

Índice

Módulo I: Introducción.....	3
1.Repaso: Telefonía IP.....	3
a.Comparativa entre la telefonía tradicional y la telefonía IP.....	3
2. Protocolo SIP.....	4
3. Asterisk.....	4
a.Configuración básica.....	4
b.Comandos básicos.....	5
c.Repaso: Redes de comunicación para sistemas de VoIP.....	5
Módulo II: Realtime.....	6
1.Características y funcionalidades Realtime.....	6
a.Ventajas y desventajas.....	6
2.Fundamentos de Bases de datos MySQL.....	6
a.Interfaz gráfica phpMyAdmin.....	6
b.Creación de usuarios y permisos.....	7
3.Configuración de Realtime.....	7
a.Conector a bases de datos ODBC.....	7
b.Configuración Asterisk para Realtime.....	7
Taller II.....	8
1.Despliegue de máquina virtual con Debian.....	8
2.Nociones básicas de manejo en Linux/Debian.....	9
3.Instalación MySQL y phpMyAdmin.....	10
4.Configuración de Realtime.....	13
a.Configuración en Asterisk.....	13
b.Conexión con Base de datos.....	13
c. Adición de extensiones Realtime.....	16
Módulo III: Colas, Meetmes: Dialplan avanzado utilizando Realtime.....	18
1.Colas.....	18
a.Sintaxis en el dialplan.....	18
b.Colas inteligentes y sus estrategias.....	18
c.Agentes estáticos y dinámicos.....	19
d.Locuciones en colas.....	19
2.Meetme.....	19
a.Salas de conferencia dinámicas.....	19
Taller III.....	20
1.Configuración de Colas Inteligentes.....	20
1.Adición de una cola por GUI.....	20
2.Adición de una cola dinámicas.....	20
3.Adición de agentes.....	20
4.Colocación de locuciones.....	20
2.Configuración de MeetMe.....	20
Módulo IV: AGI.....	21
1.Conceptos clave.....	21
2.Métodos AGI: AGI, FastAGI, EAGI, AsyncAGI.....	21
3.Variable de entrada: agi_*.....	21



ELE-0217CL VoIP Nivel II
 Julio 2017 - Palencia

Actividades formativas profesorado de especialidades vinculadas a la F.P.

4.Entornos de programación: BASH, php, perl, Python, C.....	22
Módulo V: AMI/AJAM.....	23
1.Conceptos clave.....	23
2.Configuración de Asterisk Manager Interface.....	24
3.Sintaxis de AMI: Action, Response, Events.....	24
4.Manager Commands: manager show commands.....	25
5.Ejemplo Asynchronous Javascript Asterisk Manager.....	25
6.Configuración Click2Call.....	26
Taller V.....	26
1.Configuración AMI.....	26
2.Creación y configuración de botón Click2Call.....	27
Módulos VI: CDR + CEL (accounting).....	28
1.Definición y configuración de CDR.....	28
2.Sintaxis en el Dialplan.....	28
3.CDR en Realtime (Adaptive CDR).....	29
4.CEL, complemento de CDR.....	29
Taller VI.....	29
1.Configuración de CDR en Realtime.....	29
2.Creación de pequeño registro de llamadas.....	30

Módulo I: Introducción

1. Repaso: Telefonía IP

La telefonía IP, o como se conoce en inglés VoIP (Voz sobre protocolo de Internet) es un método por el cual, las señales de audio analógicas se transforman en datos digitales que pueden ser transmitidos a través de internet hacia una dirección IP determinada.

Aunque la telefonía IP existe desde hace muchos años, debido a la inestabilidad de las redes de Internet, y su pobre calidad de servicio, no ha sido hasta ahora cuando su implementación ha empezado a ser real en el mundo empresarial y doméstico.

Actualmente, empresas como, entre otras, Telefónica, han descartado continuar usando la telefonía tradicional para dar paso a la telefonía IP.

a. Comparativa entre la telefonía tradicional y la telefonía IP

	TELEFONÍA TRADICIONAL	TELEFONÍA IP
CALIDAD SONIDO	Baja	Desde baja hasta alta definición
MÉTODO DE TRANSMISIÓN	Líneas telefónicas de cobre convencionales	A través de cualquier red IP. (LAN, Wi-Fi, 4G, FTTH, DSL, Satélite)
TERMINALES	Teléfonos analógicos, Fax	Teléfonos analógicos, Teléfonos digitales, Móviles, Ordenadores
MOVILIDAD	No	Si
NODOS REQUERIDOS	Muchos	Pocos
COSTE DEL MANTENIMIENTO DE LA RED	Alto	Bajo
COSTE DE IMPLEMENTACIÓN EN EMPRESAS	Alto	Escalable
CANALES POR LÍNEA	Hasta 30 canales por línea (PRI)	Escalable
SOPORTE VÍDEOLLAMADAS	No	Sí

2. Protocolo SIP

También conocido por sus siglas en inglés como Protocolo de Inicio de Sesión. Es un protocolo de señalización utilizado ampliamente para establecer y terminar sesiones de comunicación multimedia. Guarda cierta similitud con protocolos como el HTTP (navegación web) y el SMTP (envío de correo electrónico). Es un protocolo estándar definido en la RFC 3261, y ha sido desarrollado por la comunidad de software libre.

El protocolo sirve para crear, modificar y finalizar sesiones entre dos o más partes que constan de uno o varios flujos de transmisión multimedia. Es ampliable y se adapta a distintas arquitecturas y entornos de implementación.

Sus funciones básicas son:

1. Traducción de un nombre de usuario a su dirección de red actual
2. Gestión de llamadas (adición, transferencia o eliminación de participantes)
3. Negociación entre los distintos participantes para ofrecer compatibilidad
4. Posibilidad de realizar cambios de compatibilidad durante una llamada

3. Asterisk

a. Configuración básica

Disposición de los archivos Asterisk (directorios de instalación por defecto en Linux Debian):

Ruta	Descripción
/etc/asterisk	Archivos de configuración
/usr/sbin	Ejecutables binarios
/var/log/asterisk	Mensajes de error, logs y CDR
/var/lib/asterisk (compilado)	Documentación, Música en espera, Sonidos
/usr/share/asterisk (repositorio)	Documentación, Música en espera, Sonidos

b. Comandos básicos

Iniciaremos asterisk ejecutando:

```
# /etc/init.d/asterisk start
```

Podremos acceder a la consola a través de:

```
# asterisk -rvvvvvvvvvv
```

Reiniciar el servicio de Asterisk para recargar la configuración:

```
# service asterisk restart
```

Ver los códecs instalados en la centralita Asterisk desde la consola:

```
aula-pbx-00*CLI> core show codecs
```

Recargar la configuración para cargar un nuevo códec:

```
aula-pbx-00*CLI> module load codec_g729.so
```

Editar la configuración SIP

```
# nano /etc/asterisk/sip.conf
```

Recargamos la configuración:

```
aula-pbx-00*CLI> sip reload
```

c. Repaso: Redes de comunicación para sistemas de VoIP

La capacidad de polimorfismo de la telefonía IP permite un gran número de topologías y arquitecturas distintas, así como la posibilidad de proveer distintos servicios sobre ellas. La tecnología de Voz IP es capaz de ser implementada en entornos como:

Redes LAN

La implementación de telefonía IP sobre redes privadas están orientadas hacia servicios de centralita telefónicos para una empresa, un hospital, o un hotel. Las comunicaciones entre todos los terminales de la empresa se realizan a través de una centralita IP.

Redes WAN

Implementaciones comúnmente destinadas para la comunicación entre varias sedes en distintos puntos geográficos a través de una red virtual privada, sin la necesidad de contratar a un operador.

También pueden implementarse en este tipo de redes sistemas de Voz IP para comunicar una empresa con la red telefónica convencional a través de un operador IP.

Redes MAN

La implementación de telefonía IP sobre este tipo de redes es la más extendida en la actualidad por todo tipo de operadores. Sirve para transportar Voz a grandes distancias con unos costes bajos.

Módulo II: Realtime

1. Características y funcionalidades Realtime

Asterisk Realtime (o Asterisk en tiempo real) es una tecnología que permite la interconexión de Asterisk con distintos tipos de bases de datos y su utilización para el almacenamiento de la configuración de Asterisk. De esta forma, cualquier cambio de configuración realizado en la base de datos será replicado automáticamente a la centralita sin necesidad de reiniciarla.

Su principal uso está destinado a la integración de paneles de control visuales con los que controlar la centralita, o la interconexión con otros sistemas, como el Click 2 Call.

a. Ventajas y desventajas

La principal ventaja de la utilización de esta tecnología es la capacidad de manejo de la centralita desde otras aplicaciones sin necesidad de reinicios y cortes de servicio.

Como desventaja, la tecnología realtime requiere de conocimientos avanzados y su configuración inicial puede ser compleja.

2. Fundamentos de Bases de datos MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle y está considerada como la base datos open source más popular del mundo. ^{Wikipedia}

a. Interfaz gráfica phpMyAdmin

Las bases de datos relacionales suelen utilizar su propio lenguaje basado en el estándar SQL. MySQL cuenta con una línea de comandos a través de la cual, las bases de datos pueden ser creadas, manejadas y administradas.

Sin embargo, existen interfaces gráficas que ayudan a los usuarios a utilizar el sistema de forma más eficiente.

phpMyAdmin es la interfaz gráfica más utilizada para manejar MySQL. Una de sus principales ventajas es que funciona desde cualquier explorador de forma remota.

b. Creación de usuarios y permisos

Como veremos más adelante, MySQL permite ser gestionada por distintos usuarios, cada uno de ellos configurados con distintos permisos.

Esto aumenta el nivel de seguridad y permite que ciertos usuarios tengan acceso a una o varias bases de datos. Además, con el sistema de permisos se puede restringir el acceso y permitir la lectura, actualización, inclusión de datos, o su borrado.

3. Configuración de Realtime

a. Conector a bases de datos ODBC

Para que Asterisk se relacione con una base de datos, y así pueda almacenar y leer su configuración, es necesario contar con un conector.

Un conector básicamente es un controlador que se encarga de la comunicación entre una base de datos y otro programa.

En el caso de Asterisk, existen varios conectores disponibles para relacionarse con bases de datos. A continuación, los tres más utilizados habitualmente:

-Conector ODBC: se trata de un conector genérico que utiliza un lenguaje estandarizado y es posible conectar a la mayoría de bases de datos relacionales (Oracle, MySQL, SQL Server, Postgres, ...)

-Conector MySQL: Asterisk, además incluye soporte nativo para conectar con bases de datos MySQL. La interconexión es directa y no necesita de una traducción intermedia a un lenguaje estándar.

-Conector Postgres: es el soporte nativo de bases de datos Postgres.

b. Configuración Asterisk para Realtime

La configuración de Asterisk dependerá de que motor de base de datos utilicemos, y que conector deseemos emplear.

En versiones anteriores de Asterisk solamente era posible emplear el conector ODBC, puesto que los nativos de MySQL y Postgres no estaban todavía desarrollados.

Hoy en día, no tiene sentido utilizar el conector estándar ODBC si vamos a utilizar uno de estos dos motores de base de datos.

En cualquier caso, los archivos de configuración de Realtime se separan en dos bloques: la configuración de la conexión con la base de datos y las relaciones entre tablas y módulos de asterisk.

Configuración conectores	Configuración tablas
res_config_myql.conf res_config_sqlite.conf res_pgsqllite.conf res_odbc.conf	extconfig.conf

Taller II

1. Despliegue de máquina virtual con Debian

Accede a la memoria USB proporcionada, abre la carpeta *Software* y haz doble clic sobre el instalador de Virtual Box. También puedes descargar la última versión del programa desde <https://www.virtualbox.org/wiki/Downloads>

Haz doble clic sobre el fichero Oracle_VM_VirtualBox_Extension_Pack-X.X.XX-XXXXXX.vbox-extpack e instala este addon.

Copia la carpeta llamada "Asterisk VM" al escritorio. Una vez completado, accede y haz doble clic sobre el fichero con la extensión .vbox

Se abrirá Virtual Box con la máquina virtual ya configurada. Ahora encenderemos la máquina haciendo doble clic desde el menú lateral, y esperaremos a que arranque.

```
Usuario: user  
Contraseña: 123  
  
Usuario: root  
Contraseña: 123
```

La primera tarea a reiniciar será cambiar la dirección IP de la máquina editando el fichero */etc/network/interfaces* según las instrucciones proporcionadas. Una vez hecho, será necesario reiniciar la máquina con el comando *reboot* para aplicar los cambios.

2. Nociones básicas de manejo en Linux/Debian

Comando	Ejemplo	Función
ls	ls /usr/src	Mostrar los archivos y directorios de un directorio
cd	cd /usr/src	Acceder a un directorio
cp	cp archivo1.txt archivo1.copia.txt	Copiar un fichero
cp -R	cp -R /videos /copiavideos	Copiar un directorio de forma recursiva
mv	mv archivo1.txt /tmp/archivo1.txt	Mover un archivo a otro directorio
mv	mv archivo1.txt archivo2.txt	Cambiar de nombre un archivo o directorio
mkdir	mkdir videos	Crear un directorio
nano	nano archivo1.txt	Editar un fichero
pwd	pwd	Mostrar la ruta actual en consola
halt	halt	apagar el sistema
reboot	reboot	reiniciar el sistema
apt-get apt	apt-get install asterisk apt install asterisk	apt-get install asterisk

3. Instalación MySQL y phpMyAdmin

Previamente será necesario tener instalado un servidor web como Apache con las librerías PHP correctamente configuradas. En este caso, la máquina virtual proporcionada ya viene con el software preinstalado. En otro caso, se debería instalar de la siguiente forma:

```
# apt install apache2 php5 libapache2-mod-php5
```

La instalación de MySQL es bastante sencilla. Simplemente hay que ejecutar el siguiente comando:

```
# apt install mysql-server mysql-client
```

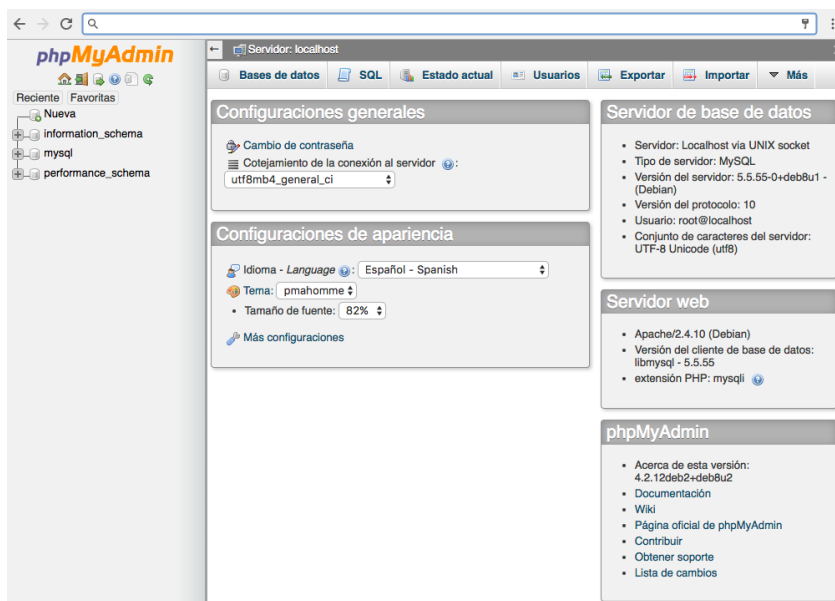
Durante el proceso de instalación será requerido asignar una contraseña maestra. Por conveniencia, utilizaremos 123.

La instalación de phpMyAdmin en sistemas Debian se limita a ejecutar el siguiente comando, y seguir el asistente de configuración:

```
# apt install phpmyadmin
```

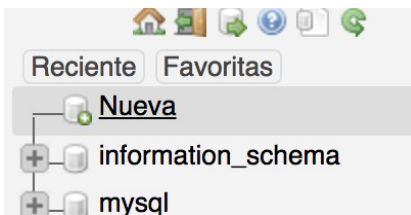
Durante el proceso de instalación será requerido introducir la contraseña asignada previamente durante la instalación de MySQL.

Por último, accedemos mediante cualquier explorador web a la siguiente dirección `http://{DIRECCIÓN IP}/phpmyadmin` con el usuario `root`, y la contraseña `123`



Nuestra primera tarea será crear una base de datos donde almacenaremos todas las tablas que recogerán la configuración de asterisk.

Para ello accederemos a nuestro panel de phpmyadmin, y en la parte izquierda haremos clic en **Nueva** para crear nuestra base de datos.



La llamaremos **asterisk**.



Ahora crearemos un usuario con permiso limitados que se encargue de leer y escribir únicamente en la base de datos de asterisk.

Para ello pulsaremos en la pestaña Usuarios de la parte superior del panel.



Después, Agregar usuario



Rellenaremos los datos requeridos de la siguiente forma

Agregar usuario

Información de la cuenta

Nombre de usuario: Use el campo de texto

Servidor: Local

Contraseña: Use el campo de texto

Debe volver a escribir:

Generar contraseña:

(Contraseña: 123)

Tras pulsar **Continuar** y crear el usuario, pulsaremos sobre el enlace de **Editar los privilegios** del mismo usuario. Después, pulsaremos en la barra superior en la pestaña Base de datos y clicaremos en el campo inferior “Añadir privilegios a esta base de datos” sobre **asterisk**.

Editar los privilegios: Usuario 'cursovoip'@'local'

Privilegios específicos para la base de datos

Base de datos	Privilegios	Conceder	Privilegios específicos para la tabla	Acción
			Ninguna	

Añadir privilegios a esta base de datos:

Una nueva ventana aparecerá con todos los privilegios disponibles y asignables al usuario sobre la base de datos. Marcaremos todos los privilegios, y después guardaremos con Continuar.

Privilegios específicos para la base de datos Marcar todos

Nota: Los nombres de los privilegios de MySQL están expresados en inglés.

Datos	Estructura	Administración
<input checked="" type="checkbox"/> SELECT	<input checked="" type="checkbox"/> CREATE	<input checked="" type="checkbox"/> GRANT
<input checked="" type="checkbox"/> INSERT	<input checked="" type="checkbox"/> ALTER	<input checked="" type="checkbox"/> LOCK TABLES
<input checked="" type="checkbox"/> UPDATE	<input checked="" type="checkbox"/> INDEX	<input checked="" type="checkbox"/> REFERENCES
<input checked="" type="checkbox"/> DELETE	<input checked="" type="checkbox"/> DROP	
	<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES	
	<input checked="" type="checkbox"/> SHOW VIEW	
	<input checked="" type="checkbox"/> CREATE ROUTINE	
	<input checked="" type="checkbox"/> ALTER ROUTINE	
	<input checked="" type="checkbox"/> EXECUTE	
	<input checked="" type="checkbox"/> CREATE VIEW	
	<input checked="" type="checkbox"/> EVENT	
	<input checked="" type="checkbox"/> TRIGGER	

Continuar

4. Configuración de Realtime

a. Configuración en Asterisk

Como primera tarea, modificaremos y descomentaremos (quitar ;) en `/etc/asterisk/asterisk.conf` los siguientes parámetros:

```
verbose = 3
debug = 3

defaultlanguage = es
```

b. Conexión con Base de datos

Configurar la conexión de realtime en asterisk requiere de la edición del fichero **res_config_mysql.conf**, que como cualquier otro archivo de configuración de este software, se encuentra en `/etc/asterisk/`

En este fichero definimos la comunicación entre Asterisk y MySQL.

Al final del documento, dentro del contexto *[general]*, editaremos el documento con los siguientes parámetros.

```
[asterisk]
dbhost = 127.0.0.1
dbname = asterisk
dbuser = cursovoip
dbpass = 123
dbport = 3306
dbsock = /var/run/mysqld/mysqld.sock
dbcharset = latin1
requirements=warn
```

De esta forma, indicamos a Asterisk que debe conectarse a una base de datos MySQL que se encuentra en la misma máquina (*127.0.0.1*), a la base de datos llamada *asterisk*, y debe acreditarse con el usuario *cursovoip* y la contraseña *123*.

Ahora, reiniciaremos el módulo correspondiente de Asterisk para recargar la configuración

```
# rasterisk
*CLI> module reload TODO
*CLI>
```

Alternativamente, en caso de querer utilizar el conector ODBC, sería necesario habilitar el módulo ODBC en el fichero `modules.conf`

```
[modules]
preload => res_odbc.so
preload => res_config_odbc.so
```

Después, instalar lo siguiente:

```
# apt install unixODBC unixODBC-dev libmyodbc
```

Configurar el fichero `/etc/odbcinst.conf` para indicar que conectaremos el conector con `mysql`

```
[MySQL]
Description = ODBC for MySQL
```

```
Driver          = /usr/lib/odbc/libmyodbc.so
Setup           = /usr/lib/odbc/libodbcmyS.so
FileUsage       = 1
```

Tras reiniciar, si todo funciona correctamente, al ejecutar el comando `odbcinst -q -d` deberíamos obtener como resultado `[MySQL]`

Ahora, crearemos un fichero donde definiremos los datos de conexión con la base de datos MySQL en `/etc/odbc.ini`

```
[asterisk-connector]
Description       = MySQL connection to
'asterisk' database
Driver           = MySQL
Database         = asterisk
Server           = localhost
Port             = 3306
Socket           = /var/run/mysqld/mysqld.sock
```

Por último, indicaremos a Asterisk dónde debe encontrar los datos de conexión a la base de datos en `/etc/asterisk/res_odbc.conf`

```
[asterisk]
enabled => yes
dsn => asterisk-connector
username => asterisk
password => welcome
pooling => no
limit => 1
pre-connect => yes
```

Ahora, reiniciamos el módulo ODBC de asterisk y comprobamos que funcione correctamente.

```
# rasterisk

*CLI> module reload res_odbc.so
*CLI> odbc show
ODBC DSN Settings
-----
Name:    asterisk
DSN:    asterisk-connector
Last connection attempt: 1969-12-31 19:00:00
Pooled: No
Connected: Yes
```


c. Adición de extensiones Realtime

Una vez configurado el conector, debemos definir qué tablas utilizará Asterisk para las extensiones Realtime dentro de la base de datos definida. Esta configuración se encuentra en el archivo **extconfig.conf**

Teniendo en cuenta que la etiqueta `[settings]` ya está descomentada, añadiremos las líneas necesarias para que el documento quede de esta forma:

```
[settings]

sippeers => mysql,asterisk,sip_buddies
queues => mysql,asterisk,queues
queue_members => mysql,asterisk,queue_members
meetme => mysql,asterisk,meetme
extensions => mysql,asterisk,extensions
```

Ahora, deberemos añadir las tablas a la base de datos donde luego asterisk consultará en tiempo real.

Para ello volveremos al panel de control de phpMyAdmin, clicaremos en la parte izquierda sobre la base de datos **asterisk**, y clicaremos en la opción **Importar** situado en el menú superior.

Utilizando el botón de **Seleccionar archivo...** cargaremos los siguientes scripts .sql localizados junto a este documento.

Una vez hecho esto, ya tenemos todas las tablas necesarias para trabajar durante este curso.

Como el cometido del curso no es la programación, sino la integración de Asterisk con aplicaciones externas ya programadas, procederemos a instalar un mini panel de administración de asterisk con fines docentes.

```
$ su
# apt install git
# cd /usr/src/
# git clone https://github.com/PlaneaSoluciones/MiniRealtime-Asterisk
# mv MiniRealtime-Asterisk/panel /var/www/html/panel
# cd /var/www/html/panel
```

Revisaremos que la configuración del panel de control con la base de datos sea correcta con **nano config.php**

```
<?php  
  
define('DBUSER','cursovoip');  
define('DBPWD','123');  
define('DBHOST','localhost');  
define('DBNAME','asterisk');
```

Por último, para comprobar que el panel de control está correctamente instalado, accederemos desde el navegador a: <http://{DIRECCION IP}/panel>

Si todo está configurado correctamente, podremos añadir nuevas extensiones desde el propio panel de control, y configurarlas en los teléfonos y los softphones.

Portada	Extensiones SIP	Colas	Salas de conferencia	Listado de llamadas	Click 2 Call
Listado	Anadir				
Id	Nº extensión	Nombre	Secret		
1	101	Manuel	101	Editar Eliminar	
2	102	Pedro	102	Editar Eliminar	
3	110	Perico	1234	Editar Eliminar	

Módulo III: Colas, Meetmes: Dialplan avanzado utilizando Realtime

1. Colas

a. Sintaxis en el dialplan

La sintaxis para las colas en dialplan es la siguiente:

```
Queue(queueename,[options,[URL,[announceoverride,[timeout,[AGI,[macro,[gosub,[rule,[position]]]]]]]]))
```

queueaname: extensión de la cola

options: pueden ser consultadas en
<https://wiki.asterisk.org/wiki/display/AST/Asterisk+11+Application+Queue>

timeout: tiempo de expiración de la cola

Para que una cola funcione en realtime, es posible definir el dialplan en base de datos. En concreto, la tabla extensions cumple esta función. Como ejemplo, se podría implementar del siguiente modo:

id	context	exten	priority	app	appdata
12	internas	_9XX	1	Answer	
13	internas	_9XX	2	Queue	\${EXTEN},t

De este modo, las llamadas a las colas de 3 dígitos que comiencen por 9 entrarían en el sistema de colas.

b. Colas inteligentes y sus estrategias

Las colas inteligentes, a diferencia de las colas tradicionales permiten seguir distintas estrategias de recepción de llamadas.

Estas son las estrategias soportadas por asterisk:

- ringall: suenan todas las extensiones de los agentes configurados al mismo tiempo.
- roundrobin: suena por turnos entre los agentes
- leastrecent: suena por orden según los agentes que lleven más tiempo sin responder una llamada
- fewestcalls: suena por orden según los agentes que han atendido menos llamadas
- random: suena de forma aleatoria
- rrmemory: igual que roundrobin, pero recordando en qué posición quedó la última ronda de llamadas

c. Agentes estáticos y dinámicos

Los agentes estáticos son aquellas extensiones predefinidas para una cola que siempre estarán disponibles para recepcionar llamadas.

Los agentes dinámicos son aquellas extensiones que pueden entrar o salir de una cola de llamadas bajo demanda.

d. Locuciones en colas

Asterisk permite la configuración de distintas locuciones que se reproducen mientras los llamantes escucha música en espera.

`announce_holdtime`: anuncia el tiempo de espera previsto y la posición en la cola

`periodic_announce`: anuncia el mensaje que se defina cada X tiempo, donde X es el tiempo definido en `periodic_announce_frequency`

Los mensajes personalizables son:

`queue_youarenext` "Eres el siguiente"

`queue_thereare` "hay"

`queue_callswaiting` "Llamadas esperando"

`queue_holdtime` "tiempo de espera"

`queue_minutes` "minutos"

`queue_seconds` "segundos"

`queue_lessthan` "menos de"

`queue_thankyou` "gracias"

`queue_reporthold` "Quedan"

Y son reproducidos según `announce_frequency`

También es posible que el agente escuche una locución al responder una llamada configurando el parámetro `announce`

2. Meetme

Originariamente, el módulo de Asterisk encargado de gestionar salas de conferencia (`call conference`) se llamaba Meetme. En la actualidad, es módulo ha quedado abandonado a favor de la utilización de `Confbridge`. De cualquier modo, de forma coloquial es común seguir refiriéndose a este módulo como `meetme`.

a. Salas de conferencia dinámicas

Las salas de conferencia dinámicas son aquellas llamadas con más de 2 participantes que pueden ser creadas bajo demanda en realtime.

Conbridge además cuenta con la posibilidad de la utilización de un menú en el que se puede regular el volumen de la llamada desde el teclado numérico.

Sin embargo, como principal diferencia a Meetme, Conbridge no cuenta de forma nativa con opciones como la autenticación, control de asistentes máximos, o menú de administración. En cualquier caso, todo esto puede ser configurado mediante dialplan o a través de su archivo de configuración `conbridge.conf`

Taller III

1. Configuración de Colas Inteligentes

1. Adición de una cola por GUI

Creación de una cola desde el panel de control.

2. Adición de una cola dinámicas

Creación de una cola dinámica desde phpMyAdmin

3. Adición de agentes

Creación de agentes desde phpMyAdmin

4. Colocación de locuciones

Configuración de locuciones desde phpMyAdmin

2. Configuración de MeetMe

Configuración de una sala de conferencia

Módulo IV: AGI

1. Conceptos clave

Asterisk Gateway Interface (Interfaz - Puerta de enlace – Asterisk) o AGI provee de una interfaz estándar que permiten el control del dialplan de Asterisk. Suele utilizarse para aplicar lógica avanzada a Asterisk desde otros recursos.

Los canales de comunicación utilizados con AGI son STDIN, STDOUT y STDERR.

El canal STDIN se encarga de recoger información de Asterisk

El canal STDOUT se encarga de enviar la información de Asterisk

El canal STDERR se encarga de enviar información de DEBUG y errores.

2. Métodos AGI: AGI, FastAGI, EAGI, AsyncAGI

La aplicación EAGI (Enhanced AGI) funciona como AGI, pero además permite la lectura de la entrada de sonido.

La aplicación DeadAGI funciona como AGI, pero se utiliza sobre llamadas ya colgadas.

La aplicación FastAGI permite utilizar scripts AGI a través de la red. De esta forma múltiples servidores asterisk pueden utilizar un mismo script localizado en otro servidor.

AsyncAGI permite utilizar aplicaciones AMI de forma asíncrona para y ejecutar scripts de forma asíncrona (bajo demanda).

3. Variables de entrada: agi_*

agi_request – El nombre de tu script

agi_channel – El canal que origina una llamada

agi_language – Código de idioma (“es”)

agi_type – Tipo de canal (“SIP”)

agi_uniqueid – Un código identificador único de llamada

agi_version – Versión de asterisk

agi_callerid – Número Identificador de llamada

agi_calleridname – Nombre del identificador de llamada

agi_dnid – Identificador del número marcado

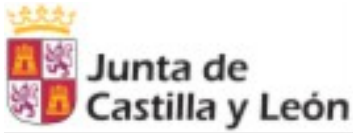
agi_context – Contexto de asterisk

agi_extension – Extensión marcada

agi_priority – Prioridad en dialplan

agi_threadid – Identificador de la ejecución del script

Otras variables pueden ser consultadas en <https://www.voip-info.org/wiki-Asterisk+AGI>



ELE-0217CL VoIP Nivel II
Julio 2017 - Palencia

Actividades formativas profesorado de especialidades vinculadas a la F.P.

4. Entornos de programación: BASH, php, perl, Python, C...

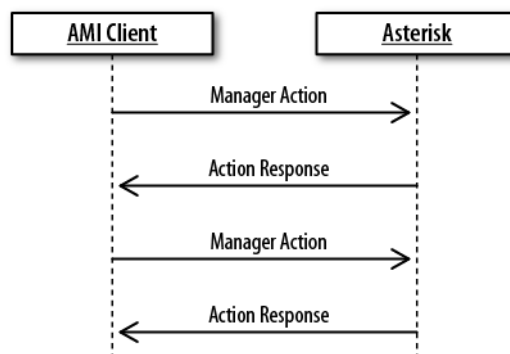
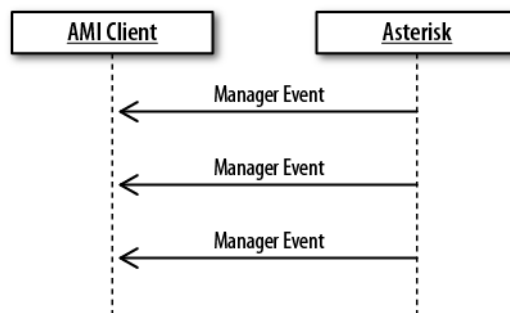
La comunicación entre un programa y AGI se realiza a través de llamadas a scripts. Es por ello que cualquier entorno de programación puede comunicarse con asterisk siempre que las librerías estén instaladas en el servidor.

Módulo V: AMI/AJAM

1. Conceptos clave

Asterisk Manager Interface (AMI) es una interfaz de Asterisk que permite a programas externos a controlar y monitorizar el sistema Asterisk. Esta interfaz es habitualmente utilizada para integrar Asterisk con aplicaciones de procesos de negocios como CRMs.

AMI está a la escucha de conexiones por un puerto de red. Un programa puede conectarse a este puerto, autenticarse y enviar comandos a Asterisk. La centralita responderá a las peticiones.



2. Configuración de Asterisk Manager Interface

La configuración de AMI se define en el fichero /etc/asterisk/manager.conf

Ejemplo:

```
[general]
enabled = yes
;webenabled = yes

port = 5038
bindaddr = 0.0.0.0

[admin]
secret = 123
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.0
read =
system,call,log,verbose,command,agent,user,originate
write =
system,call,log,verbose,command,agent,user,originate
```

3. Sintaxis de AMI: Action, Response, Events

Events: información recibida sobre eventos Asterisk.

```
Event: Hangup
Privilege: call,all
Channel: SIP/0004F2060EB4-00000000
Uniqueid: 1283174108.0
CallerIDNum: 2565551212
CallerIDName: Russell Bryant
Cause: 16
Cause-txt: Normal Clearing
```

Actions: información enviada para realizar alguna acción en la centralita.

Responses: información recibida sobre una acción enviada.

```
Action: Login
Username: russell
Secret: russell

Response: Success
```

Message: Authentication
accepted

4. Manager Commands: manager show commands

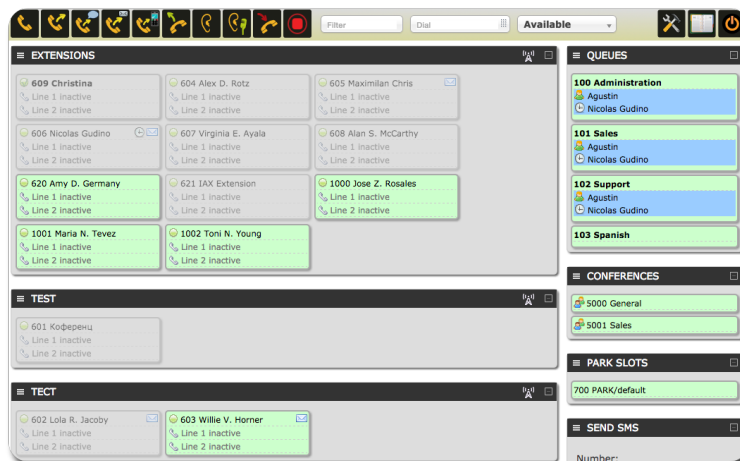
Todas las acciones disponibles con AMI pueden consultarse con el comando *manager show commands*

5. Ejemplo Asynchronous Javascript Asterisk Manager

La comunicación de AMI se realiza a través de Telnet, es por ello que cualquier entorno de programación con una librería de conexión de este tipo puede interactuar con Asterisk.

AJAM (Asynchronous Javascript Asterisk Manager) es una librería Javascript capaz de comunicarse con Asterisk desde cualquier navegador de hoy en día.

Esto permite que aplicaciones como, por ejemplo, Flash Operator Panel, reciban información en directo de una centralita y muestren el estado de las llamadas entrantes, colas, etc.



6. Configuración Click2Call

Seguramente habrás observado que algunos sitios web (generalmente tiendas en línea) cuentan con unos botones en los que introduces tu número de teléfono, e inmediatamente recibes una llamada de un agente.

Este software, generalmente, está realizado a través de una centralita Asterisk y su interfaz AMI.

Básicamente, se envía una acción de llamada a una extensión telefónica (o una cola), e inmediatamente después, se lanza otra acción de llamada al número introducido en el sitio web.

Taller V

1. Configuración AMI

Para comprender su funcionamiento, podemos conectar a través del comando telnet a nuestra centralita:

```
# telnet 127.0.0.1 5038
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
Asterisk Call Manager/2.10.0
Action: login
Username: admin
Secret: 123
```

2. Creación y configuración de botón Click2Call

A continuación, un ejemplo programado en PHP totalmente funcional.

```
$oSocket = fsockopen(127.0.0.1, 5038, $errnum, $errdesc) or die("Connection  
to host failed");  
fputs($oSocket, "Action: login\r\n");  
fputs($oSocket, "Events: off\r\n");  
fputs($oSocket, "Username: admin\r\n");  
fputs($oSocket, "Secret: 123\r\n\r\n");  
fputs($oSocket, "Action: originate\r\n");  
fputs($oSocket, "Channel: $strChannel\r\n");  
fputs($oSocket, "WaitTime: $strWaitTime\r\n");  
fputs($oSocket, "CallerId: Click2Call\r\n");  
fputs($oSocket, "Exten: $ext2\r\n");  
fputs($oSocket, "Context: $strContext\r\n");  
fputs($oSocket, "Priority: 1\r\n\r\n");  
fputs($oSocket, "Action: Logoff\r\n\r\n");  
sleep(3);  
fclose($oSocket);
```

Módulos VI: CDR + CEL (accounting)

1. Definición y configuración de CDR

CDR (Call Data Record) es el registro de llamadas de Asterisk.

Este módulo se encarga de registrar todos los eventos de llamada realizados desde una centralita, y es capaz de almacenarlos en ficheros o en base de datos.

2. Sintaxis en el Dialplan

Más allá de la configuración inicial, CDR no requiere de código en el dialplan, exceptuando casos de personalización.

Su sintaxis para estos casos es la siguiente:

CDR(<name>[options])

Donde *name* puede ser:

accountcode: Código de canal

amaflags: Etiquetas personalizables (FACTURABLE, GRATUITA, BONO)

answer: Fecha en la que fue respondida

billsec: Tiempo facturable

channel: Nombre del canal

clid: Nombre identificador de llamada

dcontext: Contexto destino

disposition: ANSWERED, NO ANSWER, BUSY

dst: Destino de la llamada

dstchannel: Canal de destino (SIP, internal, IAX2)

duration: Duración total de la llamada

end: Fecha de finalización de la llamada

src: Origen

start: Fecha de origen de la llamada

uniqueid: Identificador único

userfield: Campo personalizable

Puedes encontrar la lista completa aquí <https://www.voip-info.org/wiki/view/Asterisk+func+cdr>

3. CDR en Realtime (Adaptive CDR)

Como se explicaba anteriormente, Asterisk puede almacenar estos registros en base de datos a tiempo real.

Actualmente, al igual que con el Realtime de extensiones, colas y conferencias, también hay varios conectores con bases de datos nativos incluyendo ODBC y mysql.

4. CEL, complemento de CDR

Channel Event Logging provee del mecanismo para almacenar todos los eventos generados en una llamada, de tal forma que es posible consultar paso a paso que procesos a seguido Asterisk y así poder analizar posibles fallos o problemas, cuantas extensiones no han contestado a una misma llamada, etc.

Taller VI

1. Configuración de CDR en Realtime

Como primer paso deberemos activar el registro de CDR en Asterisk sobre el fichero **cdr.conf** en `/etc/asterisk/`

```
[general]
enable=yes
unanswered = yes
```

En este curso utilizaremos el conector mysql. CDR se configura de forma independiente al resto de módulos Realtime.

Esto debe hacerse sobre el fichero **cdr_mysql.conf** en

```
[global]
hostname=localhost
dbname=asterisk
password=123
user=root
```




ELE-0217CL VoIP Nivel II
Julio 2017 - Palencia
Actividades formativas profesorado de especialidades vinculadas a la F.P.

```
table=cdr
:port=3306
:sock=/var/run/mysqld/mysqld.sock
:userfield=1
```

Por último, será necesario reiniciar Asterisk (*core restart now*) para recargar la configuración.

2. Creación de pequeño registro de llamadas

Puedes encontrar un ejemplo de registro de llamadas en el MiniRealtime Asterisk Panel.