

# CURSO ASTERISK

---



**Asterisk™**

*The Open Source PBX*

## Asterisk PBX

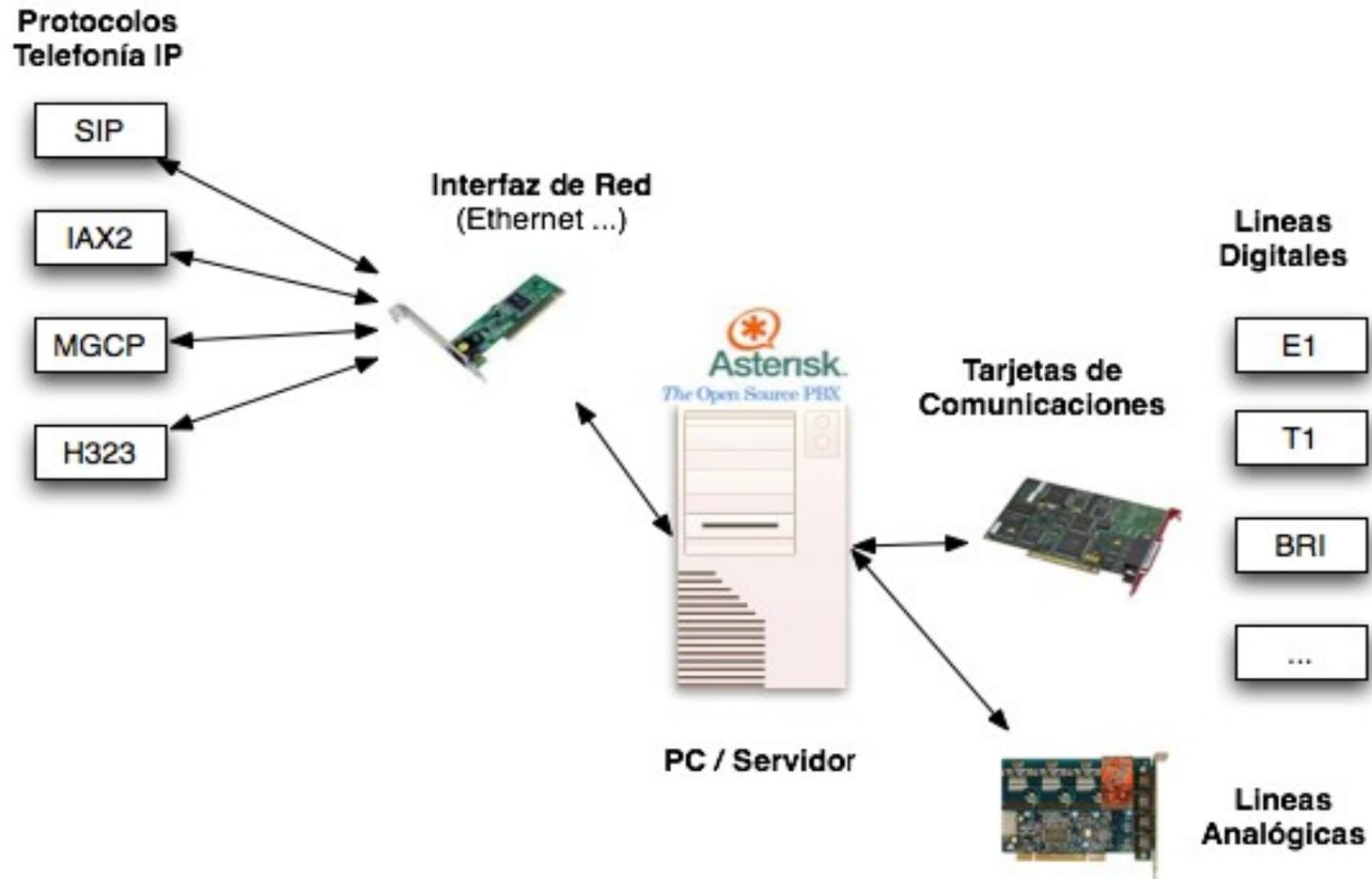
---

### ¿ Que es ?

- Asterisk es software. Exclusivamente software.
- Es software open source, desarrollado principalmente por la empresa americana DIGIUM.
- Liberado con la licencia GPL.
- Se ejecuta en PC estandar (arquitectura x86,x86\_64, ppc) bajo GNU/Linux, BSD o MacOSX.
- Soporta todas las funcionalidades de las centralitas tradicionales y muchas más !

## Asterisk PBX

### Esquema Conceptual



## Introducción

---



## Asterisk PBX: Instalación

---

### Requisitos Técnicos del sistema

- Requisitos: Dependen directamente de:
  - Llamadas concurrentes.
  - Conferencias y Aplicaciones complejas simultáneas.
  - Transcodificaciones necesarias (recodificación).
- Principalmente, Asterisk requiere microprocesador.
- Según Digium: Equipo Dual Intel Xeon 1.8 Ghz 1 Gb Ram soporta 60 llamadas concurrentes codificando con el codec G.729.
- Difícil determinar con exactitud, mejor apuntar alto para poder escalar.

## Preparando la instalación

---

- Requisitos Hardware
  - PC
  - Tarjetas de telefonía
  - Café
- Requisitos Software
  - Distribución GNU/Linux
  - <http://voip-info.org>

## Asterisk PBX: Instalación

---

### Instalación de librerías dependientes

- Asterisk necesita para su correcta compilación y funcionamiento los siguientes paquetes:
  - openssh-server: para acceso remoto seguro.
  - gcc,make: compilador de lenguaje C y herramientas
  - libc-dev,libssl-dev,zlib1g-dev,libncurses-dev,libmysqlclient-dev

## Asterisk PBX: Instalación

---

### Descarga del código fuente de Asterisk (I)

- **Asterisk:** Núcleo (core) del sistema.
- **Asterisk-sounds:** Voces de calidad pregrabadas.
- **Asterisk-addons:** Software adicional.
- **Libpri:** Librería para gestionar enlaces RDSI Primarios.
- **Zaptel:** Interfaz del Kernel para acceder a tarjetas de comunicaciones para líneas analógicas o digitales.
- **mISDN** Interfaz del kernel para acceso a tarjetas RDSI BRI
- **mISDNuser** Parte de misdn que se ejecuta en userspace

### Administración básica: arranque

- Asterisk es un demonio que se ejecuta en segundo plano. Se invoca con el comando 'asterisk':

```
sudo asterisk
```

- Una vez ejecutado, nos devuelve el control de la shell, haciendo un 'detach'. Podemos comprobar que se está ejecutando correctamente con un listado de procesos habitual:

```
ps aux | grep asterisk
```

### Administración básica: conexión al CLI

- En este punto tenemos el programa Asterisk en funcionamiento con la configuración de /etc/asterisk
- Asterisk soporta un intérprete de comandos (CLI: Command Line Interface), del estilo de muchos routers.
- Para conectarse basta con ejecutar el comando:

```
sudo asterisk -r
```

```
Asterisk 1.2.9.1, Copyright (C) 1999 - 2005 Digium.
```

```
Written by Mark Spencer <markster@digium.com>
```

```
Connected to Asterisk 1.2.1 currently running on ironiturion (pid = 28572)
```

### Administración básica: CLI

- El intérprete de comandos de Asterisk es bastante potente, y permite controlar y monitorizar gran parte de la situación de la centralita.
- Soporta el empleo de la tecla <Tabulador>, al estilo de las consolas de UNIX/GNU Linux, por lo que para ver un listado de todos los comandos disponibles, basta con presionar varias veces la tecla.
- Para ver los posibles argumentos de un comando o completar un parámetro largo o complicado.

### Administración básica: detención

- Es posible realizar una desconexión del CLI de Administración con 'quit'. Asterisk continuará ejecutándose en segundo plano.
- Para matar al propio Asterisk desde el CLI, se puede utilizar el comando stop, en sus tres variantes:
  - **stop now**: Detiene Asterisk al momento
  - **stop when convenient**: Detiene Asterisk cuando no haya carga.
  - **stop gracefully**: Detiene asterisk cuando no haya carga y deja de aceptar peticiones de llamadas a a partir de este momento.

### Administración básica: Verbose

- Nivel de “Verbose”: Este valor indica la cantidad de mensajes que se recibirán sobre los eventos generales del sistema. Cuanto más alto, más información sobre lo que sucede en la centralita se recibirá.
- Este nivel, se puede establecer de varias formas:
  - Al arrancar el demonio:  
`sudo asterisk vvvvvv`
  - Al conectarse al demonio:  
`sudo asterisk -rvvvvvvvv`
  - Desde el CLI:  
`CLI> Set Verbose 30`

### Administración básica: Debug

- Nivel de “*Debug*”: Este valor indica la cantidad de mensajes que se recibirán sobre los eventos generales del sistema, pero utilizado normalmente para depurar problemas de drivers o de aplicaciones.
- Este nivel, se puede establecer de varias formas:
  - Al arrancar el demonio:  
`sudo asterisk dddd`
  - Al conectarse al demonio:  
`sudo asterisk -rddd`
  - Desde el CLI:  
`CLI> Set Debug 30`

## Asterisk PBX

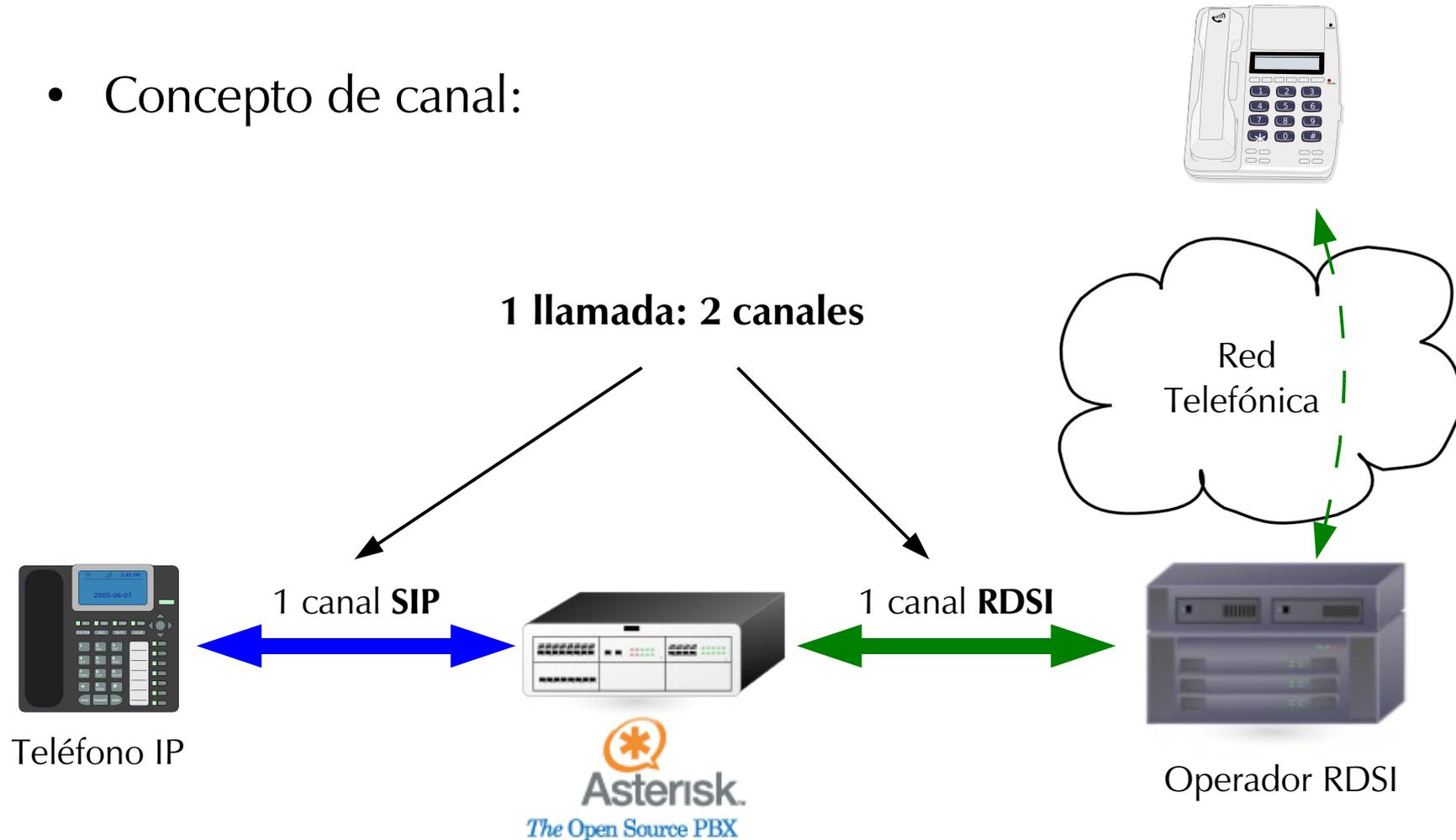
---

### Conceptos Generales sobre Asterisk: Terminología

- **Canal:** Es una conexión que conduce una llamada entrante o saliente en el sistema Asterisk. La conexión puede venir o salir hacia telefonía tradicional analógica o digital o VoIP.
- Asterisk soporta una serie de canales, los más importantes:
  - IAX2, SIP: Protocolos VoIP
  - Zap: Líneas analógicas y digitales.
  - misdn: RDSI

## Asterisk PBX

- Concepto de canal:



## Asterisk PBX

---

### Conceptos Generales sobre Asterisk: Terminología (II)

- **Dialplan:** Se trata de la configuración de la centralita Asterisk que indica el itinerario que sigue una llamada desde que entra o sale del sistema hasta que llega a su punto final. Se trata en líneas generales del comportamiento lógico de la centralita.

### Conceptos Generales sobre Asterisk: Terminología (III)

- **Extension:** En telefonía tradicional, las extensiones se asocian con teléfonos, interfaces o menús. En Asterisk, una extensión es una lista de comandos a ejecutar.
- Las extensiones se acceden cuando:
  - Se recibe una llamada entrante por un canal dado.
  - El usuario que ha llamado marca la extensión.
  - Se ejecuta un salto de extensiones desde el Dialplan de Asterisk.

## Asterisk PBX

---

### Conceptos Generales sobre Asterisk: Terminología (IV)

- **Contexto** (Context): El Dialplan o lógica de comportamiento de Asterisk se divide en uno o varios contextos. Un contexto es una colección de extensiones.
- Los contextos existen para poder diferenciar el 'lugar' donde se encuentra una llamada, para:
  - Aplicar políticas de seguridad: Asterisk no se comporta igual cuando llama un usuario y marca el 1 y cuando un usuario local marca el mismo 1.
  - Menús y submenus diferenciados.
  - En general, es una forma de diferenciación.

## Asterisk PBX

---

### Conceptos Generales sobre Asterisk: Terminología (V)

- **Aplicación** (Application): Asterisk ejecuta secuencialmente los comandos asociados a cada extensión. Esos comandos son realmente aplicaciones que controlan el comportamiento de la llamada y del sistema en sí. Algunos ejemplos:
  - Hangup: Colgar la llamada.
  - Monitor: Comenzar la grabación a disco de la llamada.
  - Dial: Realiza una llamada saliente.
  - Goto: Salta a otra extensión o contexto.
  - PlayBack: Reproduce un fichero de sonido.

### Configuración de Asterisk

- Asterisk puede configurarse desde varios puntos, los más importantes son:
  - Pare desde el propio CLI
  - Desde los ficheros de configuración (.conf) en /etc/asterisk
- La configuración se carga al iniciar Asterisk, por lo que para aplicar cualquier cambio será necesario recargarla, para ello basta con ejecutar el comando *core reload* en el cli:

```
pbxubuntu01*CLI>core reload
```

## Configuración Simple:

---

- misdn.conf
- zapata.conf
- **sip.conf**
- **extensions.conf**
- iax.conf
- features.conf
- voicemail.conf
- meetme.conf
- queues.conf
- ...

## Asterisk PBX: Voz sobre IP

---

### Configuración para canales de Voz IP: SIP e IAX2

- Los ficheros a manipular son sip.conf e iax.conf, la instalación crea ficheros de ejemplo con la sintaxis bastante comentada a modo de guía.

### SIP.CONF

- En este fichero se definen:
  - Variables generales de SIP.
  - Clientes SIP.
  - Servidores SIP.

## Asterisk PBX: Voz sobre IP

---

### SIP.CONF: Sección General

- En primer lugar existe la sección [general], donde se definen variables globales y aspectos por defecto para todos los canales SIP.
- La syntaxis es la siguiente:

```
[general]  
variable1=valor1  
variable2=valor2
```

### SIP.CONF: Sección General

- Las variables generales más importantes son:
  - **allow y disallow**: indican los codecs permitidos / no permitidos.
  - **dtmfmode**: permite especificar el método por el cual se enviaran los tonos (digitos pulsados durante la conversación), valores posibles:
  - **nat**: Informa a Asterisk del tipo de NAT en el que se encuentra.
  - **externip**: Dirección Pública tras el NAT.
  - **context**: Contexto por defecto donde entraran las llamadas entrantes por SIP.
  - **port**: Puerto en el que escuchar (5060 ). \_\_\_\_\_

## Asterisk PBX: Voz sobre IP

---

### SIP.CONF: Clientes y Servidores

- En sip.conf se definen tanto los clientes que se conectarán a Asterisk, como los proveedores que se utilizarán para encaminar llamadas. Conceptualmente, se distinguen
  - user: Envía llamadas a Asterisk
  - peer: Recibe llamadas de Asterisk (proveedor).
  - friend: Recibe y Envía llamadas (usuario).
- La sintaxis para definir un friend o un peer es:

```
[nombre]  
type = friend / peer  
variable = valor  
variable2 = valor  
....
```

## Asterisk PBX: Voz sobre IP

---

### SIP.CONF: Clientes y Servidores

- Las variables más importantes que deben ser configuradas inicialmente son:
  - type: peer / friend
  - context: Contexto donde entraran las llamadas generadas.
  - nat: Indica si el usuario o peer se encuentran tras un nat.
  - host: IP remota o dynamic.
  - username: nombre de usuario.
  - secret: contraseña de acceso.
  - allow y disallow: Configuraciones de codecs específicas para cada friend/peer.
  - qualify: Evalúa el estado del extremo SIP para conocer su accesibilidad y latencia.

## Asterisk PBX: Voz sobre IP

---

### SIP.CONF: Ejemplo, declarando clientes:

- Vamos a declarar de forma básica dos clientes en el fichero sip.conf:

```
[jon]  
type = friend  
secret = jon  
disallow=all  
allow = g729  
allow = gsm  
host=dynamic
```

```
[leire]  
type = friend  
secret = leire  
allow = all  
host=dynamic
```

## Asterisk PBX: Voz sobre IP

---

### SIP.CONF: Ejemplo, declarando peers:

- Vamos a configurar un proveedor en sip.conf:

```
[sarenet]
type=peer
host = 10.11.114.200
disallow = all
allow = g729
allow = gsm
fromuser= 100
secret=ghost
```

### SIP.CONF: Verificación de la configuración con el CLI

- Mediante el comando *core reload* en el CLI de Asterisk, le indicamos que recargue la configuración. Aunque es posible recargar de forma independiente: CLI> sip reload
- Una vez recargada, podemos comprobar los “friends” que hemos definido con el comando: sip show users
- Para ver los “peers” definidos: sip show peers

## Asterisk PBX: Dialplan

---

### Introducción al Dialplan

- Hasta el momento conocemos como configurar usuarios del sistema Asterisk y proveedores externos.
- Pero: ¿ Que sucede cuando un usuario marca un determinado número ? ¿ Como podemos llamar utilizando alguno de los proveedores configurados ?
- La respuesta a esta y muchas más preguntas es: el Dialplan. Es el corazón del comportamiento de Asterisk, en él se configura toda la lógica.

## Asterisk PBX: Dialplan

---

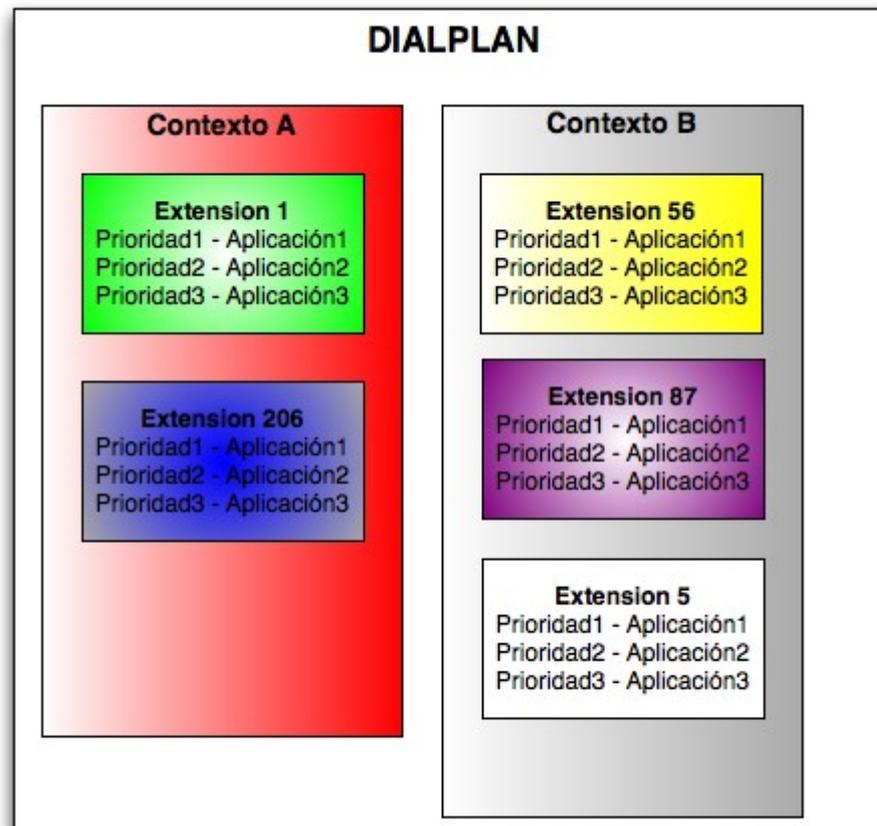
### Introducción al Dialplan (II)

- En lenguaje natural, un ejemplo muy sencillo podría ser el siguiente:
  - Cuando un usuario marca un número:
    - Si el número empieza por 0, llamar al destino utilizando un proveedor externo.
    - Si el número tiene 3 cifras y empieza por 1, llamar a un determinado usuario de la centralita.
    - Si cuando llamamos a ese usuario, no coge en 60 segundos, reproducir un mensaje de alerta.
- En situaciones normales, el dialplan se puede complicar considerablemente..

## Asterisk PBX: Dialplan

### Arquitectura del dialplan

- El dialplan se define en extensions.conf, su “forma” genérica se asemeja a este esquema:



## Asterisk PBX: Dialplan

---

### Flujo en el dialplan: Ejemplo simplificado

- Contexto “desde oficina”
  - Extensión 100, Prioridad 1, Aplicación: Reproducir Canción
  - Extensión 100, Prioridad 2, Aplicación: Llamar a Jon
  - Extensión 100, Prioridad 3, Aplicación: Colgar

## Asterisk PBX: Dialplan

---

### Flujo en el dialplan: Ejemplo simplificado aplicado

- La sintaxis en extensions.conf correspondiente sería:

```
[desde-oficina]  
exten => 100,1,PlayBack(thebeatles)  
exten => 100,2,Dial(SIP/jon)
```

- Y al declarar el friend se hace necesario especificar el contexto:

```
[usuario1]  
type=friend  
....  
context=desde-oficina
```

## Asterisk PBX: Dialplan

---

### Flujo en el dialplan: Ejemplo simplificado aplicado (II)

- La prioridad “n”, es equivalente a sumar uno a la prioridad anterior. Esto permite editar y modificar rápidamente el DialPlan.
- La sintaxis en extensions.conf equivalente sería:

```
[desde-oficina]  
exten => 100,1,PlayBack(thebeatles)  
exten => 100,n,Dial(SIP/jon)
```

### Aplicaciones en el dialplan

- Para obtener un listado de las aplicaciones disponibles, desde el CLI se puede ejecutar: `core show applications`
- Para obtener una descripción de una determinada aplicación: `core show application nombre`

### Aplicaciones en el dialplan: Aplicaciones Generales

- Las aplicaciones generales más importantes son:
  - Wait (n)
    - Espera n segundos, ignorando los dígitos marcados durante.
  - WaitExten (n)
    - Espera n segundos, pero gestionando los dígitos marcados.
  - WaitMusicOnHold(n)
    - Reproduce música en espera durante n segundos.

## Asterisk PBX: Aplicaciones en el dialplan

---

### Aplicaciones en el dialplan: Gestión de llamadas

- Los comandos de gestión de llamadas más importantes:
  - Answer()
    - Acepta la llamada entrante por el canal.
  - Busy()
    - Envía la señal de ocupado al origen.
  - Hangup()
    - Cuelga la llamada.
  - Ringing()
    - Envía la señal de tono de llamada.
  - Dial ( tipo/identificador,timeout,opciones,url )
    - Aplicación para llamar (ver más adelante).

### Aplicaciones en el dialplan: Control de Flujo

- Algunos comandos de control de flujo y temporización:
  - Goto (contexto, extension, prioridad)
    - Salta al contexto, extensión y prioridad del argumento.
  - GotoIf (condicion ? prioridad1 : prioridad2 )
    - Salta a la prioridad1 si la condición se cumple.
    - Salta a la prioridad2 si la condición no se cumple.
  - GotoIfTime(<times> | <weekdays> | <mdays> | <months>?[[context | ]exten | ]priority)

## Asterisk PBX: Aplicaciones en el dialplan

---

### Aplicaciones en el dialplan: Reproducción de sonidos

- Algunas aplicaciones para la reproducción:
  - PlayBack (fichero )
    - Reproduce el fichero, continua la ejecución cuando finaliza.
  - Background (fichero)
    - Reproduce el fichero, pero continua la ejecución inmediatamente.
  - SayDigits ( dígitos)
    - Reproduce los dígitos.

## Asterisk PBX: Aplicación Dial

### Aplicación Dial

- La aplicación Dial realiza una llamada a un determinado destino, si el destino acepta la llamada, Asterisk 'une' el origen primario de la llamada con este nuevo interlocutor.
- No importa la tecnología o protocolos utilizados: Un canal entrante por RDSI puede finalizar en una llamada por IP, un usuario SIP puede llamar a otro usuario IAX.



## Asterisk PBX: Aplicación Dial

---

### Aplicación Dial (II)

- La sintaxis genérica del comando es:  
`Dial ( Tecnología / identificador | timeout | opciones | URL )`
- Siendo:
  - Tecnología: SIP, IAX2, ZAP, MISDN, ...
  - Identificador: Es el destino (número o nombre) que se llamará por el canal. La sintaxis depende del canal.
  - Timeout: Tiempo máximo de espera.
  - Opciones
  - URL: Envía una URL al destino, si lo soporta.

## Asterisk PBX: Aplicación Dial

---

### Aplicación Dial (III)

- Algunas de las posibles opciones:
  - t: permite que la llamada sea transferida por el destino.
  - T: permite que la llamada sea transferida por el origen.
  - m: reproduce música en espera mientras el destino acepta la llamada.

## Asterisk PBX: Aplicación Dial

---

### Aplicación Dial (IV)

- Identificadores en base a tecnología:
  - SIP
    - usuario o extensión destino : llamada directa
    - usuario o extensión destino @ peer
    - Ejemplos: Dial(SIP/202) , Dial(SIP/201 | 100 | t), Dial (SIP/944416024@proveedor)
  - IAX2
    - peer / numero destino
    - Ejemplo: Dial (IAX2/peerdestino/902123123)

### Manejo de Extensiones

- La syntaxis general en el dialplan es:  
`exten => EXTENSION, PRIORIDAD, Aplicación`
- En el caso de llamadas internas o funcionamiento simple, las extensiones son conocidas.
- ¿Pero que pasa cuando un usuario llama a un número que no se pre-conoce ?
- Solución: Utilizar patrones en las extensiones.

### Manejo de Extensiones: Patrones de Coincidencia

- Para indicar patrones, se utiliza el carácter: “\_”
- Se pueden utilizar:
  - X: Indica un dígito del 0 al 9
  - Z: Indica un dígito del 1 al 9
  - N: Indica un dígito del 2 al 9
  - [129] Indica el 1, 2 o 9
  - . Indica uno o más caracteres (¡Atención! Coincide con las extensiones especiales: h,i,t .., recomendable: \_X.)
- Ejemplos:
  - Fijos Nacionales: **exten=> \_9XXXXXXXXXX**
  - Internacionales: **exten=> \_00.**

## Asterisk PBX: Variables

---

### Variables

- En el Dialplan de Asterisk existen variables, que pueden ser modificadas por el propio Asterisk en su ejecución lógica o por comandos expresos del Dialplan.
- Las aplicaciones pueden cambiar variables.
- Los tipos de variables son:
  - Globales: Declaradas en extensions.conf (o por comando).
  - Canal: Son propias a cada canal.
  - Entorno: Variables de entorno (UNIX Like).
- La sintaxis de una variable es:

`#{variable}`

## Asterisk PBX: Variables

---

### Manejo de Variables

- Asignación de variables:
  - SetVar(Variable=valor)
  - SetGlobalVar(Variable=valor)
- Manejo de cadenas:
  - Subcadenas: `${Variable : offset : longitud }`
    - Devuelve la subcadena de variable que comienza en offset y con la longitud especificada.
    - Ejemplo: `${ 123456789:2:3}` devuelve 345
  - Longitud: `${LEN(Variable)}`
  - Concatenación: `${Variable1}${Variable2}`

## Asterisk PBX: Variables

---

### Variables de canal definidas automáticamente

- Listado de variables más importantes:
  - `${CALLERID}`: Caller ID actual, nombre y número.
  - `${CONTEXT}`: Contexto actual.
  - `${EXTEN}`: Extensión actual.
  - `${CHANNEL}`: Canal actual.
  - `${DIALSTATUS}`: Estado de la llamada: unavailable, congestion, busy, noanswer, answer, cancel, hangup.
  - `${DATETIME}`: Hora actual.
- Un comando útil para ver el contenido es NoOp:
  - NoOp ( `${VARIABLE}` )
    - Nos mostrará en el CLI el valor.

## Asterisk PBX: Expresiones

---

### Syntaxis de las Expresiones

- Es posible utilizar expresiones en las llamadas a aplicaciones (principalmente: GotoIf)
- Syntaxis:

`$(expr1 operador expr2)`

- Operadores Lógicos: |(or) , &(AND)
- Operadores de Comparación: =, !=, <, >, <=, >=
- Operadores Aritméticos: +, -, \*, /, %
- [...] , Ejemplos:

`exten => 1,1,SetVar(total=$(1 + 1))`

`exten => 1,2,GotoIf(${${CALLERID}=123456}?10:20)`

# CURSO ASTERISK

## Ejemplo I

---

[usuarios]

exten => 300,1,NoOp(Llamada a la secretaria)

exten => 300,n,Dial(SIP/300,10,t)

exten => \_3XX,1,Dial(SIP/\${EXTEN},10)

[superusuarios]

include => usuarios

exten => \_9XXXXXXXX,1,Dial(Zap/1/\${EXTEN},20,m)

exten => \_6XX.,1,Dial(misdn/g:1/\${EXTEN})

## Ariketo

---

### Hotel Eghost (I)

- Recepción
- Cocina
- Reservas

Extensión 500 permite hablar con recepción

Extensión 501 permite hablar con cocina

Extensión 502 permite hablar con reservas sólo los laborables de 9 a 14 y de 16 a 20. En caso contrario reproduce un mensaje de disculpa y cuelga.

## Ariketo I

---

SIP.CONF

[recepcion] ;mejor [500]

type=friend

host=dynamic

disallow=all

allow=alaw

secret=ironsecret

context=servicios-hotel

callerid=Recepcion

qualify=yes

canreinvite=yes

# CURSO ASTERISK

## Ariketo I

---

EXTENSIONS.CONF

[servicios-hotel]

exten => 500,1,Dial(SIP/recepcion,20,t)

exten => 501,1,Dial(SIP/501,20,tm)

exten => 502,1,GotoIfTime(9:00-14:00|mon-fri|\*|\*?on,1)

exten => 502,n,GotoIfTime(16-20:00|mon-fri|\*|\*?on,1)

exten => 502,n,Playback(locuciones/cerrado-se-siente)

exten => on,1,NoOp(Llamada de \${CALLERID} a Reservas)

exten => on,n,Dial(SIP/reservas,,m)

## Ariketo II

---

### Hotel Eghost II

Al hotel le ponen por fin las líneas rdsi y se le asignan dos identificadores (DID): 94 444 1234 & 94 444 1235. Se quiere reservar el primer DID para recepción y el segundo DID se usará para el fax.

Nota: El rdsi también pertenece a un contexto que se supone configurado en misdn.conf

# CURSO ASTERISK

---

[rdsi-entrantes]

exten => 944441234,1,NoOP(Llamada entrante por la rdsi)

exten => 944441234,n,Dial(SIP/500,20,t)

exten => 944441234,n,Voicemail(1000)

exten => 944441235,1,NoOp(Fax entrante)

exten => 944441235,n,Dial(IAX2/800)

[servicios-hotel]

...

## Ariketo III

---

Hotel Eghost III

Queremos que los que llaman desde fuera puedan acceder tanto a recepción como a reservas. Como sólo disponemos de un DID, hay que hacer un IVR para que el llamante pueda elegir con quién contactar:

Nota: Aplicación WaitExten(n)

## Ariketo III

---

[rdsi-entrantes]

exten => 944441234,1,NoOP(Llamada entrante por la rdsi)

exten => 944441234,n,Background(locuciones/bienvenido)

exten => 944441234,n,WaitExten(4)

exten => 1,1,Dial(SIP/500,20,t)

exten => 1,n,Voicemail(1000)

exten => 2,1,GotoIfTime(9:00-14:00|mon-fri|\*|\*?servicios-  
hotel,on,1)

exten => 2,n,GotoIfTime(16-20:00|mon-fri|\*|\*?servicios-  
hotel,on,1)

exten => 2,n,Voicemail (1001)

exten => t,1,Goto(1,1)

exten => i,1,Goto(1,1)

### Asterisk y la Telefonía Tradicional

- Para poder inter-operar con la telefonía tradicional, Asterisk necesita hardware específico.
- El principal sponsor y desarrollador de Asterisk: DIGIUM es el principal fabricante de hardware.

### Asterisk y la Telefonía Tradicional

- Para operar con líneas analógicas, se necesitan tarjetas con interfaces FXO
  - Ejemplo: Digium TDM01B
- Para operar con teléfonos analógicos o centralitas clásicas, se requieren interfaces FXS
  - Ejemplo: Digium TDM10B
- En ambos casos, el driver a utilizar es zaptel, la configuración se almacena en /etc/zaptel.conf

### Asterisk y la Telefonía Tradicional (II)

- En líneas digitales (RDSI), en Europa existen dos tipos:
  - BRI : Acceso básico, proporciona 2 canales de voz.
  - PRI: Acceso primario, proporciona 30 canales de Voz (E1).
- Asterisk soporta perfectamente ambos tipos de líneas digitales, con hardware específico:
  - Para primarios, Digium proporciona tarjetas de hasta 4 puertos, siendo zaptel el driver a utilizar.
  - Para primarios, Digium proporciona tarjetas de hasta 4 puertos, siendo mISDN el driver a utilizar.

### Asterisk y la Telefonía Tradicional: Lineas analógicas

- Para operar con las tarjetas con interfaces FXS / FXO, Asterisk utiliza el subsistema Zaptel: Zapata Telephony
- Es necesario descargar e instalar el paquete, está disponible en el ftp de Digium
- Una vez instalado el sistema zaptel, es necesario configurarlo en /etc/zaptel.conf
  - Definir zonas (para frecuencias de tonos)
  - Definir interfaces en los canales: FXS / FXO
- Se puede verificar la configuración correcta con el comando instalado: **ztcfg -v**

### Asterisk y la Telefonía Tradicional: Lineas analógicas II

- Asterisk utiliza los módulos provistos por zaptel para acceder al hardware.
- Este enlace se configura en: `/etc/asterisk/zapata.conf`
- Aspectos importantes a configurar:
  - Context: contexto donde iran las llamadas generadas por cada canal.
  - echo cancel: cancelación de echo (problema importante en telefonía).
  - Para utilizar correctamente las lineas españolas son necesarios los siguientes parámetros (ya no es necesario parchear asterisk):

`answeronpolarityswitch=yes`  
`hanguponpolarityswitch=yes`

## Asterisk como PBX

---

### Asterisk como Centralita

- Toda la secuencia y programación del dialplan es el verdadero núcleo del sistema centralita, si bien, las siguientes funcionalidades se configuran en features.conf:
  - Transferencias de llamadas: transferencia de llamadas entre diversos usuarios, independientemente de la tecnología que usen.
  - Call Parking: Parking de llamadas.
  - Call Pickup: Auto-transferencia de un teléfono que esté sonando.



## Asterisk como PBX

---

### Asterisk como Centralita: Transferencias (II)

- Para poder realizar transferencias vía PBX, Asterisk debe estar a la escucha de los tonos configurados, para ello:
  - La aplicación Dial debe haber recibido como parámetro 't' para permitir transferencia en destino o 'T' para hacerlo en origen (o ambas).
  - En el caso de SIP, los DTMF pueden ser enviados de varias formas (RFC2833/out of band, SIP Info,inband), Asterisk debe saber como los enviará el cliente SIP.
- Ejemplo de llamada Dial:

exten => 100, 1, NoOp( llamando a Peio)

exten => 100, n, Dial ( SIP/peio|10|t)

## Asterisk como PBX

---

### Asterisk como Centralita: Música en Espera

Asterisk puede poner un canal dado en espera ('HOLD'), principalmente en las siguientes situaciones:

- Durante una transferencia.
- Durante una llamada si se ha especificado el parámetro 'm', que indica que no se oirá tono de llamada sino música en espera.
- Durante una espera en el parking.
- Si la aplicación MusicOnHold o WaitMusicOnHold ha sido llamada desde el DialPlan
- Si el destino de la llamada ha solicitado explícitamente que la llamada sea puesta en espera
- Es posible tener distintos tipos de música en espera.
- La música en espera se configura en *musiconhold.conf*

## Asterisk como PBX

---

### Asterisk como Centralita: CallParking (I)

- El callparking es una funcionalidad que permite transferir la llamada a un 'parking' virtual. Pudiendo colgar sin que la llamada origen sea desconectada de Asterisk, ya que se encontrará aparcada,
- Para operar, el usuario transfiere la llamada a una extensión especial (parkext en features.conf), Asterisk aparca la llamada y anuncia la posición en el parking.
- Cualquier usuario que tenga incluido en su contexto el contexto especial 'parkedcalls' puede recuperar la llamada aparcada llamando directamente a su posición en el parking.

## Asterisk como PBX

### Asterisk como Centralita: CallParking (II)

1) A y B están en conversación.



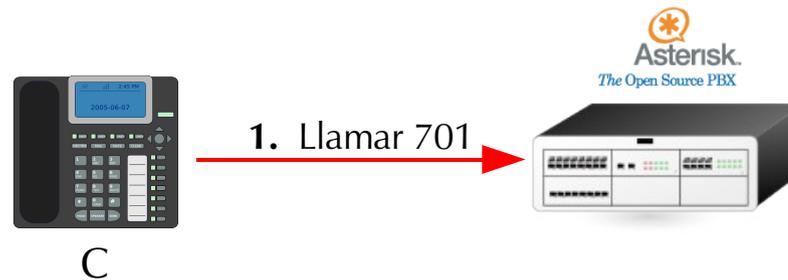
2) A transfiera al 700 y 'aparcas' a B en el Parking 701.



## Asterisk como PBX

### Asterisk como Centralita: CallParking (III)

- 3) Desde otro teléfono C se puede recoger a B del Parking marcando el 701.



- 4) C y B están en conversación.



## Asterisk como PBX

---

### Asterisk como Centralita: CallPickup

- 'Callpickup' es el hecho de poder descolgar y responder a la llamada entrante a un teléfono o grupo de teléfonos determinado desde un tercer teléfono que no está sonando.
- La configuración de los códigos DTMF para 'Callpickup' se configura en features.conf
- Se definen dos conceptos:
  - callgroup: Toda llamada que entra a una extensión determinada pertenece al/los callgroups de la extensión.
  - pickupgroup: Es el/los callgroups de llamadas entrantes que el usuario puede hacer 'pickup'.
- En cada usuario de la centralita se define esos dos parámetros.

## Asterisk como PBX

---

### Asterisk como Centralita: Colas de llamadas

- Una llamada entrante puede ser enviada a una cola de llamadas, que será gestionada por determinados usuarios.
- Se utilizan mucho en entornos tipo 'callcenter', con los canales tipo de Agentes (que hacen 'login en el sistema').
- Las colas pueden comportarse de forma distinta:
  - Suena todos los teléfonos hasta que alguno descuelgue.
  - Los teléfonos van sonando en orden
  - ...
- Existen colas con prioridad.
- Las colas de llamadas se configuran en queues.conf:

## Asterisk como PBX

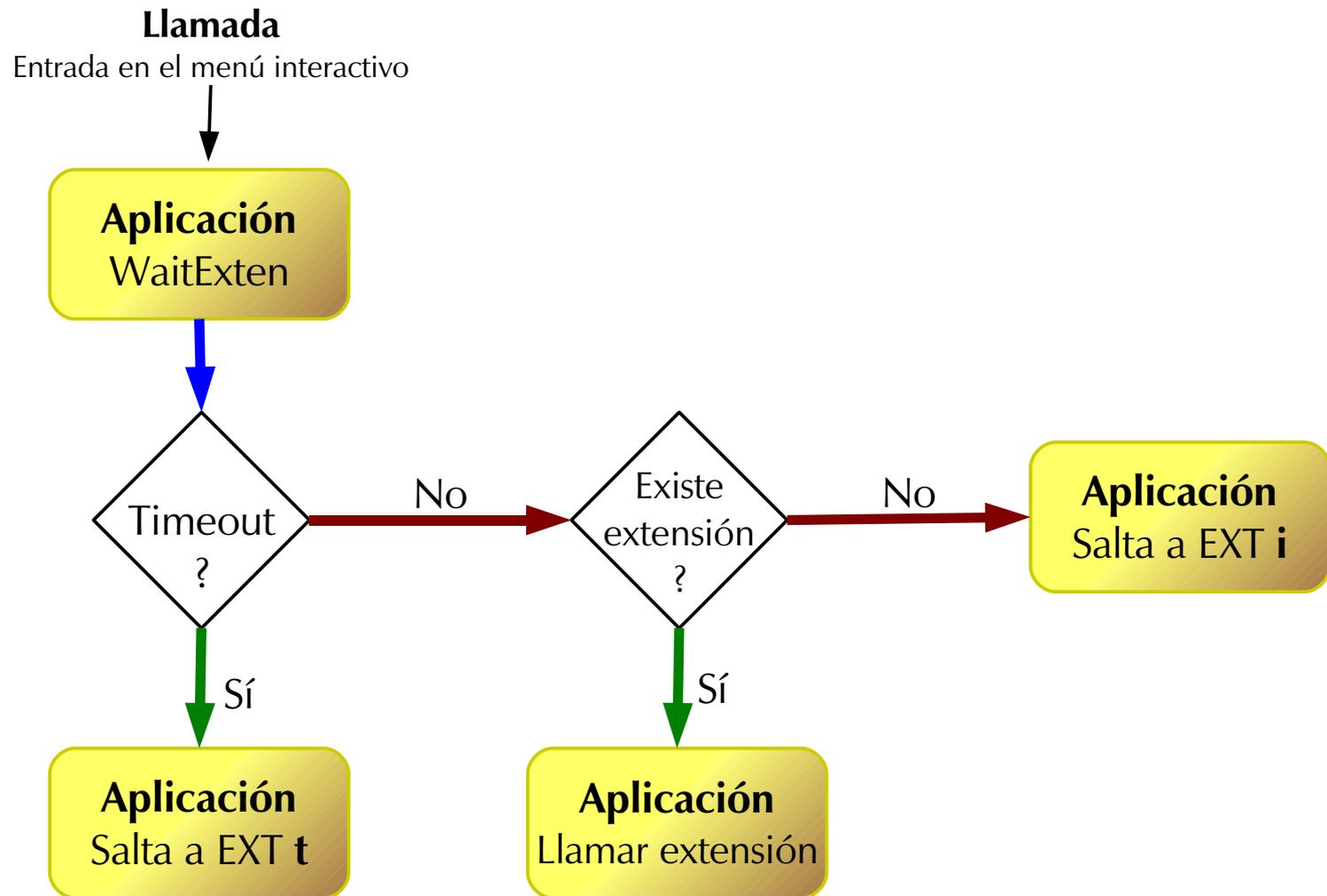
---

### Asterisk como Centralita: IVR

- **IVR:** Interactive Voice Response
- El modo de funcionamiento es bastante sencillo:
  - Al hacer una llamada a la aplicación WaitExten, Asterisk se queda a la espera de que el usuario marque una extensión.
  - En caso de timeout (el usuario no marca nada en el tiempo especificado), Asterisk salta a la extensión 't' si existe.
  - En caso contrario, Asterisk busca la extensión y la ejecuta, sino existe, salta a la extensión 'i' si existe.
- Se utiliza casi siempre en combinación con la aplicación BackGround (Se comienza a reproducir el fichero y seguido se entra en espera, para que el usuario pueda introducir extensiones sin tener que esperar)..

## Asterisk como PBX

### Asterisk como Centralita: IVR (II)



## Asterisk como PBX

---

### Asterisk como Centralita: IVR (III)

- Ejemplo de menú:

```
[menu-ivr]
```

```
exten => s, 1, Answer()
```

```
exten => s, n, BackGround(intro-menu)
```

```
exten => s, n, WaitExten(10)
```

```
exten => 1, 1, Goto(entrantes-dptocomercial,s,1)
```

```
exten => 2, 1, Goto(entrantes-dptocomercial,s,1)
```

```
exten => i, 1, Playback(opcion-invalida)
```

```
exten => i, n, Goto(menu-ivr,s,1)
```

```
exten => t, 1, Goto(entrantes-generales)
```

### Registro de llamadas

- Asterisk permite llevar un control exhaustivo de todas las llamadas que se han realizado o recibido.
- Es interesante para control propio de facturación, independiente del proveedor (si no lo somos).
- Permite realizar estadísticas.
- Este control se denomina: CDR, Call Detail Record

### Registro de llamadas (II)

- El registro del CDR se escribe por defecto en el fichero `/var/log/asterisk/cdr-csv/Master.csv`
- Existen extensiones al cdr: `cdr_mysql` por ejemplo, que permiten almacenar los registros en una base de datos.
- `cdr_mysql` está disponible en `asterisk-addons`
- El CDR se configura en el fichero `cdr.conf`, para el módulo de MySQL, se utiliza `cdr_mysql.conf`
- Para confirmar el estado del CDR desde el CLI, se puede ejecutar:

CLI> `cdr status`

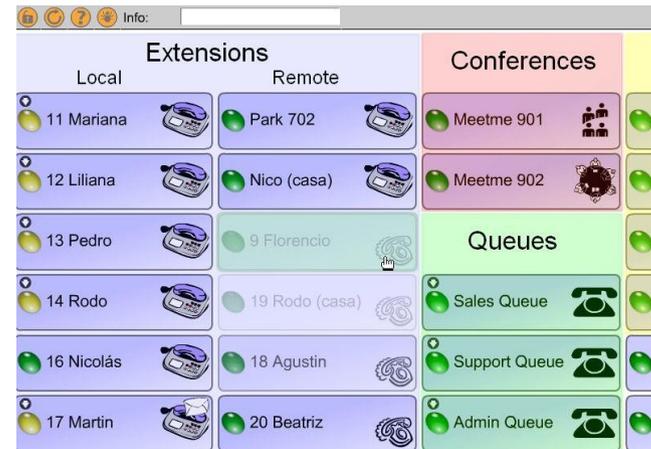
### Monitorización

- Asterisk provee el protocolo AMI: Asterisk Manager TCP/IP API
- Utilizando este protocolo, Asterisk permite:
  - Crear llamadas
  - Monitorizar llamadas
  - Monitorizar canales y colas
  - Ejecutar comandos
- Se trata de un demonio más de Asterisk, que escucha en el puerto tcp 5038
- Se configura en manager.conf

## Asterisk PBX: Control de llamadas

### Monitorización (II)

- Una de las mejores aplicaciones opensource para controlar y visualizar el estado en tiempo real de la centralita Asterisk es: Flash Operator Panel (FOP).
- FOP ha sido desarrollado en Perl y MING, y se compone de dos partes:
  - Servidor de datos
  - Flash Applet



## Asterisk PBX: Integración

---

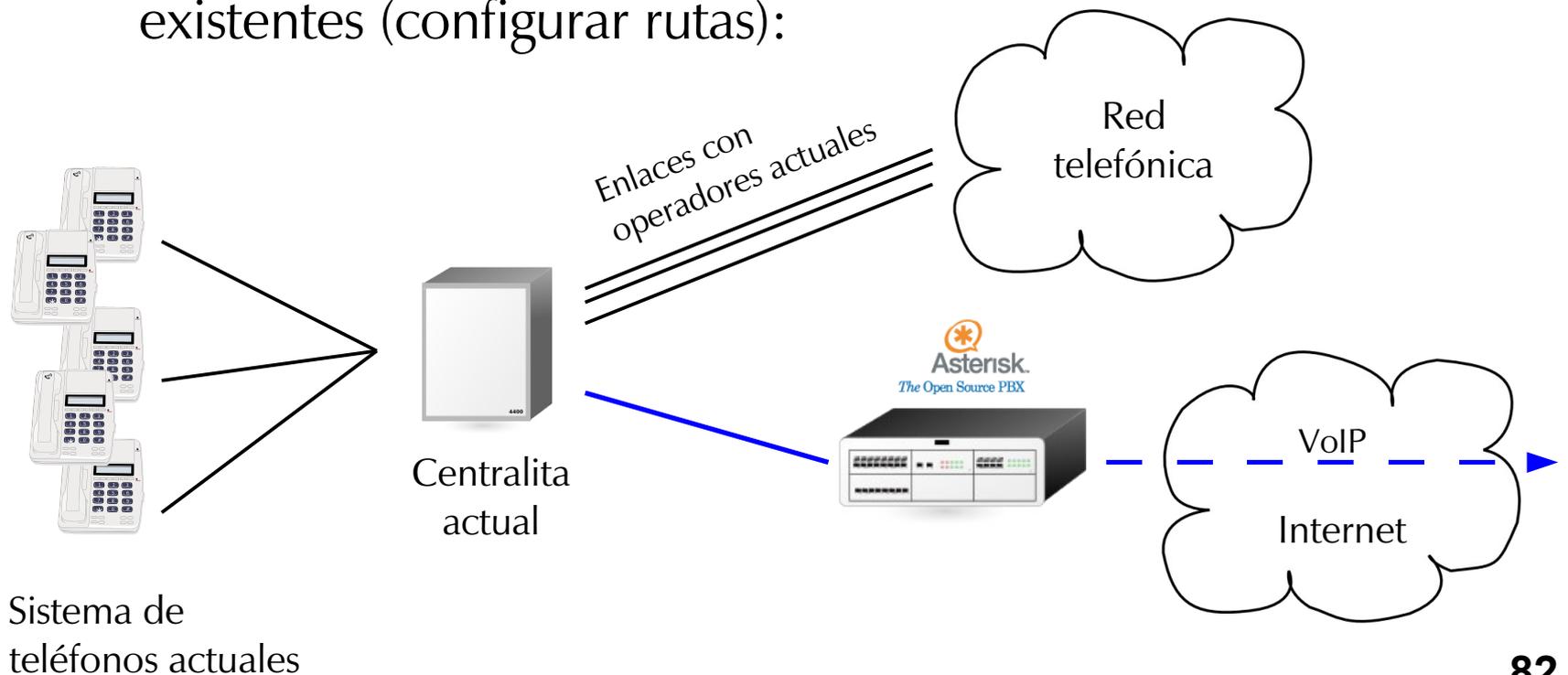
### Integración

- Asterisk puede ser implantado como un centralita analógica, digital e IP sin problemas.
- Sin embargo, en muchos entornos las centralitas o sistemas telefónicos acaban de ser renovados y plantear un cambio drástico no es viable.
- Se ofrecen varias posibilidades, entre ellas:
  - Integración como pasarela a IP
  - Integración como servicios adicionales

## Asterisk PBX: Integración

### Integración como pasarela a IP (I)

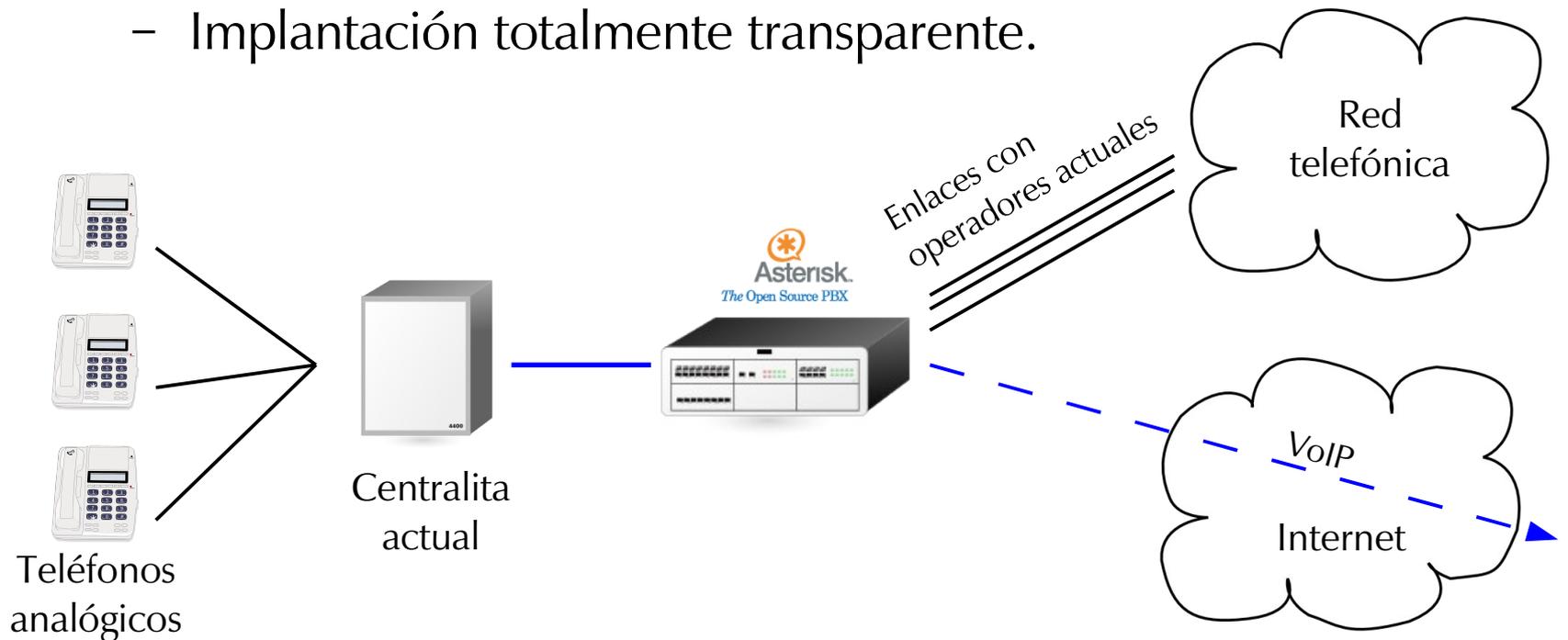
- Asterisk puede suponer un abaratamiento de costes muy importante, configurandolo como pasarela a VoIP.
- La implantación es casi transparente para los sistemas existentes (configurar rutas):



## Asterisk PBX: Integración

### Integración como pasarela a IP (II)

- De forma transparente para la actual centralita:
  - El mismo esquema pero cediendo la lógica de enrutamiento a Asterisk.
  - Implantación totalmente transparente.



## Asterisk PBX: Integración

---

### Integración como servicios adicionales

- La posibilidad de ser implantado en 'cabeza' de la centralita actual, permite:
  - Ampliar el número de extensiones actuales (ya sean IP o tradicionales), colgando las nuevas extensiones del nuevo Asterisk.
  - Crear un sistema de buzón de voz avanzado
  - Crear un menú IVR (requiere la configuración los DIDS en la central actual).

## Asterisk PBX: Integración

---

### Integración como centralita IP

- Uno de los servicios en los proveedores de VozIP son los números 90x y geográficos por IP, tanto nacionales como internacionales.
- La ventaja de basarse en telefonía IP es la facilidad de crecer en extensiones.
- El proveedor nos asigna un número de la numeración tradicional y nos encamina las llamadas por SIP/IAX2...
- Esta nueva posibilidad supone gran movilidad y ahorro de costes: no es necesario tener más líneas, únicamente tener ancho de banda y una latencia aceptable.
- Sin embargo: Dependencia de Internet.

## AGI: Asterisk Gateway Interface

---

### ¿Qué es AGI?

- Interfaz para comunicar Asterisk con programas externos en diferentes lenguajes de programación:  
C, C#, Bourne Shell, PHP, Ruby, Python, Perl, Java
- El programa debe cumplir:
  - Ser ejecutable (mediante un intérprete es válido).
  - Localizado por defecto en `/var/lib/asterisk/agi-bin`
  - Asociado con alguna extensión en el Dial Plan:  
`exten => 200,1,AGI(mi_programa.php|argumentos)`
- Comunicación con Asterisk:
  - El programa escribe en la salida estándar para enviar un comando a Asterisk.
  - El programa lee de entrada estándar para obtener una respuesta de Asterisk.

## Asterisk: Auto-dial out

---

### Auto-dial out

- Permite iniciar llamadas desde aplicaciones externas.
- Se copia un fichero tipo call(callfiles) en:  
/var/spool/asterisk/outgoing/
  - Asterisk inmediatamente llamará al canal especificado en Channel y lo conectará con el contexto dado (también es posible especificar una aplicación).
- Ejemplo:

```
Channel: SIP/bt100
```

```
MaxRetries: 1
```

```
RetryTime: 60
```

```
WaitTime: 30
```

```
Context: outgoing
```

```
Extension: 944416024
```

```
Priority: 1
```

## Referencias

---

### Bibliografía

- *Asterisk, The Future of Telephony*. Jim Van Meggelen, Jared Smith & Leif Madsen – O' Reilly

### Sitios Web

- *Voip-info*: <http://www.voip-info.org>

### Lista de correo

- Asterisk-es

## Licencia

---



C O M M O N S D E E D

Copyright © 2007 Irontec <contacto@irontec.com>

- Detalles de la licencia:
  - <http://creativecommons.org/licenses/by/2.5/es/deed.es>