

**ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED**

**TRUNK DE COMUNICACIONES ENTRE DOS PAÍSES USANDO  
RASPBERRY PI.**

**YUSEF AISSA MIMON**

## Indice

Introducción y justificación .....	3 - 4
Objetivos .....	5
Análisis del contexto .....	6
Desarrollo del contenido .....	7
Definición de cada componente del proyecto .....	8
Descripción de software utilizado para botear .....	9
Boteo de Raspbx en tarjeta sd y Configuración de red .....	10
Creación de extensiones locales y comprobación .....	11 - 14
Configuración necesaria para realizar llamadas externas .....	15 - 16
Comprobación llamadas externas .....	17
Explicación de que es un trunk SIP .....	18 - 19
Configuración necesaria para realizar el trunk SIP entre dos raspberry .	20 – 21
Conclusión y valoración personal .....	22
Bibliografía y fuentes de consulta.....	23

## **Introducción y justificación**

El proyecto que voy a realizar se denomina trunk 3g, este proyecto es algo que me llamo la atención desde el primer día ya que teniendo en cuenta la zona geográfica donde se localiza Melilla es un gran beneficio por que al estar cerca de Marruecos donde suelo pasar largos periodos de tiempo y teniendo en cuenta el coste que me supone estar comunicado con mi familia esta era una gran oportunidad para llevar a cabo este proyecto.

El gran beneficio de este proyecto es como he dicho me permite estar comunicado de manera permanente con mi familia mientras estoy en marruecos prácticamente gratis, es decir sin ningún coste ya que con las tarifas existentes en las compañías telefónicas actuales la llamada a un numero de la misma compañía sale gratis por lo que prácticamente la comunicación en principio tiene un coste cero.

Este proyecto es un reto pero a la vez es algo con lo que se aprende bastante en cuanto a como funciona algo tan habitual como las centralitas de teléfono que aunque seamos ajenos a ello es lo que nos permite estar comunicados con nuestros familiares, compañeros, amigos estemos donde quiera que estemos en cualquier punto de la tierra.

Es un beneficio para nosotros ya que en su momento revoluciono el mundo de la tecnología de la comunicación esto de poder comunicarse con otras personas en cualquier parte de la tierra ya que antes de esto la manera de comunicarse solía ser a través de cartas y estas podían tardar bastante tiempo en llegar al destinatario.

Como ya he dicho anteriormente lo que me hizo inclinar la balanza a favor de este proyecto fue el gran beneficio que suponía para mi poder realizar lo, además de que su realización no supone un gran coste sino que por muy poco dinero se puede montar uno su propia centralita de teléfono. Pero no todo son ventajas y aunque esto suene tan bonito a las compañías telefónicas

no les hace demasiada gracia ya que es una competencia directa en cuanto a los servicios que ofrecen, es decir puedo beneficiarme de los mismos servicios que me ofrecen sin tener que estar pagando cada mes a estas compañías por esto servicios lo que no les agrada bastante y teniendo en cuenta el poder de estas multinacionales pueden hacer que las políticas de un país cambien en su beneficio, si eso es el poder del dinero.

Por lo que estas compañías telefónicas no permiten que se lleven a cabo estas prácticas para su funcionamiento habitual, es decir si como un proyecto, algo temporal pero no como algo habitual o comercial.

Es más estas compañías están al acecho, es decir si detectan cualquier tipo de anomalía relacionado con este tema bloquean estos servicios y que no te pillen por que no les importa denunciar y llevarte a los tribunales y si eso ocurre mal tema.

Como ya he dicho anteriormente estas compañías telefónicas tienen gran poder, estamos hablando de grandes multinacionales que son capaces de cambiar las políticas de un país en su beneficio así que nos enfrentamos a una gran sanción en caso de utilizar este proyecto de manera comercial.

## Objetivos

Últimamente y mas en estos últimos tiempos se ha oído hablar mucho de la tecnología VoIP y quien no a utilizado esta tecnología, tenemos un ejemplo en algo que se ha convertido en algo tan habitual como skype.

Gracias a estas tecnologías podemos llegar a ahorrar una gran cantidad de dinero en cuanto a la comunicación y esto es nuestro objetivo, poder comunicarnos por muy poco y en algunos casos totalmente gratis.

Asterisk es una aplicación desarrollada en un principio para linux lo que echaba a mucha gente para atrás, Sin embargo, desde hace no mucho tiempo tenemos **Asterisk NOW**, una distribución Linux autoinstalable que nos permite tener operativo una máquina con Asterisk en una media hora.

Por lo tanto mi objetivo realizando este proyecto es poder comunicarme con mi familia sin altos costes de dinero desde cualquier parte del mundo, principalmente Marruecos que es donde suelo pasar mas tiempo. Otro de los objetivos por lo que he realizado este proyecto es que supone un reto ya que es algo nuevo para mi además de obtener mas conocimiento en este campo de la informática.

Para llevar a cabo mi objetivo necesitare configurar el entorno según mis necesidades, algo que, sin duda es un reto para mi ya que no dispongo de mucho conocimiento en el funcionamiento de centralitas de teléfono además es algo complejo y requiere de bastante investigación.

Pero gracias a Internet y a la ayuda de los profesores se hace mas fácil poder hacerlo aunque no siempre se encuentra concretamente lo que uno busca ya que el entorno de cada persona que sube información a Internet es diferente, lo que conlleva un poco más de investigación para poder adoptar estos servicios a las necesidades de cada uno.

### **Análisis del contexto**

En cuanto al análisis del contexto como he dicho anteriormente Melilla se encuentra en una zona estratégica geográficamente hablando, es más junto a Ceuta se les llama la puerta de Europa. Situada en el norte de África a orillas del mar Mediterráneo y separada de la península por este mar a unos 184,4 km de Motril (Granada) es una zona que viene como anillo al dedo para este proyecto ya que los habitantes de Melilla solemos viajar constantemente ya sea a la península o al país vecino lo que conlleva un gasto adicional en cuanto a la comunicación con nuestros familiares aunque teniendo en cuenta las promociones de las compañías telefónicas este gasto se ha reducido bastante en cuanto a unos años atrás.

Teniendo en cuenta todas estas condiciones como ya he dicho este proyecto nos viene como anillo al dedo ya que estamos situados entre la península y Marruecos.

Esta situación geográfica en la que se encuentra Melilla a propiciado que las familias parte este situada en el norte de África y parte en la península y en Marruecos como es mi caso, además de que no puede haber un lugar mejor para implantar este proyecto ya que aquí se le sacaría el máximo potencial.

## **Desarrollo del contenido**

El proyecto trunk 3g no consta de bastante contenido ya que no requiere de mucho trabajo sino que mas bien requiere de mucha investigación. Para llevar a cabo este proyecto sera necesario el siguiente material:

- Dos Raspberry Pi
- Dos USB 3G/GPRS/EDGE modem
- Dos tarjetas SIM (una de cada país).
- Software llamado Asterisk

Para conocer mejor este proyecto voy a hablar un poco de cada componente:

### **- Que es la Raspberry-Pi? ¿Que utilidad tiene?**

La famosa Raspberry pi es una pequeña computadora de bajo coste desarrollada en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

En realidad, se trata de una diminuta placa base de 85 x 54 milímetros (del tamaño aproximado de una tarjeta de crédito) en el que se aloja un chip Broadcom BCM2835 con procesador ARM hasta 1 GHz de velocidad, GPU VideoCore IV y 512 Mbytes de memoria RAM (Las primeras placas contaban con sólo 256MB de RAM). Para que funcione, necesitamos de tarjetas de memoria SD o microSD), conectarlo a la corriente utilizando cualquier cargador microUSB de al menos 1000mah para las placas antiguas y de al menos 2000mah para las modernas. En cuanto a la conexión de red, dispone de un puerto Ethernet (los modelos A y A+ no disponen de puerto Ethernet) para enchufar un cable RJ-45 directamente al router o podemos utilizar cualquier adaptador inalámbrico.

En cuanto al software la Raspberry Pi usa generalmente sistemas operativos basados en núcleo Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, se lanzó durante julio de 2012 y es la distribución recomendada por la fundación para iniciarse. Para mi proyecto esta maravillosa computadora actuará como un servidor de

comunicaciones VoIP, es decir una centralita de teléfono, para lo cual utilizare una distribución creada específicamente para esta función, es decir para una centralita de teléfono. Esta distribución se denomina Raspbx la cual trae Asterisk, nuestra centralita, ya integrada, además de FreePbx para poder administrar nuestra centralita via web y de manera remota..

**Que es Asterisk?** Asterisk es un software de código abierto que puede convertir un simple ordenador como la Raspberry-Pi en un sofisticado servidor de comunicaciones VoIP.

Lanzado hace 10 años y constantemente desarrollada por la comunidad open source, Asterisk se ha convertido en uno de los servidores de comunicaciones más ricos en funcionalidades, escalables y sofisticados de los disponibles en la actualidad.

- Los usb/módem 3G son un pequeño aparato que permite acceder a Internet a través del ordenador cuando no se dispone de ninguna conexión a Internet o cuando no se dispone de Wi-Fi. El módem USB 3G, una especie de módem inalámbrico de tipo Wi-Fi, utiliza la red de los operadores de telefonía para conectarse a Internet.

- En cuanto a las tarjetas sd todos sabemos que son. Son un pequeño dispositivo flash que permite almacenar todo tipo de archivos. Estas tarjetas fueron diseñadas principalmente para las cámaras fotográficas y otros dispositivos de este tipo aunque con el tiempo se ha ido expandiendo su uso hasta llegar a ser imprescindibles para muchos dispositivos que utilizamos en la actualidad como por ejemplo teléfonos móviles que no disponen de gran capacidad de almacenamiento etc...

Ahora que ya conocemos el material con el que voy a trabajar resulta mas fácil comprender la función de cada componente. Lo realmente increíble de la informática por lo menos en mi opinión es lo mucho que se puede hacer con tan poco y pongo de ejemplo este proyecto. Con un pequeño ordenador no muy potente, unos cuantos componentes tan simples como una tarjeta sd y dos módem 3G, además del software Asterisk podemos proveer servicios de telefonía VoIP o usarlo para beneficio propio.



Bien, ahora que ya que conocemos en que consiste el proyecto y conocemos los componentes que lo formaran es hora de comenzar a trabajar. Manos a la obra.....

Lo primero que necesitamos antes de comenzar ha hacer nada es una tarjetas sd en la cual tendremos la distribución Raspbx que como he dicho antes contiene Asterisk, es decir nuestro servidor centralita de teléfono.

Esta distribución podemos encontrarla en la web oficial:

<http://www.raspberry-asterisk.org/downloads/>

Una vez que tenemos Raspbx y todo el material necesario para comenzar, en nuestro ordenador descargaremos dos herramientas **SDFormatter**(opcional) y **win32DiskImager**.

Estas herramientas son las que nos permitirán botear raspbx en nuestra tarjeta sd. SDFormatter es una herramienta muy eficaz cuya función principal es formatear la tarjeta sd, pero pensareis por que una herramienta para formatear si se puede realizar en Windows 8 o Ubuntu sin necesidad de ningún programa, bien muchas veces las tarjetas sd suelen estar corruptas o estar infectadas por virus, en algunos casos como me paso a mí el sistema suele mostrar mucho menos espacio de almacenamiento que el que en realidad tiene la tarjeta y esto es un problema ya que no nos permite cargar nada y el sistema ya sea Windows 8 o Ubuntu no tienen la capacidad de solucionar este problema o por lo menos yo no conozco ninguna manera de hacerlo con estos sistemas.

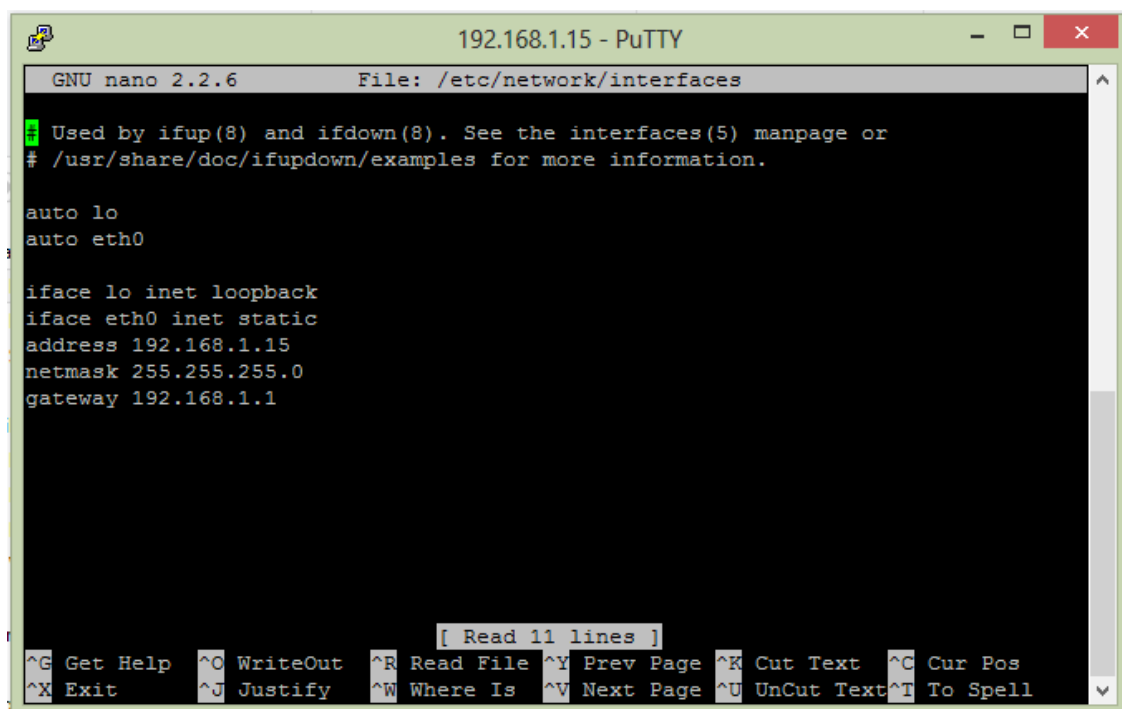
Para solucionar este problema buscando por Internet encontré esta maravillosa herramienta que con solo dos pasos me permitió recuperar todo el espacio de la tarjeta y continuar trabajando.

En cuanto a win32DiskImager es una herramienta cuya función es botear imágenes con extensión .img en tarjetas sd, usb y otros dispositivos que permiten esta función. Es una herramienta fácil de usar y que no requiere mucho conocimiento para poder usarla.

Ahora que sabemos para que sirve cada herramienta vamos a botear raspbx en la tarjeta sd, para ello formateamos la tarjeta, preparamos la imagen y abrimos la herramienta win32DiskImage y una vez abierta seleccionamos la imagen y la tarjeta sd y pulsamos sobre write. En unos minutos listo ya tenemos raspbx en nuestra tarjeta sd.

Una vez que ya tenemos el sistema y arrancamos el siguiente paso configurar la red de nuestro servidor asignándole una ip fija, de esta manera podemos indicar a los teléfonos sip el servidor al que deben conectarse para poder registrarse, además también nos sirve para conectarnos de manera remota a través de Freepbx.

Para asignar la ip fija abrimos el siguiente archivo /etc/network/interfaces y cambiamos la linea donde pone dhcp por static y añadimos la ip que queremos, la mascara subred y la puerta de enlace.



```
192.168.1.15 - PuTTY
GNU nano 2.2.6 File: /etc/network/interfaces
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.

auto lo
auto eth0

iface lo inet loopback
iface eth0 inet static
address 192.168.1.15
netmask 255.255.255.0
gateway 192.168.1.1

[ Read 11 lines ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Una vez que ya tenemos nuestro servidor asterisk con una ip fija ya podemos comenzar a trabajar pero antes de nada actualizamos los repositorios con apt-get update y el sistema con raspbx-upgrade, este paso suele llevar bastante tiempo ya que en la actualización del sistema hay que descargar bastantes paquetes e instalarlos.

Después de esto vamos a comenzar a trabajar con asterisk, para ello me conecto remotamente vía web para proceder a configurar extensiones locales para comprobar que el servidor realiza su función. Para añadir nuevas extensiones, en la página principal de asterisk seleccionamos la pestaña que dice aplicaciones y en el menú desplegable seleccionamos la opción extensiones.

Se nos abrirá un formulario que según nuestras necesidades rellenaremos unos parámetros u otros.

Los parámetros más significativos son los que nombraré a continuación.

- **Port:** es el puerto que utilizará Asterisk para escuchar conexiones SIP entrantes. El puerto SIP por defecto es el 5060.
- **Binaddr:** interfaz de red que utilizará Asterisk para atender las conexiones SIP entrantes. El valor por defecto es 0.0.0.0, es decir, escucha en todas las interfaces de red disponibles.
- **Context:** contexto que se aplica a las llamadas entrantes. Se aplica cuando una cuenta SIP no tiene definido un contexto.
- **Nat:** habilitamos o deshabilitamos el soporte de nat para la extensión.
- **Secret:** Contraseña para proteger nuestras llamadas.
- **Type:** define la clase de conexión que tendrá el cliente. Hay tres tipos de clientes SIP:
  - peer: solo puede recibir llamadas.
  - user: solo puede realizar llamadas.
  - friend: puede recibir y realizar llamadas.

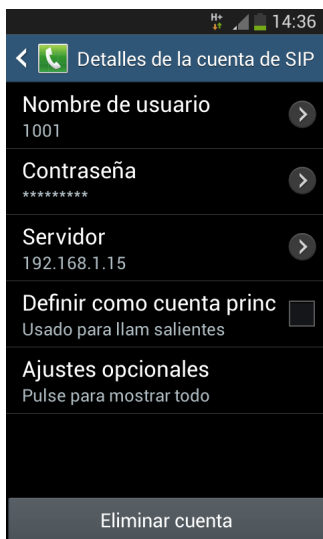
- **Permit:** Permitir una IP.
- **Deny:** No permitir una IP.
- **Allow:** permite habilitar un codec. Pueden ponerse varios en un mismo. usuario Posibles Valores: "allow=all" ,"allow=alaw", "allow=ulaw", "allow=g723.1" ; allow="g729" , "allow=ilbc" , "allow=gsm".
- **Insecure :** Define como manejar las conexiones con peers Tiene los siguientes valores very, yes, no, invite, port por defecto es "no" que quiere decir que hay que autenticarse siempre.
- **Incominglimit:** Limite de llamadas simultaneas para un cliente.
- **Restrictcid:** Se usa para esconder el ID del llamante. Anticuada y en desuso.
- **Qualify:** Para determinar cuando el dispositivo puede ser alcanzado.
- **Defaultip:** IP por defecto del cliente **host=** cuando es especificado como "dynamic".

Una vez que tenemos las extensiones creadas lo siguiente es descargar una aplicación en nuestro ordenador para que funcione como un softphone. Seguidamente en esta aplicación registramos una de las extensiones creadas y en otro teléfono registramos la otra extensión creada.

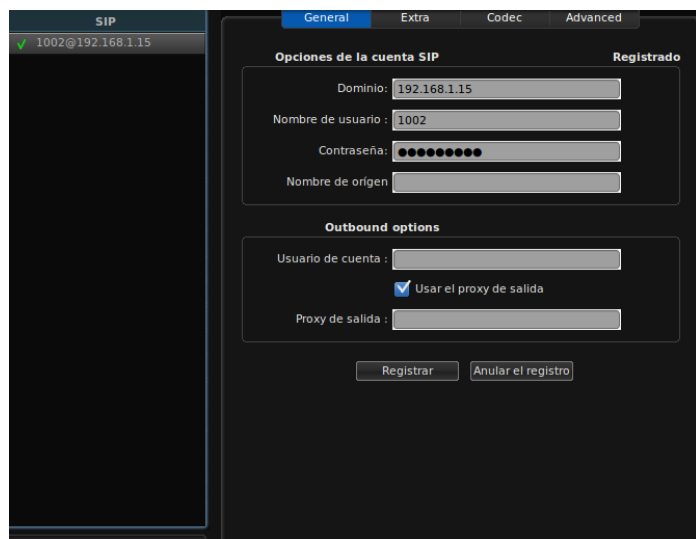
En esta imagen se puede ver las extensiones que he creado para comprobar las llamadas locales y las llamadas externas.

Es importantes recordar que cuando estemos creando extensiones para llamadas locales el parámetro nat debe estar en Yes al igual que para las extensiones externas ya que estas extensiones estarán detrás de un router. A continuación registramos las extensiones como ya he dicho en los softphone y comprobamos que funcionan las llamadas locales, este tipo de llamadas son las que se realizan dentro de una misma red, es decir detrás del mismo router.

### Teléfono móvil



### Aplicación Zoiper PC



Ahora ya tenemos los dos teléfonos registrados con una extensión en nuestro servidor asterisk.

Lo siguiente es comprobar que funcione las llamadas locales para lo cual he realizado las siguientes capturas de pantalla.



Ahora que ya sabemos que las llamadas locales funcionan, ahora lo siguiente es comprobar que podemos realizar llamadas externas, es decir llamar a una extensión que no este fuera de nuestro dominio. Esta parte es un poco mas complicada que llamadas locales ya que tenemos un problema tanto con los puertos como por la ip pública de nuestro router. Pero todo se puede arreglar.

Para comenzar voy a intentar solucionar el primer problema que se presenta que es la ip pública ya que estas ip's suelen ser dinámicas y cada vez que se reinicia el router esta cambia de ip y no podemos andar cambiando la ip a nuestro softphone donde tenemos registrada la extensión.

Para esto lo que haré será crear una cuenta no ip que me permita asignar un dominio a mi ip pública de manera que no tenga que estar cambiando la ip al softphone cada vez que tenga que reiniciar el router, no ip lo que hace es que asigna un dominio al la ip pública y cada vez que esta cambia no ip se actualiza y asigna a nuestro dominio la nueva ip que coge el router. Para esto en la pagina <http://www.noip.com/> podemos crear nuestra cuenta no ip.

Want to help us with new product development? [Take our survey](#) and receive \$5 off any service.

**Hosts/Redirects**

- Add Host
- Manage Hosts
- Manage Groups
- Download Client
- Upgrade to Enhanced

**Need Help?**

- Support Center
- Troubleshooting Guide
- Dynamic Update Client

### Manage Hosts

Current Hosts: 1 [Need More Hosts? Enhance Your Account!](#) [Enhance Your Account](#)

Host	IP/URL	Action
<b>Hosts By Domain</b>		
ddns.net		
centralitayusef.ddns.net	83.57.165.6	<a href="#">Modify</a> <a href="#">Remove</a>

[Add A Host](#)

[Add Google Apps to your Domain](#)

Ahora que ya tenemos nuestra cuenta no ip, en nuestro raspberry instalamos el cliente. Para instalar el cliente es muy sencillo simplemente copiamos y pegamos en nuestra terminal el siguiente comando:  
[wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz](http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz)

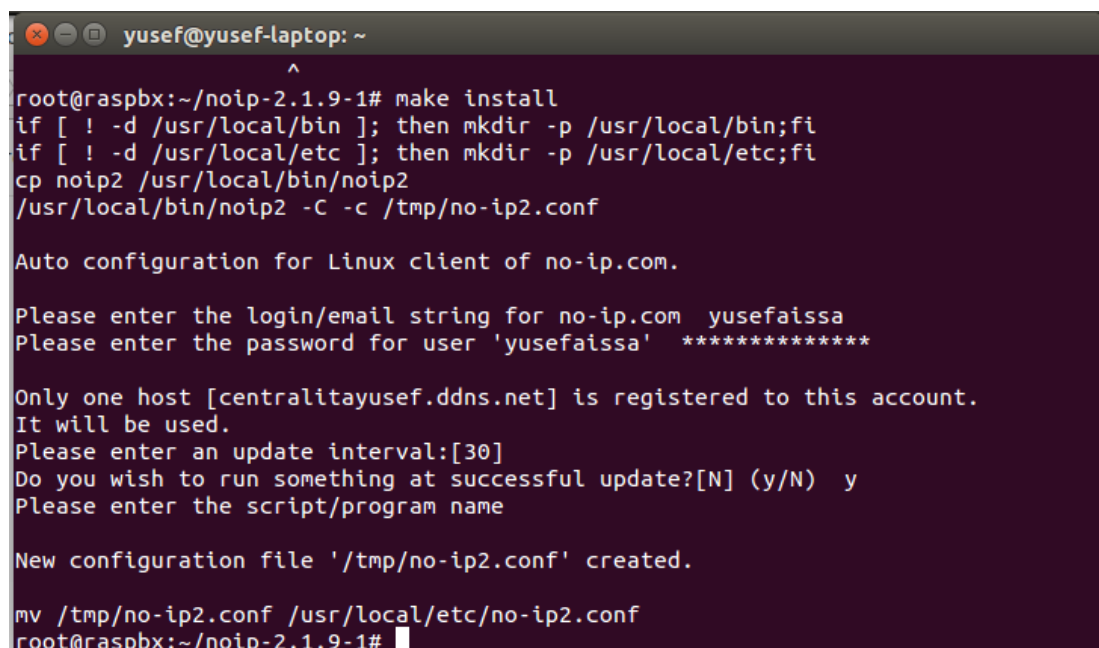
Esto lo que hará será descargar el paquete. Ahora lo siguiente es descomprimir este paquete, una vez descomprimido nos situamos en ese directorio, es decir dentro de la carpeta descomprimida y ejecutamos el comando `make` y `make install`.

Al ejecutar estos comandos nos pedirá que introduzcamos nuestro inicio de sesión o email y la contraseña y de esta manera cogerá automáticamente nuestro dominio y lo asociará a nuestra raspberry, además de añadir las siguientes líneas al archivo `/usr/local/bin/noip2`:

```
#!/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
```

Esto lo que hace es que permite que no-ip se auto-inicie con la Raspberry.

Captura de la instalación del cliente no ip en la raspberry conectado remotamente a través de ssh.



```
yusef@yusef-laptop: ~
^
root@raspbx:~/noip-2.1.9-1# make install
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.

Please enter the login/email string for no-ip.com yusefaissa
Please enter the password for user 'yusefaissa' *****

Only one host [centralitayusef.ddns.net] 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) y
Please enter the script/program name

New configuration file '/tmp/no-ip2.conf' created.

mv /tmp/no-ip2.conf /usr/local/etc/no-ip2.conf
root@raspbx:~/noip-2.1.9-1#
```

Bien hemos solucionado un problema, lo siguiente es abrir los puertos:

- 5060, puerto por el que escucha Asterisk
- Del 10000 al 20000, puertos de voz y datos.

Ahora debemos dirigirnos a nuestro navegador y acceder a nuestro router a través del portal alejandra si se trata de router telefónica como es mi caso. Una vez dentro procedo a abrir los puertos 5060, del 10000 al 20000 e indico que rediriga todo el tráfico que venga de estos puertos a mi raspberry.

**Apertura de puertos**

Introduce los valores para la apertura de un nuevo puerto:

Protocolo

IP Interna

TCP

Selecciona el tipo de apertura de puerto que más te convenga.

☐ Puerto único ☒ Rango de puertos

Introduce a continuación los puertos interno y externo:

Puerto Interno Inicio

Puerto Externo Inicio

Puerto Interno Fin

Puerto Externo Fin

Añadir

**Gestión de puertos**

Protocolo	Ip	Interno Inicio	Interno Fin	Externo Inicio	Externo Fin	Puerto Único	
UDP	192.168.1.15	10000	10000	10000	10000	<input type="checkbox"/>	<input type="checkbox"/>
UDP	192.168.1.15	20000	20000	20000	20000	<input type="checkbox"/>	<input type="checkbox"/>
UDP	192.168.1.15	5060	5060	5060	5060	<input type="checkbox"/>	<input type="checkbox"/>

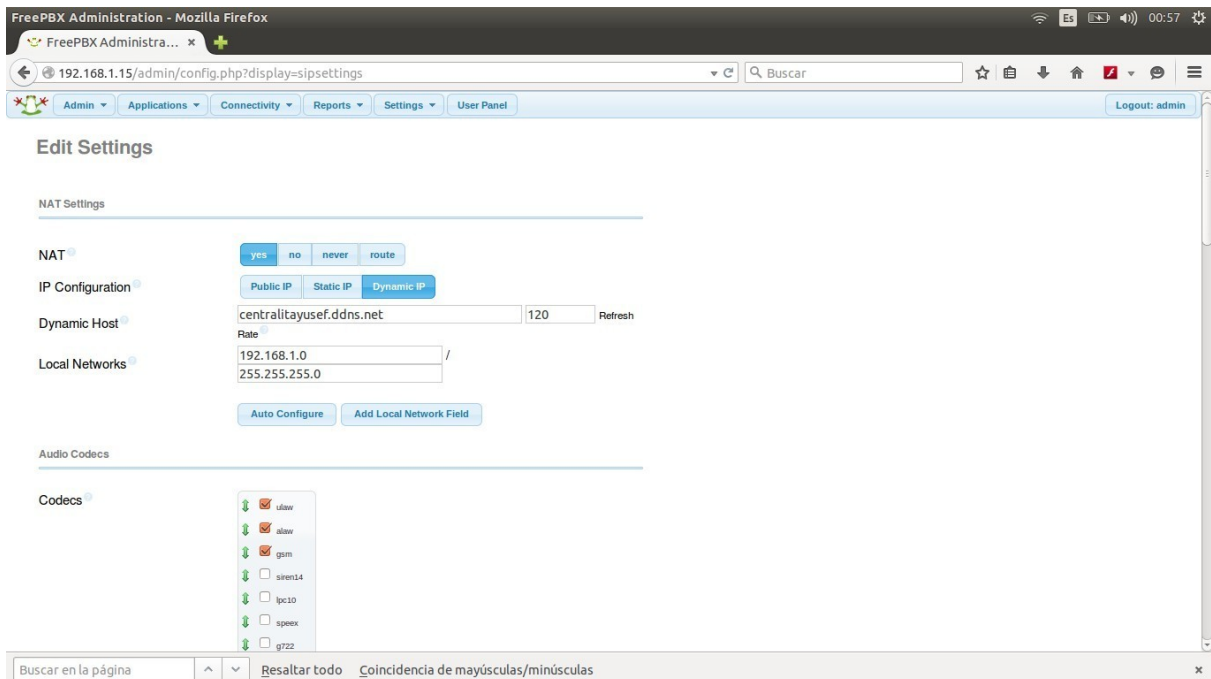
Eliminar

Aceptar Cancelar

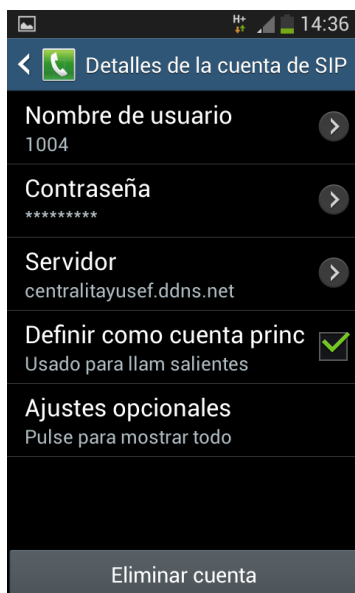
Otro problema menos, ya solo falta configurar asterisk y crear una extensión para comprobar que las llamadas externas funcionan. En freepbx pulsamos sobre la pestaña setting y seleccionamos la opción Asterisk sip Configuration, se abrirá un formulario donde seleccionaremos la opción dynamic ip lo que abrirá otro formulario con tres campos, dos de estos campos local networks y mascara subred se generan automáticamente mientras que en el campo dynamic host indicamos nuestro no-ip.



En la siguiente imagen se puede ver la configuración.



Bien, ya tenemos configurado todo lo necesario para realizar las llamadas externas. Para comprobar que funciona he creado una nueva extensión y la he registrado en mi teléfono móvil.



Hasta aquí todo bien. Pero ahora es cuando empiezan los problemas. Yo tenía pensado hacer el proyecto 3G pero no he podido comprar los módem 3G y en el instituto no quedaban, entonces decidí realizar el mismo proyecto pero sin los módem 3G, si no haciendo un trunk entre las dos raspberrys al numero virtual que tendría cada una, pero de nuevo había otro problema y es conseguir un numero virtual lo que cuesta dinero y no he podido conseguirlos.

Después de todos estos problemas pensé en otra alternativa y es poner una ip detrás de otro router en otra casa y hacer un trunk con la raspberry que tengo en mi casa.

Para poder hacer el trunk entre las dos raspberry tenía que configurar la segunda raspberry.

La configuración sería prácticamente igual que en la primera raspberry salvo que cambiarían un par de cosas. Primero la ip de la raspberry y segundo crear una cuenta no ip para la segunda raspberry y abrir los puertos en el otro router de la misma forma que en el primer router con la primera raspberry.

Una vez instalado raspbx en la segunda raspberry tengo que configurar el adaptador de red al igual que en la primera raspberry.

Una vez que ya tengo una ip fija actualizo los repositorios con apt-get update y el sistema con raspbx-upgrade.

Lo siguiente es crear unas extensiones para comprobar que funcionan de manera correcta tanto llamadas locales como llamadas externas. Para poder realizar las llamadas externas al igual que en la primera raspberry tuve que crear una cuenta no ip, instalar el cliente no ip en la raspberry, abrir los puertos y configurar asterisk principalmente en el apartado Sip Asterisk Settings como en la primera raspberry y por último abrir los puertos y redireccionar el tráfico de estos puertos hacia la raspberry.

Ahora que ya esta todo configurado solo falta comprobar que funcionen las llamadas locales y externas registrando dos extensiones diferentes una en cada softphone y realizar llamadas, todo esto es igual que en la configuración mostrada en las páginas anteriores.

Bien nuestra segunda raspberry ya cumple su función y permite realizar tanto llamadas locales como llamadas externas.

Ahora ya tenemos nuestras dos raspberry funcionando correctamente cada una en una casa diferente y en redes diferentes.

Es hora comenzar a realizar el trunk entre los dos servidores asterisk. Para llevar a cabo este paso debemos realizar un trunk SIP, lo que haré a traves de la terminal, es decir para este paso no utilizare Freepbx e iré poniendo la configuración que debe ir en cada archivo.

Estos archivos son dos principalmente y se llaman sip.onf y extensions.conf, la configuración la realizaré en mis dos servidores asterisk.

- El archivo sip.con es un archivo que sirve para configurar todo lo relacionado con el protocolo SIP y añadir nuevos usuarios o conectar con proveedores SIP.

- El archivo extensions.conf es el más importante del Asterisk y tiene como misión principal definir el dialplan o plan de numeración que seguirá la centralita para cada contexto y por tanto para cada usuario.

Estos dos archivos son fundamentales en asterisk, es más podría decirse que son como el motor de un coche, por la tanto antes de comenzar a realizar ninguna configuración en estos archivos he pensado en realizar una copia de seguridad para cada archivo, así en caso de algún error que no pueda solucionar o no sepa su origen simplemente restaurando estos archivos en principio todo volverá a estar en su sitio excepto si el problema viene de algún archivo que no sea uno de estos dos.

Bien, es hora de empezar a trabajar. Comenzare por el servidor asterisk con ip 192.168.1.15 que es el primero que realice y el que tengo en mi casa.

Para trabajar más cómodo los voy a llamar Asterisk A(192.168.1.15) y Asterisk B(192.168.1.16).

Para empezar hay que abrir el archivo sip.conf que se encuentra en /etc/asterisk/sip.conf. Ahora en este archivo agrego la siguientes líneas:

```
[general]
nat = yes
externip=ip_externa_Asterisk A (centralitayusef.ddns.net)
localnet=192.168.1.0/255.255.255.0
externrefresh=180

register => Asterisk A: 1234@ip\_externa\_Asterisk B(centralitayusefAstB.ddns.net)/IP Asterisk B(192,168,1,16)
```

```
[1001]
username=1001
type=friend
secret=1234
callerid= "Yusef"<1001>
host=dynamic
nat=yes
disallow=all

allow=ulaw
context=internal
pickupgroup=1
callgroup=1
```

```
[Asterisk B] (debajo de 1001)
username=AsteriskSedeB
type=friend
secret=1234
context=internal
qualify=yes
host=dynamic
language=es

insecure=invite
disallow=all
allow=ulaw
allow=alaw
```

Abrimos el archivo /etc/asterisk/extensions.conf y ponemos las siguientes líneas:

```
[internal]
exten => _10xx,1,Dial(SIP/${EXTEN},15,tT)
exten => _10xx,n,Hangup

exten => _20xx,1,Dial(SIP/\${EXTEN}@Asterisk B,10)
exten => _20xx,n,Hangup()
```

Una vez echo esto ya solo debemos ir a nuestro asterisk y cargar la configuración con  
sip reload y dialplan reload

Una vez hemos acabado con el Asterisk A nos vamos al servidor Asterisk B y  
hacemos lo mismo.

Abrimos el archivo sip.conf que se encuentra en /etc/asterisk y añadimos las  
siguientes líneas.

[general]

nat=yes

externip=ip\_externa\_ASTERISK\_B(centralitayusefAstB.ddns.net)

localnet=192.168.1.0/255.255.255.0

register => Asterisk B:1234@ip\_externa\_Asterisk\_A/Ip Asterisk A(192,168,1,15)

[2001]

username=2001

type=friend

secret=1234

callerid="Yusef2" <2001>

host=dynamic

nat=yes

disallow=all

allow=ulaw

context=internal

pickupgroup=1

callgroup=1

[Asterisk A] **(Debajo de 2001)**

username=Asterisk A

type=friend

secret=1234

context=internal

qualify=yes

host=dynamic

language=es

insecure=invite

disallow=all

allow=alaw

Abrimos el archivo /etc/asterisk/extensions.conf y ponemos las siguientes líneas

[internal]

exten => \_10xx,1,Dial(SIP/\${EXTEN},15,tT)

exten => \_10xx,n,Hangup

exten => \_80xx,1,Dial([SIP/\\${EXTEN}@Asterisk A](#),10)

exten => \_80xx,n,Hangup()

Una vez echo esto ya solo debemos ir a nuestro asterisk y cargar la configuración con  
sip reload y dialplan reload.

Con esto en principio podríamos llamar desde una extensión de un servidor al otro y  
debería funcionar, a mi no me ha funcionado y no he puesto ninguna comprobación  
ya que no me ha funcionado y no he logrado saber por que, es decir no he  
conseguido hacer el trunk SIP entre las dos raspberry.

## **Conclusión y valoración personal**

En conclusión ha sido un proyecto más complicado de lo que pensaba, ya que me he encontrado con bastantes obstáculos por el camino, principalmente la falta de tiempo y estrés de tener que llevar el proyecto a la par que realizaba las prácticas FCT a las cuales debía dedicar gran parte de mi tiempo lo que me retrasaba en el proyecto.

Aparte de los inconvenientes ha sido muy entretenido y sin duda he aprendido bastante sobre el mundo de las telecomunicaciones y esto me ha echo abrir los ojos ya que antes pensaba que era bastante más fácil de lo que en realidad es.

Personalmente ha sido un gran proyecto en el que yo no he estado a la altura sin importar los obstáculos que me haya encontrado por el camino, en definitiva estoy bastante contento de haber llegado hasta donde he llegado en cuanto al proyecto ya que teniendo en cuenta la falta de tiempo, el estrés y la falta de material que me ha echo cambiar en el último momento de idea he llegado más lejos de lo que pensaba.

## Bibliografía y fuentes de consulta

Las fuentes de las que he obtenido la información son bastantes por lo que pondré las mas significativas.

<http://www.voip-info.org/wiki/view/Asterisk+config+sip.conf>

<http://mviera.io/blog/configurando-los-clientes-sip/>

[http://asteriskdocs.org/en/3rd\\_Edition/asterisk-book-html](http://asteriskdocs.org/en/3rd_Edition/asterisk-book-html)

<http://www.voip-info.org/wiki/view/Asterisk+config+extensions.conf>

<http://www.en.voipforo.com/asterisk/extensions-conf-config.php>

[www.dialone.com.ar/download/manual-freepb-espanol.pdf](http://www.dialone.com.ar/download/manual-freepb-espanol.pdf)

<http://www.freepbx.org/support/documentation/administration-guide/adding-extensions>

<https://www.twilio.com/docs/sip-trunking/sample-configuration>

<http://www.alkia.net/index.php/faqs/137-how-to-configure-sip-trunk-with-asterisk>

Proyecto Acabado el 02/06/2015

