission transmission-daemon
cd /home/pi
sudo apt-get install python-dev
echo
```

```
sudo chmod -R 777 /etc/transmission-daemon
sudo /etc/init.d/transmission-daemon stop
```

```
read -p "Introduzca nombre de usuario: " var1
```

```
sudo sed -i "s%$(head -n 51 /var/lib/transmission-daemon/info/settings.json | tail -
1)% \\"rpc-username\\": \\"$var1\\", %g" /var/lib/transmission-daemon/info/settings.json
echo
read -p "Intruzca la contraseña: " var2
```

```
sudo sed -i "s%$(head -n 48 /var/lib/transmission-daemon/info/settings.json | tail -
1)% \\"rpc-password\\": \\"$var2\\", %g" /var/lib/transmission-daemon/info/settings.json
echo
```



```
sudo sed -i "s%$(head -n 53 /var/lib/transmission-daemon/info/settings.json | tail -  
1)% \\"rpc-whitelist-enabled\\": \"\\false\"\\, %g\" /var/lib/transmission-daemon/info/settings.json
```

```
crontab tarea.txt
```

```
sudo /etc/init.d/transmission-daemon start
```

```
echo "Servicio instalado correctamente"
```

```
sleep 4
```

```
_MENU
```

```
;;
```

```
7) sudo ./Install.sh ;;
```

```
*)
```

```
clear
```

```
_MENU
```

```
;;
```

```
esac
```

```
read opc
```

```
done
```

Opc-conf-servicios.sh:

```

#!/bin/bash

_MENU()
{
    echo "Opciones y configuraciones de los servicios instalados: "
    echo
    echo "1) Servicio LAMP."
    echo "2) Servicio OpenSSH."
    echo "3) Servicio Pydio."
    echo "4) Servicio Samba."
    echo "5) Servicio NO-IP."
    echo "6) Servicio Transmission."
    echo "7) Volver atras."
    echo
    echo -n "Indique una opcion: "
}

until [ "$opc" = "7" ];
do
    case $opc in
        1) sudo ./apache.sh
            clear
            _MENU ;;
        2) sudo ./openssh.sh
            clear
            _MENU ;;
        3) sudo ./pydio.sh
            clear
            _MENU ;;
        4) sudo ./samba.sh
            clear
            _MENU ;;
        5) sudo ./noip.sh
            clear
            _MENU ;;
        6) sudo ./transmission.sh
            clear
            _MENU ;;
        7) sudo ./Install.sh ;;
        *)
            clear
            _MENU
            ;;
    esac
    read opc
done

```

Apache.sh (LAMP):

```
#!/bin/bash
_MENU()
{
    echo "Opciones del servicio Apache: "
    echo
    echo "1) Iniciar Servicio Apache."
    echo "2) Parar Servicio Apache."
    echo "3) Reiniciar Servicio Apache."
    echo "4) Iniciar Servicio Mysql."
    echo "5) Parar Servicio Mysql."
    echo "6) Reiniciar Servicio Mysql."
    echo "7) Desinstalar Servicio LAMP."
    echo "8) Volver atras."
    echo
    echo -n "Indique una opcion: "
}
until [ "$opc" = "8" ];
do
    case $opc in

        1) sudo /etc/init.d/apache2 start
            echo
            echo
            echo "Servicio iniciado correctamente"
            sleep 2
            _MENU
            ;;

        2) sudo /etc/init.d/apache2 stop
            echo
            echo
            echo "Servicio parado correctamente"
            sleep 2
            _MENU
            ;;

        3) sudo /etc/init.d/apache2 restart
            echo
            echo
            echo "Servicio reiniciado correctamente"
            sleep 2
            _MENU
            ;;

        4) sudo /etc/init.d/mysql start
```

```
echo
echo
echo "Servicio iniciado correctamente"
sleep 2
_MENU
;;

5) sudo /etc/init.d/mysql stop
echo
echo
echo "Servicio parado correctamente"
sleep 2
_MENU
;;

6) sudo /etc/init.d/mysql restart
echo
echo
echo "Servicio reiniciado correctamente"
sleep 2
_MENU
;;

7) sudo apt-get -y purge apache2 mysql-server libapache2-mod-php5 php5 php5-mcrypt
echo
echo "Servicio eliminado correctamente"
sleep 4
_MENU
;;

8) sudo ./opc-conf-servicios.sh ;;

*)
clear
_MENU
;;
esac
read opc
done
```

Openssh.sh:

```
#!/bin/bash
_MENU()
{
    echo "Opciones del servicio SSH: "
    echo
    echo "1) Iniciar Servicio OpenSSH."
    echo "2) Parar Servicio OpenSSH."
    echo "3) Reiniciar Servicio OpenSSH."
    echo "4) Desinstalar Servicio OpenSSH."
    echo "5) Volver atras."
    echo
    echo -n "Indica una opcion: "
}
until [ "$opc" = "5" ];
do
    case $opc in
        1) sudo /etc/init.d/ssh start
            echo
            echo "Servicio instalado correctamente"
            sleep 2
            _MENU
            ;;
        2) sudo /etc/init.d/ssh stop
            echo
            echo "Servicio instalado correctamente"
            sleep 2
            _MENU
            ;;
        3) sudo /etc/init.d/ssh restart
            echo
            echo "Servicio instalado correctamente"
            _MENU
            ;;
        4) sudo apt-get purge -y openssh-client openssh-server ssh
            echo
            echo "Servicio eliminado correctamente"
            _MENU
            ;;
        5) sudo ./opc-conf-servicios.sh ;;
        *)
            clear
            _MENU
            ;;
    esac
done
read opc
done
```

Pydio.sh:

```

#!/bin/bash

_MENU()
{
    echo "Opciones del servicio Pydio: "
    echo
    echo "1) Crear base de datos para Pydio."
    echo "2) Solucionar error de instalacion Pydio."
    echo "3) Información. "
    echo "4) Volver atras."
    echo
    echo -n "Indica una opcion: "
}

until [ "$opc" = "4" ];
do
    case $opc in
        1) read -p "Indique el nombre de la base de datos: " var1
            sudo echo "DROP DATABASE IF EXISTS $var1 ;" > pydio.sql
            sudo echo "CREATE DATABASE $var1;" >> pydio.sql
            read -p "Indique el nombre de usuario para la base de datos: " var2
            read -p "Indique una contraseña para la dicho usuario: " var3
            sudo echo "CREATE USER $var2@localhost IDENTIFIED BY '$var3';" >>
pydio.sql
            sudo echo "GRANT ALL ON $var1.* TO $var2@localhost;" >> pydio.sql
            sudo echo "FLUSH PRIVILEGES;" >> pydio.sql
            mysql -u root -p --default-character-set=utf8 < pydio.sql
            echo
            echo "Base de datos creada correctamente"
            sleep 4
            _MENU
            ;;
        2) sudo echo "<IfModule mod_rewrite.c>
# You must set the correct values here if you want
# to enable webDAV sharing. The values assume that your
# Pydio installation is at http://yourdomain/
# and that you want the webDAV shares to be accessible via
# http://yourdomain/shares/repository_id/
RewriteEngine on
RewriteBase /pydio
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^shares ./dav.php [L]
RewriteRule ^api ./rest.php [L]
RewriteRule ^user ./index.php?get_action=user_access_point [L]
RewriteCond %{REQUEST_URI} !^/pydio/index
RewriteCond %{REQUEST_URI} !^/pydio/plugins

```

```

RewriteCond %{REQUEST_URI}
^/pydio/dashboard|^/pydio/welcome|^/pydio/settings|^/pydio/ws-
RewriteRule (.*) index.php [L]

#Following lines seem to be necessary if PHP is working
#with apache as CGI or FCGI. Just remove the #
#See http://doc.tiki.org/WebDAV#Note\_about\_Apache\_with\_PHP\_as\_fcgi\_or\_cgi

#RewriteCond %{HTTP:Authorization} ^(.*)
#RewriteRule ^(.*) - [E=HTTP_AUTHORIZATION:%1]

#Following lines may be necessary for a PHP-FPM setup
# to make sure that authorization is transmitted.
# Just remove the # at the beginning of the line
</IfModule> > " > /usr/share/pydio/.htaccess
    sudo a2enmod ssl
    sudo a2ensite default-ssl
    sudo /etc/init.d/apache2 restart
    sudo echo "<?php
/*
 * Copyright 2007-2013 Charles du Jeu <contact (at) cdujeu.me>
 * This file is part of Pydio.
 *
 * Pydio is free software: you can redistribute it and/or modify
 * it under the terms of the GNU Affero General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * Pydio is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public License
 * along with Pydio. If not, see <http://www.gnu.org/licenses/>.
 *
 * The latest code can be found at <http://pyd.io/>.
 *
 * These configuration must be set at the very root loading of the framework
 */

/*
 * If you have a charset warning, or problems displaying filenames with accented characters,
 * check your system locale and set it in the form lang_country.charset
 * Example : fr_FR.UTF-8, fr_FR.ISO-8859-1, fr_FR.CP1252 (windows), en_EN.UTF-8, etc.
 *
 * Windows users may define an empty string
 * define("AJXP_LOCALE", "");

```

```

*/
define("AJXP_LOCALE", "es_ES.UTF-8");
//define("AJXP_LOCALE", "");

/*
* If you encounter problems writing to the standard php tmp directory, you can
* define your own tmp dir here. Suggested value is ajaxp_path/data/tmp/
* AJXP_DATA_PATH, AJXP_INSTALL_PATH are replaced automatically.
*
* See php.ini settings below for the session.save_path value as well.
*/
//define("AJXP_TMP_DIR", AJXP_DATA_PATH."/tmp");

/*
* Additionnal php.ini settings
* > Problems with tmp dir : set your own session tmp dir (create it and make it writeable!)
* > Concurrent versions of AjaXplorer : use session.cookie_path to differentiate them.
*/
$AJXP_INISET = array();
//$AJXP_INISET["session.save_path"] = AJXP_DATA_PATH."/tmp/sessions";
//$AJXP_INISET["session.cookie_path"] = "/ajaxplorer";

/*
* If you want to force the https, uncomment the line below. This will automatically
* redirect all calls to ajaxplorer via http to the same URL with https
*/
define("AJXP_FORCE_SSL_REDIRECT", true);" > /etc/pydio/bootstrap_conf.php
;;
3)
echo
echo "PARA PODER ACCEDER A PYDIO, TENDRÁ QUE DIRIGIRSE A UN
NAVEGADOR Y ESCRIBIR LA DIRECCIÓN IP DE SU RASPBERRY MÁS PYDIO, ejemplo
"192.168.1.x/pydio""
echo
_MENU
;;
4) sudo ./opc-conf-servicios.sh ;;

*)
clear
_MENU
;;

esac
read opc
done

```


Samba.sh:

```
#!/bin/bash
_MENU()
{
    echo "Opciones del servicio Samba: "
    echo
    echo "1) Añadir Usuario a SAMBA."
    echo "2) Eliminar Usuario de SAMBA."
    echo "3) Añadir directorio compartido."
    echo "4) Iniciar Servicio Samba."
    echo "5) Parar Servicio Samba."
    echo "6) Reiniciar Servicio Samba."
    echo "7) Desinstalar Servidor SAMBA."
    echo "8) Volver atras."
    echo
    echo -n "Indica una opcion: "
}
until [ "$opc" = "8" ];
do
    case $opc in
        1) read -p "Indique el nombre de Usuario: " var1;
            sudo adduser --no-create-home $var1;
            sudo smbpasswd -a $var1;
            echo
            echo
            echo "Usuario añadido correctamente a Samba"
            sleep 4
            _MENU
            ;;
        2) read -p "Seleccione el nombre de Usuario a ELIMINAR: " var2;
            sudo smbpasswd -x $var2
            echo
            echo
            echo "Usuario Eliminado"
            sleep 4
            _MENU
            ;;
        3) read -p "Indique el nombre de la carpeta: " var3
            sudo echo "[$var3]" >> /etc/samba/smb.conf
            read -p "Ubicación en el servidor: " var4
            sudo mkdir $var4
            sudo echo "path = $var4" >> /etc/samba/smb.conf
            read -p "Explorable? YES/NO: " var5
            sudo echo "browsable = $var5" >> /etc/samba/smb.conf
            read -p "Solo lectura? YES/NO: " var6
            sudo echo "read only = $var6" >> /etc/samba/smb.conf
            read -p "Usuarios validos (Separados por comas): " var7
            sudo echo "valid user = $var7" >> /etc/samba/smb.conf
```

```

read -p "Permisos para crear archivos en codigo numerico: " var8
sudo echo "create mask = $var8" >> /etc/samba/smb.conf
read -p "Permisos para crear directorios en codigo numerico: " var9
sudo echo "directory mask = $var9" >> /etc/samba/smb.conf
echo
echo
echo
echo "Carpeta Compartida añadida al fichero smb.conf. Reinicie el servicio para hacer
efectivo el cambio"
sleep 4
_MENU
;;
4) sudo service samba start
echo "Servicio iniciado correctamente"
sleep 4
_MENU
;;
5) sudo service samba stop
echo "Servicio parado correctamente"
sleep 4
_MENU
;;
6) sudo service samba restart
echo "Servicio reiniciado correctamente"
sleep 4
_MENU
;;
7) sudo apt-get -y purge samba samba-common samba-common-bin samba-doc
echo
echo
echo "Servicio eliminado correctamente"
sleep 4
_MENU
;;
8) sudo ./opc-conf-servicios.sh ;;

*)
clear
_MENU
;;
esac
read opc
done

```

No-ip.sh:

```

#!/bin/bash
_MENU()
{
    echo "Opciones del servicio NO-IP: "
    echo
    echo "1) Crear el script para iniciar automaticamente."
    echo "2) Información. "
    echo "3) Volver atras."
    echo
    echo -n "Indique una opcion: "
}
until [ "$opc" = "3" ];
do
    case $opc in
        1) echo "##! /bin/bash
        ### BEGIN INIT INFO
        # Provides: blabla
        # Required-Start: $syslog
        # Required-Stop: $syslog
        # Default-Start: 2 3 4 5
        # Default-Stop: 0 1 6
        # Short-Description: blabla
        # Description:
        #
        ### END INIT INFO
        sudo /usr/local/bin/noip2" > /etc/init.d/noip2
        sudo chmod +x /etc/init.d/noip2
        sudo update-rc.d noip2 defaults
        echo
        echo "Script creado correctamente"
        _MENU
        ;;
        2)
        echo
        echo "PARA PODER ACCEDER A NUESTRA PI DESDE LA CALLE, TENDRÁ QUE
        CONECTARSE A TRAVES DE PUTTY Y PONER SU HOST. "
        echo
        _MENU
        ;;
        3) sudo ./opc-conf-servicios.sh ;;
        *)
        clear
        _MENU
        ;;
    esac
done
read opc
done

```

Transmission.sh:

```
#!/bin/bash
_MENU()
{
    echo "Opciones y configuración del servicio Transmission: "
    echo
    echo "1) Contenido de la carpeta de Torrents COMPLETOS."
    echo "2) Contenido de la carpeta de Torrents INCOMPLETOS."
    echo "3) Iniciar Servicio Transmission."
    echo "4) Parar Servicio Transmission."
    echo "5) Reiniciar Servicio Transmission."
    echo "6) Desinstalar Servicio Transmission."
    echo "7) Información. "
    echo "8) Volver atras."
    echo
    echo -n "Indique una opcion: "
}
until [ "$opc" = "8" ];
do
    case $opc in
        1) ls /var/lib/transmission-daemon/downloads
            echo
            _MENU
            ;;
        2) ls /root/Downloads
            echo
            _MENU
            ;;
        3) sudo /etc/init.d/transmission-daemon start
            echo
            echo
            echo "Servicio iniciado correctamente"
            sleep 2
            _MENU
            ;;
        4) sudo /etc/init.d/transmission-daemon stop
            echo
            echo
            echo "Servicio parado correctamente"
            sleep 2
            _MENU
            ;;
        5) sudo /etc/init.d/transmission-daemon restart
            echo
            echo
            echo "Servicio reiniciado correctamente"
            sleep 2
            _MENU
    esac
done
```

```

;;
6) sudo apt-get -y purge transmission transmission-daemon
echo
echo
echo "Servicio eliminado correctamente"
sleep 4
_MENU
;;
7)
echo
echo "PARA PODER ACCEDER A TRANSMISSION, TENDRÁ QUE DIRIGIRSE A UN
NAVEGADOR Y ESCRIBIR LA DIRECCIÓN IP DE SU RASPBERRY MÁS :9091, ejemplo
"192.168.1.x:9091""
echo
_MENU
;;
8) sudo ./opc-conf-servicios.sh ;;
*)
clear
_MENU
;;
esac
read opc
done

```

Statuds.sh:

```
#!/bin/bash
```

```
_MENU()
```

```
{
    echo "Estatus de los servicios instalados: "
    echo
    echo "1) Estatus del Servicio LAMP."
    echo "2) Estatus del Servicio Samba."
    echo "3) Estatus del Servicio NO-IP."
    echo "4) Estatus del Servicio Transmission."
    echo "5) Volver atras."
    echo
    echo -n "Indique una opcion: "
}
until [ "$opc" = "5" ];
do
    case $opc in

        1)

            echo "-----"
            echo "-      Estatus Servicio LAMP      -"
            echo "-----"
            echo
            sudo /etc/init.d/apache2 status
            echo
            echo -----
            echo
            sudo /etc/init.d/mysql status

            echo
            _MENU
            ;;

        2)

            echo "-----"
            echo "-      Estatus Servicio SAMBA      -"
            echo "-----"
            echo
            sudo /etc/init.d/samba status

            echo
            _MENU
            ;;
```

3)

```
echo "-----"
echo "-      Estatus Servicio NO-IP      -"
echo "-----"
echo
```

```
sudo /usr/local/bin/noip2 -S
```

```
echo
_MENU
;;
```

4)

```
echo "-----"
echo "-      Estatus Servicio Transmission      -"
echo "-----"
echo
```

```
sudo /etc/init.d/transmission-daemon status
```

```
echo
_MENU
;;
```

5) `sudo ./install.sh` ;;

```
*)
clear
_MENU
;;
```

```
esac
read opc
done
```

Comprobacion.sh:

```
#!/bin/bash

var1=`ls /var/lib/transmission-daemon/downloads | wc -l`

#echo "copiada var 1"

sleep 5

var2=`ls /var/lib/transmission-daemon/downloads | wc -l`

#echo "Variables copiadas"

if [ $var2 -gt $var1 ];
then
    #echo "mensaje enviado 2"
    sudo python /home/pi/AplicacionP2P/mensaje.py
    sleep 5
    cd /home/pi/AplicacionP2P
    ./comprobar.sh
else
    #echo "No hay nuevo fichero"
    cd /home/pi/AplicacionP2P
    ./comprobar.sh
Fi
```

Mensaje.py:

```
#Importamos la libreria
import urllib2
import json
#Esta es la funcion que usara el script para enviar las notificaciones
def sendNotification(token, channel, message):
    data = {
        "body" : message,
        "message_type" : "text/plain"
    }
    req = urllib2.Request('http://api.pushetta.com/api/pushes/{0}'.format(channel))
    req.add_header('Content-Type', 'application/json')
    req.add_header('Authorization', 'Token {0}'.format(token))

    response = urllib2.urlopen(req, json.dumps(data))

#Enviamos la notificacion en formato '<strong>Apikey+Canal+Mensaje</strong>'
sendNotification("fb232cf502fda3c39c9107cb32f0b6573877798a", "mensajes", "La descarga
ha finalizado con exito!")

#Avisamos al usuario de que ya se ha enviado el mensaje
print "Mensaje enviado!"
```