



PROYECTO FINAL

---

# Centralitas VoIP Basadas en Raspberry , Asterisk y FreePBX

---

Grado Superior en  
Administración de  
Sistemas  
Informáticos en Red

Curso Académico  
2014/2015

---

Manuel Alfaya Tello

---

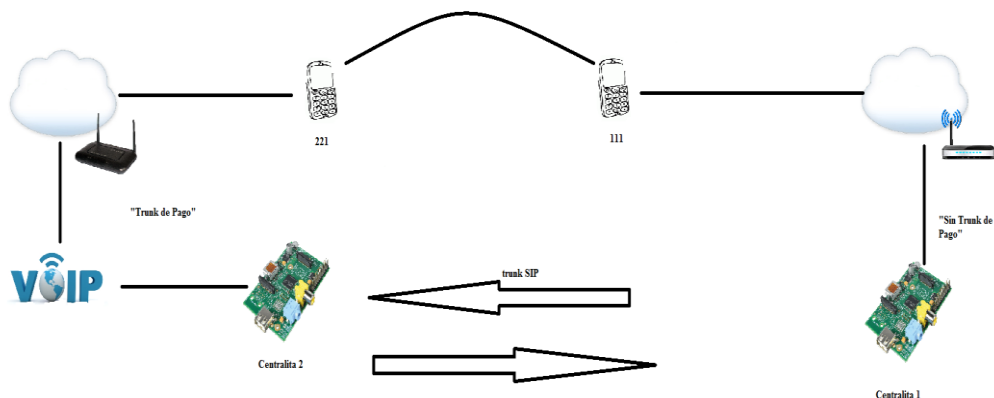
# INDICE

1. <u>Introducción</u>	3
2. <u>Objetivos.</u>	4
3. <u>Conceptos del Proyecto.</u>	5-10
3.1. <u>¿Qué es Asterisk?</u>	5-6
3.2. <u>¿Qué es FreePBX?</u>	6-7
3.3. <u>¿Qué es el VoIP?</u>	7
3.4 <u>¿Qué es telefonía IP?</u>	8
3.5 <u>¿Qué son los protocolos?</u>	8-9
3.6 <u>Protocolo SIP (Session Initiation Protocol)</u>	9
3.7 <u>¿Qué es una Raspberry Pi?</u>	9-10
4. <u>Desarrollo del Proyecto.</u>	11-20
4.1 <u>Configuración de las Extensiones.</u>	12
4.2 <u>Configuración Dominio NO-IP.</u>	12-16
4.3 <u>Habilitar Puertos Router.</u>	16
4.4 <u>Configuración terminales con la extensiones.</u>	17
4.5 <u>Configuración de Trunk de Pago.</u>	18
4.6 <u>Configuración de Trunk SIP entre las dos Centralitas.</u>	18-20
5. <u>Compobaciones.</u>	21

6. <u>Problemas de realización no resueltos.</u>	<u>22</u>
7. <u>Conclusión y Valoración personal del Proyecto.</u>	<u>22</u>
8. <u>Bibliografía.</u>	<u>23-24</u>
9. <u>Fecha Fin de Proyecto.</u>	<u>25</u>

## 1. Introducción

Voz sobre protocolo de internet o voz por protocolo de internet, es un grupo de recursos que hacen posible que la señal de voz viaje a través de internet empleando un protocolo IP (protocolo de internet). Esto significa que se envía la señal de voz en forma digital, en paquetes de datos, en lugar de enviarla en forma analógica a través de circuitos utilizables sólo por telefonía convencional. Con la que nos ayudaremos con las máquinas Raspberry's que con el Sistema operativo que le instalaremos viene con el programa Asterisk que nos trae en esta imagen que le instalaremos un entorno gráfico llamado FreePBX, para que nos sea más cómo trabajar con el programa y para que se vean las dos "Centralitas" las conectaremos por medio de un Trunk SIP para que se conecten entre ellas y una de ellas tendrá un "Trunk de pago" y otra de ellas no. El en siguiente esquema veremos cómo estará representado nuestro proyecto:



### 2. **Objetivos.**

El objetivo de este proyecto es implementar un sistema de llamadas de VOIP basado en Asterisk para la comunicación entre terminales por medio de extensiones creadas en las centralitas de llamadas locales como exteriores, y tener una centralita un Trunk de Pago que tendrá su propia extensión para llamarla y otra que sea sin un trunk de pago para poder llamarla a través de una herramienta gráfica llamada FreePBX.

### **3. Conceptos del Proyecto.**

#### **3.1. ¿Qué es Asterisk?**

Asterisk es un entorno de trabajo de código abierto, creado para diseñar aplicaciones de comunicación. Asterisk convierte un ordenador cualquiera, en un completo servidor de comunicaciones. Además potencia los sistemas PBX IP, las pasarelas VoIP, servidores de conferencias, y mucho más.

Originalmente fue concebido como una plataforma para la generación de un sistema PBX, pero con el tiempo ha ido evolucionando a otro tipo de usos, como Pasarelas VoIP, sistemas integrales para call-centers, salas de conferencias, buzones de voz, y todo tipo de aplicaciones que tengan relación con las comunicaciones en tiempo real como hemos comentado antes.

Comparando Asterisk es para el mundo de las comunicaciones, lo mismo que sería Apache para el mundo de las aplicaciones web. Apache es un servidor web, y Asterisk es un servidor de comunicaciones.

Los servicios telefónicos en las redes de datos está marcando el inicio de la unificación de los principales servicios de una empresa en una sola red y la facilidad para el operario de manejar todos sus recursos desde su terminal de computadora.

Quizá lo más interesante de Asterisk es que reconoce muchos protocolos VoIP como pueden ser SIP, H.323, IAX y MGCP. Asterisk puede interoperar con terminales IP actuando como un registrador y como gateway entre ambos.

Una vez comprendido el nuevo rumbo de las comunicaciones de voz se puede hablar de la importancia que tiene el estar preparado para esta nueva tecnología que ya es una realidad en todo el mundo. Este proyecto de investigación busca entregarlas herramientas necesarias para la implementar la voz sobre IP en las redes de datos existentes y utilizando las actuales redes telefónicas. Y haciendo un trunk entre dos raspberrys con con otros trunks SIP de cada extensión para llamarse entre sí desde el exterior.

Y en este proyecto utilizaremos una herramienta, para poder trabajar con una interfaz gráfica llamada FreePBX, que en la ISO (Imagen del sistema operativo instalado en la Raspberry) incluía los programas como esta aplicación el Raspbian, MySQL, Apache entre otros.



### 3.2.¿Qué es FreePBX?

FreePBX es una interfaz web de usuario que facilita la interacción del usuario con el sistema VoIP Asterisk. Se abstrae en ciertas ocasiones de tareas de cierta complejidad y por tanto resulta muy útil para usuarios o administradores que no estén muy familiarizados con Asterisk, por lo tanto, es muy fácil de usar y con gran capacidad. También está basado en Open Source GPL.



Permite configurar fácilmente un sistema Asterisk, cubriendo los requisitos tanto de pequeñas como de grandes empresas. Puede mantener las bases de datos de usuarios y extensiones, así como todas las funciones de valor añadido. Por citar las más importantes:

- Creación de Extensiones.
- Dialplan de llamadas entrantes y salientes.
- IVR (Recepcionista digital interactiva) – Operadora automática.
- Time conditions – Gestión de llamadas entrantes según horario y fecha.

- Grupo de llamadas (Ring Groups): Round-Robin, todas a la vez, etc.
- ACD – Sistema de colas y agentes.
- Creación de Trunks.
- Monitorización de llamadas.
- Sistema de mensajería vocal.
- Música en espera.
- Sala de Conferencias.
- Grabación de las llamadas (sólo recomendado para pequeños volúmenes).

### 3.3 ¿Qué es el VoIP?

Esto quiere decir que es un grupo de recursos que hacen posible que la señal de voz viaje a través de Internet empleando un protocolo IP. Esto significa que se envía la señal de voz en forma digital, en paquetes de datos, en vez de enviarla en forma analógica a través de circuitos utilizables solo por telefonía convencional como las redes PSTN (Public Switched Telephone Network), Red Telefónica Pública Conmutada.

Inicialmente se implementó para reducir el ancho de banda mediante compresión vocal, aprovechando los procesos de compresión diseñados para sistemas celulares en la década de los años 80; de esta forma se logró reducir los costes en el transporte internacional de llamadas.

Actualmente los sistemas VoIP son ampliamente utilizados en redes de datos de ámbito LAN e Internet, lo que transforma nuestra Equipo en un Terminal telefónico con la adición de un micrófono o auricular. La tendencia natural de los sistemas de voz sobre IP es que las llamadas puedan ser cursadas entre redes IP y redes telefónicas tradicionales, pasando de esta forma a denominarse la VoIP

### 3.4 ¿Qué es telefonía IP?



La telefonía IP es considerada una típica aplicación de VoIP y tiene como meta intentar proveer las mismas características y calidad que la telefonía tradicional (PSTN) Public Switched Telephone Network, dicha telefonía es una red con conmutación de circuitos tradicional optimizada para comunicación de voz en tiempo real. Además el término Telefonía por Internet es regularmente usado como equivalente a telefonía IP pero que utiliza la Internet como red de datos.

En términos simples el proceso para llevar a cabo la VoIP es el siguiente: la voz, la cual es una información de tipo analógica, es codificada digitalmente y convertida en paquetes IP, los cuales son transportados o movidos a través de una red de datos. El movimiento de estos paquetes hacia su destino es realizado a través de uno o varios caminos, gracias al uso de protocolos de enrutamiento. Finalmente en su destino, los paquetes son reensamblados, reordenados y entregados al receptor, en el formato original, es decir, una señal analógica.

Algunas características de VOIP:

- Los datos se dividen en paquetes.
- Varios usuarios pueden utilizar la misma línea.
- Varias conversaciones o datos al mismo tiempo.
- Integración en una sola red de Voz y datos.

### **3.5 ¿Qué son los protocolos?**

El objetivo del protocolo de VoIP es dividir en paquetes los flujos de audio para transportarlos sobre redes basadas en IP. Los protocolos de las redes IP originalmente no fueron diseñados para el fluido del tiempo real de audio o cualquier otro tipo de medio de comunicación.

La PSTN está diseñada para la transmisión de voz, sin embargo tiene sus limitaciones tecnológicas.

Es por lo anterior que se crean los protocolos para VoIP, cuyo mecanismo de conexión abarca una serie de transacciones de señalización entre terminales que cargan dos flujos de audio para cada dirección de la conversación.

**3.6 Protocolo SIP (Session Initiation Protocol):** (Es la que usaremos en este proyecto).

Es un protocolo de señalización para conferencia, telefonía, presencia, notificación de eventos y mensajería instantánea a través de Internet.

- Este protocolo considera a cada conexión como un par y se encarga de negociar las capacidades entre ellos.
- Tiene una sintaxis simple, similar a HTTP o SMTP.
- Posee un sistema de autenticación de pregunta/respuesta.
- Tiene métodos para minimizar los efectos de DoS (Denial of Service o Denegación de Servicio), que consiste en saturar la red con solicitudes de invitación.
- Utiliza un mecanismo seguro de transporte mediante TLS.
- No tiene un adecuado direccionamiento de información para el funcionamiento con NAT.

### 3.7 ¿Qué es una Raspberry Pi?

Raspberry Pi es un ordenador de placa reducida o (placa única) (SBC) de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.



El diseño incluye un System-on-a-chip Broadcom, que contiene un procesador central (CPU) a 700 MHz (el firmware incluye unos modos “Turbo” para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía, un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB). El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente.

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian), RISC OS 5, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora); y promueve principalmente el aprendizaje del lenguaje de programación Python. Otros lenguajes también soportados son Tiny BASIC, C, Perl y Ruby.

Para la realización de este proyecto lo vamos a hacer con dos dispositivos que son Raspberrys instalándoles en ella las respectivas imágenes con los sistemas.

### 4. Desarrollo del Proyecto.

Lo primero que tenemos que hacer es la instalación de la imagen descargada por su página oficial, para instalar e implementar Asterisk a nuestras Raspberrys, una vez descargada procederemos a crear la SD Booteable.

Una vez descargada la imagen comenzaremos a la configuración de Asterisk.

Lo primero que tenemos que configurar es los ajustes básicos como lugar de procedencia que sería con el comando **dpkg-reconfigure locales**, seleccionar la zona horaria **configure-timezone** y como configurar el tipo de teclado que tenemos con el comando **dpkg-reconfigure keyboard-configuration**.

Tras esto lo que tendremos que hacer es configurar la tarjeta de red en la se nos quedará el archivo **/etc/network/interfaces** así:

```
address 192.168.1.xxx
```

```
netmask 255.255.255.0
```

```
network 192.168.1.0
```

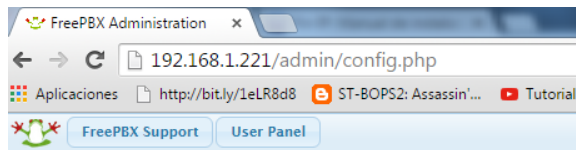
```
broadcast 192.168.1.255
```

```
gateway 192.168.1.1
```

Tras esto le haremos un **raspbx-upgrade** para que actualice el sistema y las aplicaciones descargadas para que tengan las aplicaciones como MySQL, Apache entre otros.

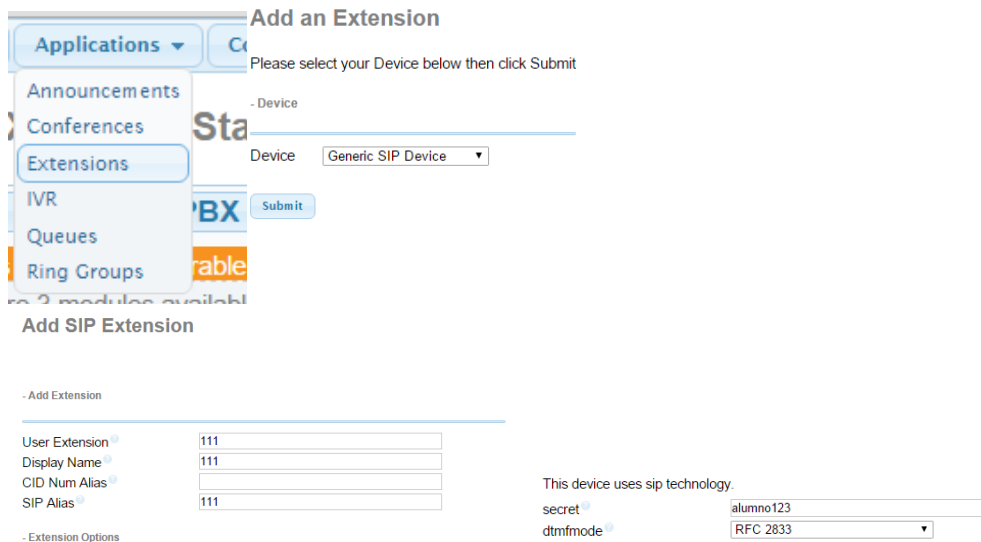
```
root@raspbx:~# raspbx-upgrade
```

Una vez configurado nos iremos a nuestro navegador y pondremos la dirección IP de una de las raspberrys configuradas para acceder al FreePBX (entorno gráfico).

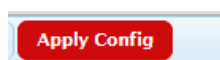


### 4.1 Configuración de las Extensiones.

Procederemos a crear las extensiones SIP que nos pedirá un nombre y una contraseña para la extensión, como vemos en las siguientes imágenes:



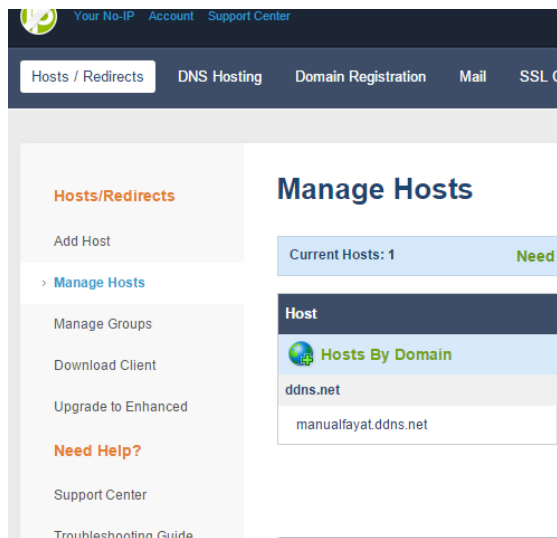
Una vez creadas nuestras extensiones le daremos al botón de aplicar cambios.



### 4.2 Configuración Dominio NO-IP.

Una vez realizado el paso anterior en red local, procederemos a crear un Host para poder hacer llamadas externas, que consiste en crearnos un Host en la página de NO-IP.

Lo primero para realizar esto tendremos que abrir un Dominio NO-IP, que esto nos dará un host que nuestra IP pública que servirá para sincronizar nuestra centralita con nuestro router.



Tras hacer estos pasos nos iremos a nuestra Raspberry Pi y nos meteremos dentro de temporal (**cd/tmp/**) en la que crearemos una carpeta llamada noip2 con el comando de **mkdir noip2** y nos descargaremos el paquete de cliente noip dentro de esa carpeta a partir de un wget (**wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz**). Una vez que lo descargamos lo descomprimos una vez descargado con el siguiente comando: **tar -xvzf noip-duc-linux.tar.gz**.

Una vez descomprimido se nos descomprimirá una carpeta llamada "**noip-2.1.9-1**", accedemos a esta misma y dentro de esta haremos un **make** y luego un **make install** en la que seguiremos los siguientes pasos:

1º Nos pedirá que pongamos el correo que ingresamos en el dominio noip creado y la respectiva contraseña y nos saldrá el dominio noip creado.

**Please enter the login/email string for no-ip.com**

[xxxxxxxxx@correo.com](#)

**Please enter the password for user 'xxxxxxxxx@correo.com'**

**\*\*\*\*\***

Si tienes más de un host deberás elegir cual poner en este server, en mi caso solo tengo un host en esa cuenta

**Only one host [manualfayainfo.ddns.net] is registered to this account.**

**It will be used.**

2º Tras esto nos pedirá introducir el intervalo de tiempo de actualización de ip, es en minutos y nos pone que pongamos [30].

3º Te pregunta que si deseas hacer la actualizacion, ponemos "y".

**Do you wish to run something at successful update?[N] (y/N)**  
**y**

4º Escribimos el nombre del script, podemos poner noip.

**Please enter the script/program name noip**

5º Nos muestra el mensaje de la nueva configuración.

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

**mv /tmp/no-ip2.conf /usr/local/etc/no-ip2.conf**

6º Para saber si está corriendo ponemos y nos muestra una respuesta pondremos el siguiente comando: **ps aux | grep noip2.**

Y con esto ya tenemos configurado nuestro no-ip en asterisk para que se autoactualice. Para correrlo en el arranque agregamos las siguientes líneas en archivo **/etc/rc.local**

```
GNU nano 2.2.6      File: /etc/rc.local

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
/usr/local/bin/noip2
exit 0

[ Wrote 18 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
```

y en `/etc/init.d/noip2`.

```
GNU nano 2.2.6      File: /etc/init.d/noip2

#!/bin/bash
### BEGIN INIT INFO
# Provides: Servicio No-IP
# Required-Start: $syslog
# Required-Stop: $syslog
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: arranque automatico para no-ip
# Description:
#
### END INIT INFO
sudo /usr/local/bin/noip2

[ Read 12 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
```

Tras hacer esto con el siguiente comando `/usr/local/bin/noip2 -S` vemos que está en funcionamiento como muestra la siguiente imagen:

```
/usr/local/bin/noip2 -S
1 noip2 process active.

Process 2089, started as noip2, (version 2.1.9)
Using configuration from /usr/local/etc/no-ip2.conf
Last IP Address set 85.57.20.8
Account manualfayat
configured for:
    host centralitalmanu.ddns.net
Executing /descargaNP/noip-2.1.9-1/noip upon successful update.
Updating every 30 minutes via /dev/eth0 with NAT enabled.
root@raspbx:~#
```



Tras haber configurado el paso anterior, nos meteremos de nuevo en nuestro FreePBX nos iremos a Settings y a SIP settings nos iremos a Dynamic IP y en Dynamic Host pondremos nuestro Dominio NOIP y luego en Local Networks pondremos la IP de la raspberry como vemos en la siguiente imagen:

### Edit Settings

NAT Settings

NAT ☐ yes ☐ no ☐ never ☐ route

IP Configuration ☐ Public IP ☐ Static IP ☒ Dynamic IP

Dynamic Host   Refresh Rate ☐

Local Networks  /

Audio Codecs

### 4.3 Habilitar Puertos Router.

Lo primero que tendremos que hacer es acceder a nuestro router de la red para abrir los puertos del tipo UDP para poder hacer las llamadas desde la red de Datos, hay que abrir el puerto le indicaremos la IP que queremos que le deje y con la otra raspberry desde otra red hacemos lo mismo:

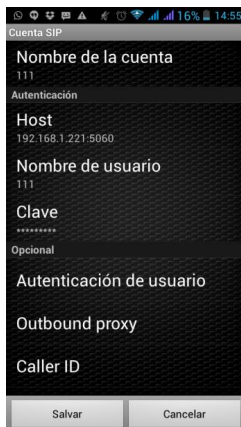
- Puerto 5060: Puerto SIP.
- Puerto 10000: Puerto de Voz y datos.
- Puerto 20000: Puerto de Voz y datos.

Asignación de puertos				
Nombre Asignado	Protocolo	Equipo remoto	Puerto externo inicial	Puerto externo final
raspberrry1	UDP	192.168.1.221	10000	10000
Gmod	TCP/UDP		27015	27015
Garrys Mod	TCP/UDP		27000	27000
raspberrry	UDP	192.168.1.221	5060	5060
raspberrry2	UDP	192.168.1.221	20000	20000

### 4.4 Configuración terminal con la extensiones.

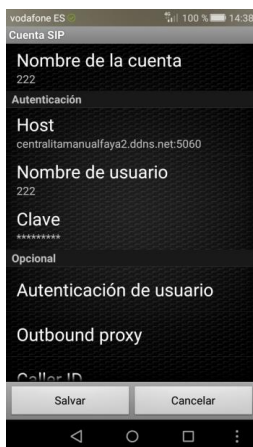
- Configuración de Terminal Local.

Tras hacer esto nos vamos a nuestros dos dispositivos móviles que estén conectados en la misma red que nuestra centralita configurándolo con las extensiones creadas y poniendo el nombre de host con ip:puerto de **Asterisk** que en mi caso he utilizado la aplicación **Zoiper** para realizar la llamada local con 192.168.1.221:5060 como vemos en las siguiente imagen:



- Configuración de Terminal con Datos.

Al hacer el paso anterior, lo siguiente es hacer llamadas desde la conexión de datos , entonces uno de los dispositivos móviles estará en la red wi-fi y el otro a partir de los datos podrá llamar desde fuera de la red wi-fi:



### 4.5 Configuración de Trunk de Pago.

Vamos a configurar el trunk de Pago que en nuestro caso será de Movistar, lo primero que tendremos que crear es un Trunk SIP en la centralita. Aquí le indicaremos en la sección PEER Details, su nombre, si lleva contraseña, etc. Y le pondremos esto:

```
username=84XXXXXXX@telefonica.net
type=peer
session-timers=refuse
secret=XXXX
port=5070
outboundproxyport=5070
outboundproxy=sbc.ngn.rima-tde.net:5070
nat=yes
language=es
insecure=port,invite
host=telefonica.net
fullcontact=84XXXXXXX@telefonica.net
fromuser=84XXXXXXX
fromdomain=telefonica.net
externip=XXX.XXX.XXX.XXX
dtmfmode=rfc2833
disallow=all
allow=alaw,ulaw
```

### 4.6 Configuración de Trunk SIP entre las dos Centralitas.

Una vez hecho este paso, vamos a unir las dos centralitas, tendremos que crear primero un Trunk SIP en cada centralita que queremos unir. Aquí le indicaremos en la sección PEER Details a quien se tiene que conectar indicándole la IP del otro servidor, su nombre, si lleva contraseña, si es interno, etc.

También tendremos que indicarle en la sección USER Details cuál es nuestro propio servidor. Después en el módulo de rutas salientes tendremos que crear una nueva indicándole un DialPlan para que se puedan comunicar y el Trunk que hemos creado.

### Centralita 1

Outgoing Settings

Trunk Name: servidor2

PEER Details:

```
host=192.168.1.222
username=servidor2
type=peer
qualify=yes
context=from-trunk
trunk=yes
disallow=all
allow=gsm
```

Incoming Settings

USER Context: servidor1

USER Details:

```
host=192.168.1.222
type=user
context=from-trunk
disallow=all
allow=gsm
```

### Centralita 2

Trunk Name: servidor1

PEER Details:

```
host=192.168.1.221
username=servidor1
type=peer
qualify=yes
context=from-trunk
trunk=yes
disallow=all
allow=gsm
```

Incoming Settings

USER Context: servidor2

USER Details:

```
host=192.168.1.221
type=user
context=from-trunk
disallow=all
allow=gsm
```

Tras esto tendremos que configurar el Outbound Route para las salidas con los trunks creados y que se vean entre ellas:

Route Settings

Route Name: servidor1

Route CID:

Route Password:

Route Type: ☐ Emergency ☐ Intra-Company

Music On Hold? default

Route Position: --No Change--

Dial Patterns that will use this Route

(  ) +  | [XX] /

(prepend) + prefix | [match pattern] / CallerID

+ Add More Dial Pattern Fields

Dial patterns wizards: (pick one)

Export Dialplans as CSV: Export

Trunk Sequence for Matched Routes

0 servidor1

1

Add Trunk

Optional Destination on Congestion

Route Settings

Route Name: Servidor2

Route CID:

Route Password:

Route Type: ☐ Emergency ☒ Intra-Company

Music On Hold? default

Route Position: --No Change--

Dial Patterns that will use this Route

(  ) +  | [3X] /

(prepend) + prefix | [match pattern] / CallerID

+ Add More Dial Pattern Fields

Dial patterns wizards: (pick one)

Export Dialplans as CSV: Export

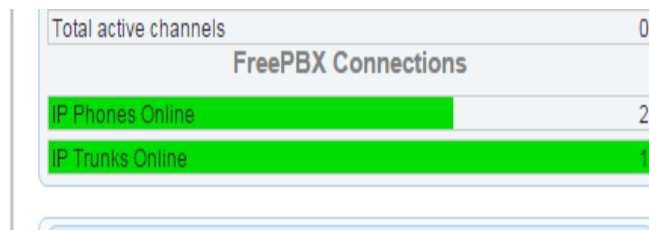
Trunk Sequence for Matched Routes

0 Servidor2

1

Add Trunk

Y como vemos en el panel de control nos dice que el Trunk Online está en buen estado:



También podemos comprobarlo al ejecutar Asterisk desde consola entrando al modo de CLI :

```
root@raspbx:~# rasterisk
```

Y ejecutaremos el siguiente comando para ver si estamos conectados a la otra centralita:

```
raspbx*CLI> sip show peers
Name/username      ACL Port      Status      Description      Dyn Forcerport
Comedia            A 56952      OK (359 ms)      (Unspecified)    D No
111/111            No          A 0          UNKNOWN          D No
112                No          A 56952      OK (279 ms)      192.168.1.222    Yes
servidor2/servidor2 Yes          5060        UNREACHABLE
4 sip peers [Monitored: 2 online, 2 offline Unmonitored: 0 online, 0 offline]
```

### 5. Compobaciones.

En este apartado lo he querido hacer para poner las compobaciones de las llamadas que se han realizado con éxito:

Llamadas locales:



Llamadas fuera de la conexión WI-FI:



### **6. Problemas de realización no resueltos.**

En este proyecto a la hora de realización me he encontrado problemas como que un día va todos los procedimiento realizados y al día siguiente no va nada. Por ejemplo las llamadas fuera de la conexión local, trunk entre las dos centralitas...

Problemas con las tarjetas SD.

He tenido problemas también que a la hora de hacerle el Trunk de pago no he podido encontrarlo para ponerle las referencias encontradas.

### **7. Conclusión y Valoración personal del Proyecto.**

El proyecto en sí me parece una muy buena idea, a la hora de acercarnos a la tecnología de las llamadas VIOPs y también a la hora de acercarnos a herramientas como Asterisk y FreePBX y periféricos como Raspberry Pi.

Gracias a esto hemos podido realizar la mayoría de los procedimientos propuestos, menos los fallos propuestos arriba, pero por lo demás sin problemas.

Gracias a la herramienta de FreePBX ha sido bastante intuitivo trabajar con Asterisk y buscando la información adecuada parece que es fácil trabajar con ello.

### **8. Bibliografía.**

Varios vídeos de Youtube entre otros –

<https://www.youtube.com/>

Wikipedia-

<https://www.wikipedia.org/>

Blogspot-

<http://elereyes.blogspot.com.es/2012/08/instalar-no-ip-en-elastix-22-linux.html>

Redeszone NO-IP-

<http://www.redeszone.net/raspberry-pi/no-ip/>

Wikiasterisk-

[http://www.wikiasterisk.com/index.php?title=P%C3%A1gina\\_principal](http://www.wikiasterisk.com/index.php?title=P%C3%A1gina_principal)

Dominio NOIP –

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

FreePBX –

<http://www.freepbx.org/>

Wiki freepbx –

<http://wiki.freepbx.org/display/F2/Trunk+Sample+Configurations>

Lalegionyoda Trunk SIP-

<https://lalegiondeyoda.wordpress.com/2011/04/20/trunk-sip-entre-2-asterisk/>

Trunk movistar-

<http://unpocodetodoqueesmucho.blogspot.com.es/2014/01/sip-movistar-ftth-parametros.html>



<http://bandaancha.eu/foros/configurar-freepbx-movistar-adsl-voip-1702058>

Códigos-

<http://code.portallinux.es/index.php?show=26>

<http://portallinux.es/raspberry-pi-configuracion-de-no-ip/>

## 9. Fecha Fin de Proyecto.

Melilla, Domingo 30 de Agosto de 2015

