



Proyecto Comunicaciones Con Raspberry con VOIP Con Asterisk y FreePBX

*Raspberry con Asterisk con un Trunk de
comunicación SIP*

Manuel Alfaya Tello

2-6-2015

INDICE

| | |
|---|---------------------|
| <u>INDICE.....</u> | <u>1</u> |
| <u>1.- ¿Qué es Aterisk?.....</u> | <u>2</u> |
| <u>2.- ¿Qué es FreePBX?.....</u> | <u>3</u> |
| <u>3.- ¿Qué es el VOIP?.....</u> | <u>4</u> |
| <u>4.- ¿Qué es telefonía IP?.....</u> | <u>5</u> |
| <u>5.- ¿Qué son los protocolos?.....</u> | <u>6</u> |
| <u>5.1.- Protocolo SIP (Session Initiation Protocol): (Es la que usaremos en este proyecto).....</u> | <u>6</u> |
| <u>5.2.- Protocolo IAX.....</u> | <u>6</u> |
| <u>5.3.- Protocolo H.323.....</u> | <u>7</u> |
| <u>5.4.- Protocolo MGCP.....</u> | <u>7</u> |
| <u>6.- ¿Qué es una Raspberry Pi?.....</u> | <u>8</u> |
| <u>7.- Objetivos del proyecto.....</u> | <u>9</u> |
| <u>8.- Desarrollo del Proyecto.....</u> | <u>10-18</u> |
| <u>9.- Conclusión y Valoración Personal.....</u> | <u>19</u> |
| <u>10.- Fecha de terminación del proyecto.....</u> | <u>20</u> |
| <u>11.- Bibliografía.....</u> | <u>21</u> |

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.



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.
- Follow-me.
- 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.- ¿Qué es el VOIP?

(Voz sobre protocolo de Internet), también llamado voz sobre IP, 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

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.

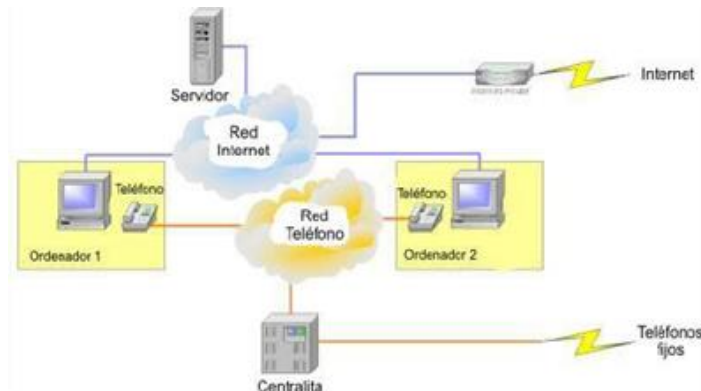


Diagrama de cómo era la telefonía PSTN

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.



Diagrama del empaquetamiento de la voz

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.

5.1.- 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. Fue desarrollado inicialmente en el grupo de trabajo IETF MMUSIC (Multiparty Multimedia Session Control) y, a partir de Septiembre de 1999, pasó al grupo de trabajo IETF SIP.

- Acrónimo de “Session Initiation Protocol”.
- 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.

5.2.- Protocolo IAX

- Acrónimo de “Inter Asterisk Exchange”.
- IAX es un protocolo abierto, es decir que se puede descargar y desarrollar libremente.
- Aún no es un estándar.
- Es un protocolo de transporte, que utiliza el puerto UDP 4569 tanto para señalización de canal como para RTP (Protocolo de Transporte en tiempo Real).
- Puede trunkar o empaquetar múltiples sesiones dentro de un flujo de datos, así requiere de menos ancho de banda y permite mayor número de canales entre terminales.

- En seguridad, permite la autenticación, pero no hay cifrado entre terminales.

5.3.- Protocolo H.323

- Originalmente fue diseñado para el transporte de vídeo conferencia.
- Su especificación es compleja.
- H.323 es un protocolo relativamente seguro, ya que utiliza RTP.
- Tiene dificultades con NAT, por ejemplo para recibir llamadas se necesita direccionar el puerto TCP 1720 al cliente, además de direccionar los puertos UDP para la media de RTP y los flujos de control de RTCP.
- Para más clientes detrás de un dispositivo NAT se necesita gatekeeper en modo proxy.

5.4.- Protocolo MGCP

- Acrónimo de “Media Gateway Control Protocol”.
- Inicialmente diseñado para simplificar en lo posible la comunicación con terminales como los teléfonos.
- MGCP utiliza un modelo centralizado (arquitectura cliente * servidor), de tal forma que un teléfono necesita conectarse a un controlador antes de conectarse con otro teléfono, así la comunicación no es directa.
- Tiene tres componentes un MGC (Media Gateway Controller), uno o varios MG (Media Gateway) y uno o varios SG (Signaling Gateway), el primero también denominado dispositivo maestro controla al segundo también denominado esclavo.
- No es un protocolo estándar.

6.- ¿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.

7.- Objetivos del proyecto.

1º Poder realizar llamadas locales.

2º Poder realizar llamadas desde el exterior.

3º Trunkar las dos centralitas una vez hecho el paso anterior.

4º Trunkar SIP a las centralitas.

5º Hacer si es posible el operador.

Realizar llamadas a precio de local entre países, con lo aplicado al proyecto, en las raspberrys estarán juntadas con un Trunk de SIP al cual tendremos que llamar y luego nos saldrá una operadora (esta parte sería opcional) indicándonos a que extensión queremos llamar de la otra centralita que tendremos enlazada a la 1º de estas con un Trunk para llamar a las otras extensiones.

8.- Desarrollo del Proyecto.

En este apartado explicaremos los pasos que hay que dar para hacer este proyecto, lo primero que tenemos que hacer es buscar la imagen para hacer la SD que nos trae la Raspberry “Booteable” para empezar con la instalación del Sistema y su debida configuración.

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
```

Una vez hecho, esto nos descargaremos la aplicación de Asterisk y nos descargaremos la aplicación de trabajar en ello con un entorno gráfico llamado FreePBX.

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. Y por si acaso haremos un **apt-get update** y **apt-get upgrade** por si se nos queda algo sin actualizar.

Tras hacer las actualizaciones, reiniciamos la máquina (este paso no es necesario) para que se efectúen los cambios.

Una vez actualizado nos tocará configurar la zona horaria en la que estamos con el comando siguiente:

```
configure-timezone
```

Luego nos tocara configurar las configuraciones locales que es con el siguiente comando:

```
dpkg-reconfigure locales
```

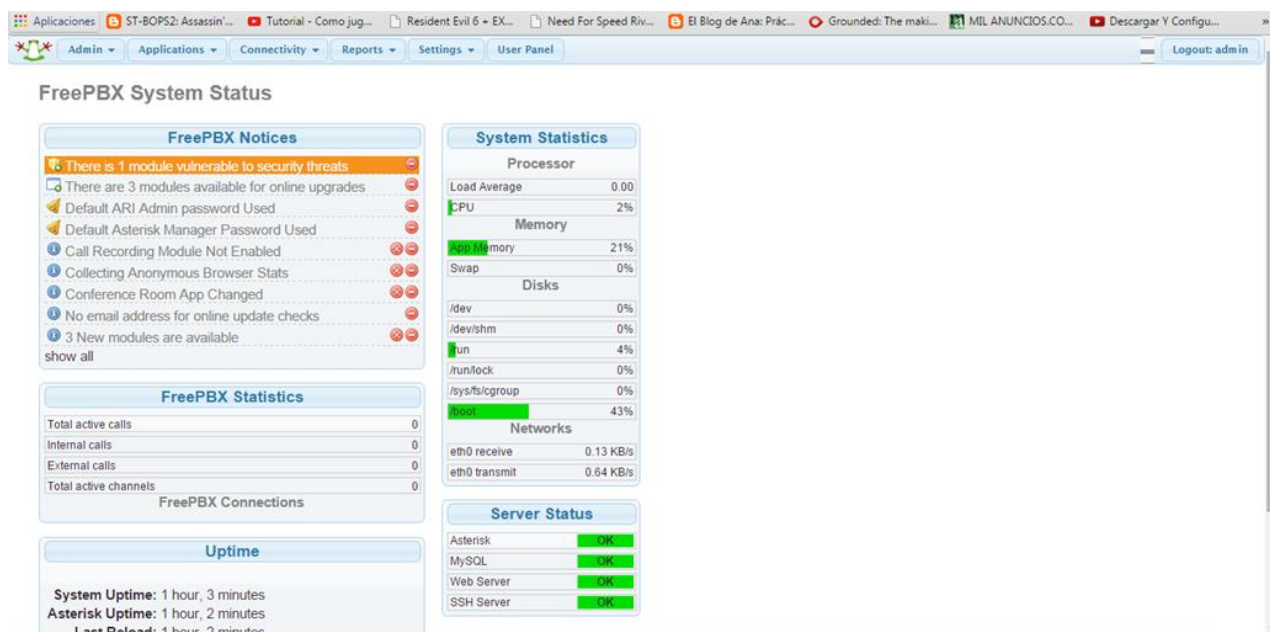
Y el último comando sirve para configurar nuestro teclado que no está en el lenguaje correcto:

```
dpkg-reconfigure keyboard-configuration
```

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).



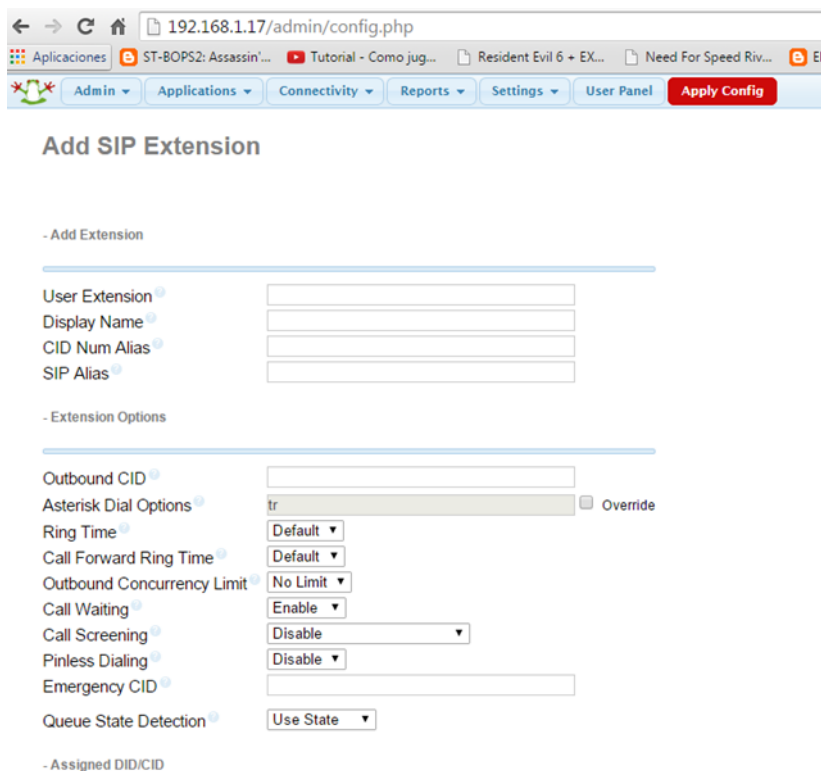
Tras acceder al **FreePBX** desde el navegador lo siguiente que tenemos que acceder como administrador en la que nos mostrará el estado de las aplicaciones y de nuestra centralita:



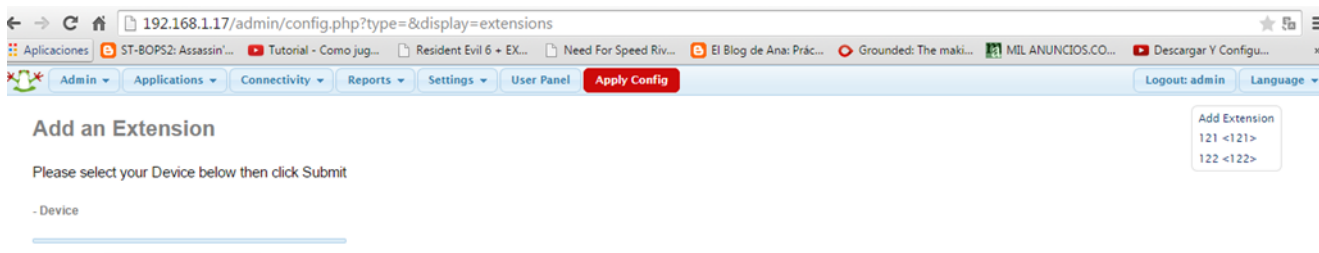
Una vez accedemos como administrador, procederemos a hacer las **llamadas locales** dentro de una misma de red entre dos móviles conectados en la misma red WIFI a partir de la aplicación Zoiper, lo primero que tendremos que hacer para proceder a las llamadas es crear las extensiones SIP como vemos en las siguientes imágenes:



Una vez elegido el tipo de extensión, procedemos a crearla dándole un número que queramos en mi caso por ejemplo le pondré: 121. Y una de las partes más importantes en este paso es ponerle una contraseña para que nadie se nos pueda meter en la llamada (en el apartado de secret). Como vemos en la siguiente imagen:



Una vez creadas nuestras extensiones le damos a “*Apply Config*” y vemos como las tenemos creadas:

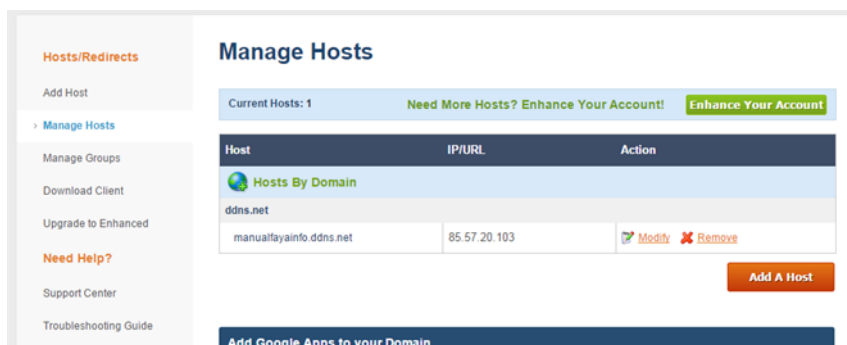


Tras hacer esto nos vamos a nuestros dos dispositivos móviles que estén conectados en la misma red que nuestra centralita configurándolo con las extensiones creadas y poniendo el nombre de host con ip:puerto de **Asterisk** que en mi caso he utilizado la aplicación **Zoiper** para realizar la llamada local con 192.168.1.17:5060 como vemos en las siguientes imágenes y vemos que da la llamada:



Al hacer este paso, lo siguiente es hacer **llamadas desde el exterior**, entonces uno de los dispositivos móviles estará en la red wifi y el otro a partir de los datos podrá llamar desde el exterior de la red, para hacer esto, hay que seguir estos pasos:

- Lo primero que tendremos que hacer es acceder a nuestro router de la red para abrir los puertos del tipo UDP para poder salir al exterior, al 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.
- Tras abrir los puertos, nos tendremos que abrir un **Dominio NOIP** 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 -xzf 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 'xxxxxxx@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.

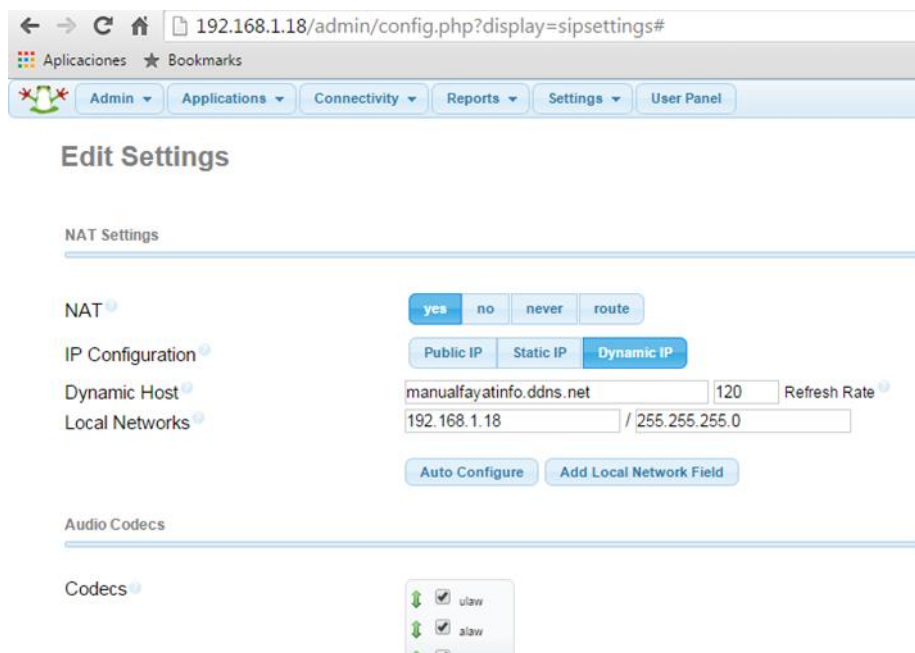
mv /tmp/no-ip2.conf /usr/local/etc/no-ip2.conf

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

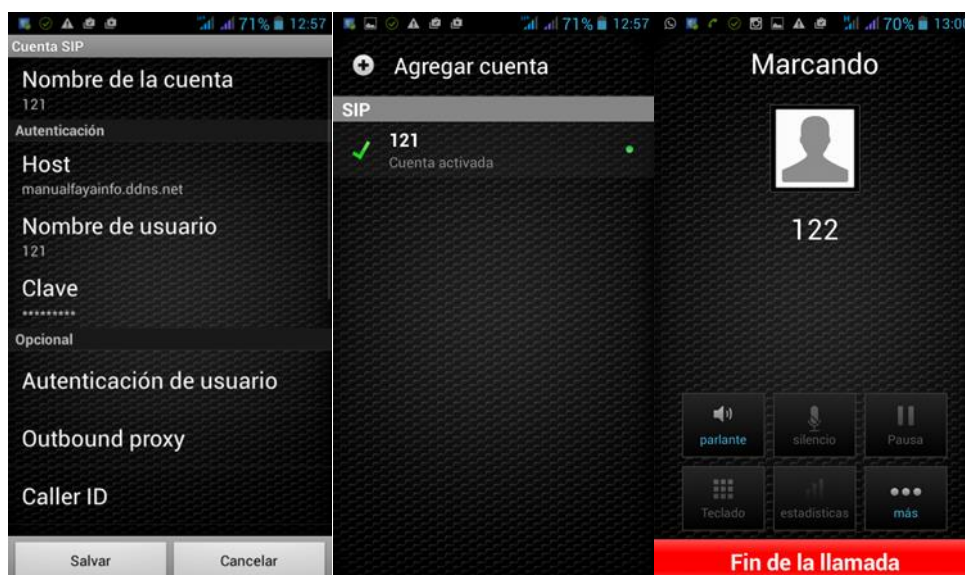
Y con esto ya tenemos configurado nuestro **no-ip** en asterisk para que se autoactualice.

Para correrlo en el arranque solo agregamos la línea **/usr/local/bin/noip2** en archivo **/etc/rc.local**.

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:



Una vez hecho esto reiniciaremos asterisk para que se produzcan los cambios con un amportal restart esperamos a que reinicie y con nuestra aplicación cogemos una de las extensiones y a una la dejamos conectada a la red wifi y al otro dispositivo le pondremos en la zona de host nuestro dominio de NOIP como vemos en la siguiente imágenes y veos que nos da la llamada desde el exterior (como explicamos anteriormente):



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.

Cambien 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

192.168.1.17/admin/config.php?display=trunks8

Admin Applications Connectivity Reports Settings

+ Add More Dial Pattern Fields Clear all Fields

Dial Rules Wizards: (pick one)

Outbound Dial Prefix:

Outgoing Settings

Trunk Name: Servidor2

PEER Details:

```
host=192.168.1.18
username=servidor2
qualify=yes
context=from-internal
trunk=yes
disallow=all
allow=gsm
```

Incoming Settings

USER Context: Servidor1

USER Details:

```
host=192.168.1.18
type=user
context=from-internal
disallow=all
allow=gsm
```

Centralita 2

192.168.1.18/admin/config.php?display=trunks&tech=SIP

Admin Applications Connectivity Reports Settings User Panel

Asterisk Trunk Dial Options

Continue if Busy: ☐ Override

Disable Trunk: ☐ Check to always try next trunk ☐ Disable

Dialed Number Manipulation Rules

(prepend) + prefix match pattern

+ Add More Dial Pattern Fields Clear all Fields

Dial Rules Wizards: (pick one)

Outbound Dial Prefix:

Outgoing Settings

Trunk Name: Servidor1

PEER Details:

```
host=192.168.1.17
username=servidor1
type=peer
qualify=yes
context=from-internal
trunk=yes
disallow=all
allow=gsm
```

Incoming Settings

USER Context: Servidor2

USER Details:

```
host=192.168.1.17
type=user
context=from-internal
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:

Centralita 1

192.168.1.17/admin/config.php?display=routing&extdisplay=1#

Admin Applications Connectivity Reports Settings User Panel

Delete Route Servidor2

Route Settings

Route Name: Servidor2

Route CID: ☐ Override Extension

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

Centralita 2

192.168.1.18/admin/config.php?display=routing&extdisplay=1

Admin Applications Connectivity Reports Settings User Panel

Edit Route

Delete Route Servidor1

Route Settings

Route Name: Servidor1

Route CID: ☐ Override Extension

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

Normal Congestion

Y como vemos en el panel de control nos dice que el Trunk Online está en buen estado, pero me falla al querer llamar desde una extensión de una centralita a otra extensión de otra centralita.

192.168.1.18/admin/config.php?display=index

Admin Applications Connectivity Reports Settings User Panel

FreePBX System Status

FreePBX Notices

- There is 1 module vulnerable to security threats
- There are 3 modules available for online upgrades
- Default ARI Admin password Used
- Default Asterisk Manager Password Used
- Call Recording Module Not Enabled
- Collecting Anonymous Browser Stats
- Conference Room App Changed
- No email address for online update checks
- 3 New modules are available

show all

FreePBX Statistics

| | |
|-----------------------|---|
| Total active calls | 0 |
| Internal calls | 0 |
| External calls | 0 |
| Total active channels | 0 |

FreePBX Connections

| | |
|------------------|---|
| IP Phones Online | 2 |
| IP Trunks Online | 1 |

System Statistics

Processor

Load Average: 0

CPU

Memory

App Memory: 2

Swap

Disks

/dev

/dev/shm

/run

/run/lock

/sys/fs/cgroup

/boot: 4

Networks

eth0 receive: 0.21 K

eth0 transmit: 0.70 K

Server Status

| | |
|----------|----|
| Asterisk | OK |
| MySQL | OK |

Y por último este paso no lo he podido realizar pero me he informado en el que una vez hecho el trunkamiento entre las dos centralitas cogeremos dos números VOIP en las que le haremos un trunk SIP con nuestro número virtual VOIP en las que le crearemos su debido Outbound Route y Inbound Route como hicimos en los Trunkeamientos de las dos centralitas. (Esto fue según la información que he estado viendo).

9.- Conclusión y Valoración Personal.

Este proyecto me ha parecido muy interesante, por la interacción de las Raspberrys que han sido un gran acierto y gracias a las aplicaciones de asterisk y FreePBX que han facilitado mucho este proyecto a la hora de tener un interfaz gráfico, esto para mí ha sido la primera vez con el tema de las llamadas VOIP y me ha sorprendido todo la gran cantidad de cosas que se podrían hacer. Pero a mí me ha ocurrido diversos problemas con este proyecto, en el que se desconfiguraba cada dos por tres, mi router no dejaba salir las llamadas externas estando los puertos abierto incluso, me tuve que mover de casa en casa para hacer esta parte, el trunkeamiento entre las dos centralitas parece que se conectan pero no deja llamar entre si y el trunkeamiento del número virtual VOIP no he podido hacerlo pero más o menos he querido ponerlo el cómo podría ser este último proceso para que veáis que he intentado hacerlo de todas las formas posibles y bueno ... ha salido más o menos aun que me hubiera gustado terminarlo, pero me ha gustado el proyecto.

10.- Fecha de terminación del proyecto.

Este proyecto se ha terminado el día 2 de Junio de 2015.

11.- 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
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/>

CONTRAPORTADA