

TICAR 2013

Foro de Responsables de TICS de las Universidades Nacionales

TALLER VOIP: Asterisk, como lo hacemos.



Diego G. Pérez (diegoperez@unnoba.edu.ar)



UNIVERSIDAD
NACIONAL DE
VILLA MARIA

Benjamín Garzón (bgarzon@unvm.edu.ar)

Agenda



- Asterisk → Arquitectura: diseño, canales, interfaces.
- Protocolo SIP: definición, sip.conf, “outside connectivity”.
- Dialplan (I): estructura, contextos, extensiones, prioridades.
- Dialplan (II): variables y expresiones, patrones, macros, aplicaciones útiles.
- Versionado de Asterisk.



Asterisk → Arquitectura

- **Diseño Modular**
- Servicios modificables en Real Time.
- Los módulos son cargados dinámicamente.
- `/etc/asterisk/modules.conf`

```
[modules]  
autoload=yes
```

```
noload=chan_ooh323.so  
noload=cdr_radius.so
```

- Desde la CLI se pueden cargar o descargar módulos.

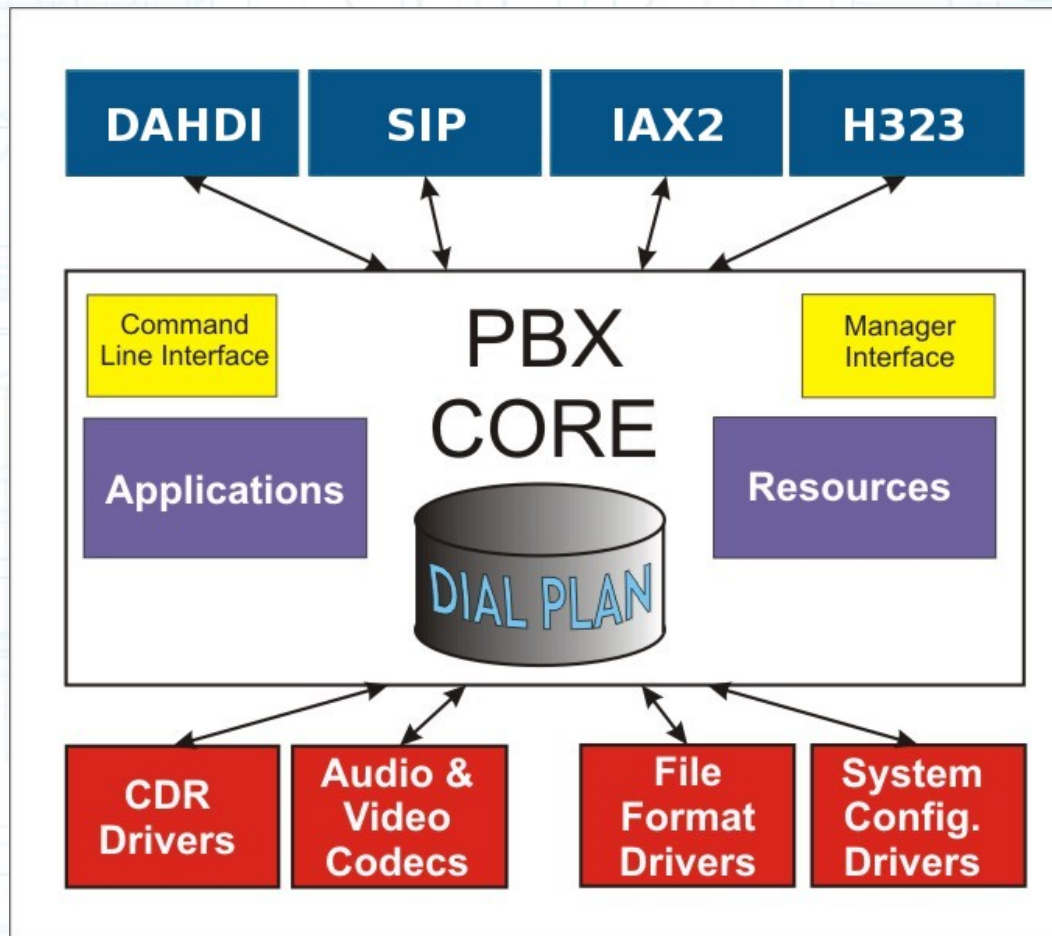
```
voip*CLI> module load chan_ooh323.so
```



Asterisk → Arquitectura

- **Channel drivers**
- Administran todo proceso de entrada y salida.
- Cada canal representa diferentes tecnologías o protocolos.
- SIP
- IAX
- H323
- DAHDI
- Local
- Motif (Asterisk 11)
- Etc.

Asterisk → Arquitectura





Asterisk → Arquitectura

- **Aplicaciones y Funciones**
- Son cargadas dinámicamente cuando se requieren, mediante módulos que las contienen.
- Se pueden ejecutar varias aplicaciones y funciones en el curso de una llamada.

Ej:

exten => 555,1,Answer()

exten => 555,2,Playback(demo-thanks)

exten => 555,3,Hangup()



Asterisk → Interfaces

- **Interfaces de Asterisk**
- CLI: The Comand Line Interface.
- AMI: The Manager Interface.
- AGI: The Asterisk Gateway Interface.



Asterisk → Interfaces

- **CLI: La interface de linea de comandos**
- Consola de administración de Asterisk.
- Es gestionado por el núcleo de Asterisk, pero cada módulo puede agregar comandos.



Asterisk → Interfaces

- **CLI: La interface de linea de comandos**
- Consola de administración de Asterisk.
- Es gestionado por el núcleo de Asterisk, pero cada módulo puede agregar comandos.

```
Archivo Editar Ver Buscar Terminal Ayuda
voipdesa:/etc/asterisk# asterisk -r
Asterisk 11.3.0, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 11.3.0 currently running on voipdesa (pid = 7288)
voipdesa*CLI> █
```



Asterisk → Interfaces

- **AMI: Asterisk Manager Interface**
- Permite administrar Asterisk desde una aplicación externa.
- Nos permite enviar comandos y recibir eventos.
- Por cada acción que Asterisk realiza, se genera un evento que puede ser capturado por una sesión de Manager.
- Todo en Tiempo Real.



Asterisk → Interfaces

- `etc/asterisk/manager.conf`



Asterisk → Interfaces

- `/etc/asterisk/manager.conf`

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
voipdesa:/etc/asterisk# cat manager.conf
[general]
enabled = yes
webenabled = no
port = 5038
bindaddr = 127.0.0.1

[test]
secret=test
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.0
read=system,call,config,command
write=system,call,config,command
voipdesa:/etc/asterisk# _
```



Asterisk → Interfaces

```
voipdesa*CLI> manager show settings

Global Settings:
-----
Manager (AMI):           Yes
Web Manager (AMI/HTTP): No
TCP Bindaddress:        127.0.0.1:5038
HTTP Timeout (minutes): 60
TLS Enable:             No
TLS Bindaddress:        Disabled
TLS Certfile:           asterisk.pem
TLS Privatekey:
TLS Cipher:
Allow multiple login:   Yes
Display connects:      Yes
Timestamp events:      No
Channel vars:
Debug:                 No
Block sockets:         No
voipdesa*CLI> _
```



Asterisk → Interfaces

- Servicio AMI

```
Archivo Editar Ver Buscar Terminal Ayuda
voipdesa:/etc/asterisk# netstat -nltup | grep asterisk
tcp        0      0 127.0.0.1:5038      0.0.0.0:*           LISTEN     7288/asterisk
udp        0      0 0.0.0.0:5060       0.0.0.0:*           7288/asterisk
udp        0      0 0.0.0.0:4569       0.0.0.0:*           7288/asterisk
voipdesa:/etc/asterisk# _
```



Asterisk → Interfaces

- **AGI: The Asterisk Gateway Interface**
- Permite programar y ejecutar aplicaciones hechas en otros lenguajes desde el DIALPLAN de Asterisk.
- Es usado comúnmente para programar rutinas complejas.
- Básicamente consiste en un script invocado desde el DIALPLAN al cual le podemos pasar parámetros.
- EJ:

exten => 500,1,AGI(hola-mundo.sh)



Asterisk → Arquitectura

- **Archivos de configuración**
- Estructura:
 - [general] → Sección
 - secret=pass → Clave = Valor
- Principales archivos de configuración:
 - asterisk.conf
 - sip.conf
 - extensions.conf
 - iax.conf



Asterisk → Protocolo SIP

- **Session Initiation Protocol**
- Señaliza, se encarga de iniciar, modificar y finalizar sesiones multimedia.
- Modelo cliente – servidor.
- Se complementa con SDP y RTP.
- Definido en RFC 3261.
- En Asterisk: puerto 5060.



Asterisk → Protocolo SIP

- **/etc/asterisk/sip.conf**

```
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6                               Fichero: sip.conf
[general]
allowguest=yes
context=guest
udpbindaddr=0.0.0.0
port=5060
transport=udp
language=es
alwaysauthreject=yes

[codecs-audio](!)
disallow=all
allow=g722
allow=alaw

[codecs-video](!)
videosupport=yes
allow=h264

[phone](!,codecs-audio)
type=friend
host=dynamic
qualify=yes

[100](phone)

[101](phone,codecs-video)
```



Asterisk → Protocolo SIP

- **Registro y Autenticación**

*Los telefonos se **registran** para decirle a Asterisk en que IP se encuentran, para que les pueda enviar las llamadas ahí.*

*La **autenticación** de llamadas salientes es algo totalmente diferente a lo anterior, y sucede por cada llamada que el dispositivo intenta realizar.*



Asterisk → PSTN

- PSTN CIRCUITS
 - Telefonía Analógica(FXO & FXS)
 - Telefonía Digital(T1 , E1)
- VOIP (trunk)
 - Asterisk – Gateway (Trunk SIP)
 - Analógicos, PRI, BRI, GSM.
 - Asterisk – Asterisk (Trunk SIP o IAX2)
 - Diferentes sedes.
 - Asterisk – Asterisk (Trunk SIP (99%))
 - Salida a la PSTN.



Asterisk → Dialplan(I)

- Es el corazón de Asterisk, concentra toda su lógica.
- Define los comportamientos que tomara la plataforma al momento de:
 - Recibir un llamado.
 - Decidir por donde será enviada una llamada.
 - Aplicar servicios sobre las comunicaciones.
- Es un “mini” lenguaje de scripting.



Asterisk → Dialplan(I)

- **Estructura**
- `/etc/asterisk/extensions.conf`
- Esta compuesto por las siguientes unidades lógicas:
 - Contextos
 - Extensiones
 - Prioridades
 - Aplicaciones



Asterisk → Dialplan(I)

- **Contextos**
- El dialplan se divide en diferentes secciones denominadas contextos.
- Son grupos de funcionalidades a los que se les da un nombre.
- Sintaxis:
 - [nombre-contexto]
 - Todo lo que se define a continuación pertenece al contexto.



Asterisk → Dialplan(I)

- **Contextos especiales**
 - [general]
 - Son definiciones que se aplican a todos los contextos.
 - [globals]
 - Se definen variables globales que podrán accederse desde el dialplan.
 - Ej: ITSP=SIP/proveedorX



Asterisk → Dialplan(I)

- **Inclusiones**
- Los contextos pueden incluirse entre si, mediante la sentencia “include”.



Asterisk → Dialplan(I)

- **Inclusiones**
- Los contextos pueden incluirse entre si, mediante la sentencia “include”.

```
[Acelular]
exten => _15X.,1,Dial(${ITSP}/${EXTEN},40)

[internos]
exten => _1XX,1,Dial(SIP/${EXTEN},40)

[usuarioAvanzado]
include => internos
include => Acelular

[usuarioBasico]
include => internos_
```



Asterisk → Dialplan(I)

- **Extensiones**
- En Asterisk, una extensión no tiene que asociarse a un número.
- Pueden ser números o no.
- Representa un conjunto de instrucciones:
- Compuestas por:
 - Nombre.
 - Prioridad.
 - Aplicación.



Asterisk → Dialplan(I)

- **Sintaxis:**
- Se define mediante la palabra “exten” seguida por una flecha (=>).
- exten => nombre, prioridad, aplicación
- Ej:

exten => 500, 1, Answer()



Asterisk → Dialplan(I)

- **Priodidad**
- Cada extensión contiene múltiples pasos que se ejecutan secuencialmente según su prioridad.
- Ej:
 - exten => 500, **1**, “tarea 1”
 - exten => 500, **2**, “tarea 2”
 - exten => 500, **3**, “tarea 3”



Asterisk → Dialplan(I)

- Problema:
 - Que sucede si tenemos una extensión muy compleja y modificamos la primer prioridad?



Asterisk → Dialplan(I)

- Problema:
 - Que sucede si tenemos una extensión muy compleja y modificamos la primer prioridad?
- Solución:
 - No enumerar las prioridades, que se enumeren solas.



Asterisk → Dialplan(I)

- Problema:
 - Que sucede si tenemos una extensión muy compleja y modificamos la primer prioridad?
- Solución:
 - No enumerar las prioridades, que se enumeren solas.
- Prioridad “**n**”
 - Permite enumerar solo la primer prioridad, asignando automaticamente el resto.

exten => 500, **1**, “tarea 1”

exten => 500, **n**, “tarea 2”

exten => 500, **n**, “tarea 3”



Asterisk → Dialplan(I)

- Nuevo Problema:
 - Que sucede si queremos hacer un salto de una prioridad a otra no consecutiva?
 - Deberíamos saber de antemano el número de prioridad.
 - Las podemos contar :(.



Asterisk → Dialplan(I)

- Nuevo Problema:
 - Que sucede si queremos hacer un salto de una prioridad a otra no consecutiva?
 - Deberíamos saber de antemano el número de prioridad.
 - Las podemos contar :(.
- Solución:
 - **Labels (etiquetas)**
 - exten => 123,**1**, IF a=b then **igual** else **distinto**
 - exten => 123,**n(igual)**, hace una cosa
 - exten => 123,**n(distinto)**, hace otra



Asterisk → Dialplan(I)

A partir de la versión 1.6.2 hay una manera de evitar tener que repetir el numero/nombre de la extensión o patrón usando: **same => prefix**



Asterisk → Dialplan(I)

A partir de la versión 1.6.2 hay una manera de evitar tener que repetir el numero/nombre de la extensión o patrón usando: **same => prefix**

exten => 123, **1**, IF a=b then **igual** else **distinto**

same => **n(igual)**, hace una cosa

same => **n(distinto)**, hace otra cosa



Asterisk → Dialplan(I)

- **Aplicaciones**
- Son las unidades funcionales del dialplan.
- Cada prioridad ejecuta una aplicación.
- Pueden recibir argumentos.
- Ej: Answer(), Playback(), Dial()
- Y otras 178 más!!!
- ***CLI>** core show applications
- ***CLI>** core show application <nombre>



Asterisk → Dialplan(II)

- **Variables**
- Simplifican nuestro dialplan.
- Agregan lógica al sistema.
- Contenedores de información.
- Existen dos tipos de variables:
 - De canal
 - Globales



Asterisk → Dialplan(II)

- **De canal:**
 - Son variables asociadas al canal en curso.
 - Viven el tiempo que dura una llamada.
 - Asignarles un valor:
 - Ej:
 - exten => 123,1,Set(COUNT=3)
 - same => n,SayNumber(\${COUNT})



Asterisk → Dialplan(II)

- **Globales:**
- Están disponibles desde cualquier parte del dialplan.
- Se usan para definir constantes.
- Contexto exclusivo [globals].
- Ej: [globals]
 DIEGO=SIP/100

 [internos]
 exten => 100,1,Dial(\${DIEGO},40)



Asterisk → Dialplan(II)

- **Expresiones**
- Son una combinación de variables, operaciones y valores, que producen un resultado.
- Sintaxis → `$(expresion)`



Asterisk → Dialplan(II)

- **Expresiones**
- Son una combinación de variables, operaciones y valores, que producen un resultado.
- Sintaxis → $\$[expresion]$
- Ej:

exten => *50,1,SayNumber($\$[2+5]$)

- Asterisk primero reemplaza las variables(si las hay) por sus valores, evalúa la expresión, y luego ejecuta la aplicación con el valor obtenido.



Asterisk → Dialplan(II)

- **Patrones**
- Permiten definir expresiones regulares, contra la cual se comparará una extensión ingresada.
- Define rango de extensiones.
- Asterisk ordena todas las extensiones dentro de cada contexto de la mas especifica a la mas general.
- Siempre comienzan con el guion bajo.

• Ej:

- X → 0-9

- Z → 1-9

- N → 2-9

Extensiones del 100 al 199

exten => _1XX,1,Noop(\${EXTEN})



Asterisk → Dialplan(II)

- **Macros**
- Bloques de códigos que realizan un trabajo.
- Permiten la re-utilización de código.
- Reciben parámetros.
- Se define como un contexto, pero el nombre debe comenzar con “ macro- ”.
- Ej: [macro-llamada]
- Se utiliza invocando la función Macro()
- Ej: exten => 200,1,Macro(llamada)



Asterisk → Dialplan(II)

- **Una macro por dentro**
- Se utiliza la extensión “s”.
- Es posible pasar argumentos.
- Se acceden mediante $\${ARG1}$, $\${ARG2}$, etc.
- Ej:

exten => 100,1,Macro(llamada, $\${EXTEN}$, 30)

$\${ARG1}$ = $\${EXTEN}$

$\${ARG2}$ = 30



Asterisk → Dialplan(II)

- **Ejemplo sencillo!**

```
[macro-llamar]
```

```
exten => s,1,Dial(${ARG2}/${ARG1},40)
```

```
same => n,Goto(${DIALSTATUS})
```

```
same => n(noanswer),VoiceMail(${ARG1}@default,u)
```

```
same => n,Hangup()
```

```
same => n(busy),VoiceMail(${ARG1}@default,b)
```

```
same => n,Hangup()
```

```
[internos]
```

```
exten => _1[0-4]X,1,Macro(llamar,${EXTEN},SIP)
```

```
exten => _1[5-9]X,1,Macro(llamar,${EXTEN},IAX2)
```



Asterisk → Aplicaciones

- **Algunas aplicaciones**
- **DIAL**
 - Tiene como tarea principal, comunicar dos endpoints.
 - Comunica usuarios que utilizan tecnologías diferentes.
 - Administra el llamado, controlando su estado.
 - Es una de las aplicaciones más complejas (gran funcionalidad)
 - Recibe 4 parámetros:
 - Dispositivos
 - Timeout
 - Opciones
 - URL



Asterisk → Aplicaciones

- **Algunos ejemplos**

- Dial(SIP/100,30)
- Dial(SIP/200,40,rtT)
- Dial(IAX2/300)
- Dial(SIP/100&SIP/200&IAX2/300,40,rtT)
- Dial(SIP/proveedor/154999999,30)
- Dial(DAHDI/1/0115555555,30)



Asterisk → Aplicaciones

- **GotolfTime()**

GoTolfTime(times,days_of_week,days_of_month,month?label)

Ej: exten => s,1,GoTolfTime(09:00-17:59,mon-fri,*,*?open:close)



Asterisk → Aplicaciones

- **GotolfTime()**

GoTolfTime(times,days_of_week,days_of_month,month?label)

Ej: exten => s,1,GoTolfTime(09:00-17:59,mon-fri,*,*?open:close)

- **Wait() y WaitExten()**

Ej: exten => s,Wait(1.5)

exten => s,WaitExten(3)



Asterisk → Aplicaciones

- **GoTolftime()**

GoTolftime(times,days_of_week,days_of_month,month?label)

Ej: exten => s,1,GoTolftime(09:00-17:59,mon-fri,*,*?open:close)

- **Wait() y WaitExten()**

Ej: exten => s,Wait(1.5)

exten => s,WaitExten(3)

- **MixMonitor()**

Ej: exten => s,MixMonitor(archivo.ext,opciones)

same => n,Dial(\${alumnos},30)



Asterisk → Versionado

Release Series	Release Type	Release Date	Security Fix Only	EOL
1.2.X		2005-11-21	2007-08-07	2010-11-21
1.4.X	LTS	2006-12-23	2011-04-21	2012-04-21
1.6.0.X	Standard	2008-10-01	2010-05-01	2010-10-01
1.6.1.X	Standard	2009-04-27	2010-05-01	2011-04-27
1.6.2.X	Standard	2009-12-18	2011-04-21	2012-04-21
1.8.X	LTS	2010-10-21	2014-10-21	2015-10-21
10.X	Standard	2011-12-15	2012-12-15	2013-12-15
11.x	LTS	2012-10-25	2016-10-25	2017-10-25
12.x	Standard	2013-10 (tentative)	2014-10 (tentative)	2015-10 (tentative)
13.x	LTS	2014-10 (tentative)	2018-10 (tentative)	2019-10 (tentative)

Muchas Gracias

Diego G. Pérez (diegoperez@unnoba.edu.ar)

Benjamín Garzón (bgarzon@unvm.edu.ar)



VOIP-ARIU

Diego → 84611711

Benjamín → 8412345