

# El futuro de la telefonía y la VoIP ha llegado



Asterisk<sup>TM</sup>



# Antes de empezar

---

- Conocimientos previos recomendados:
  - Manejo básico de la consola de GNU/Linux.
  - Interés
  - Ganas de aprender
- ¿Qué sabes de Asterisk?
  - ¿Lo has utilizado?
  - ¿Que intenciones de uso tienes?



# Asterisk: Introducción e Historia



# ¿Qué es Asterisk?

---

- Software
  - Cumple todas las funcionalidades de una centralita tradicional y más.
  - Open Source.
  - **“Asterisk is an OpenSource Multiprotocol PBX”**
- Distintos tipos de uso
  - Para uso doméstico.
  - Uso empresarial (ABE)
  - Operadores de telefonía IP.
  - etc...



# Historia

---

- Inicialmente desarrollado por Mark Spencer (creador de GAIM).
- Tenía una empresa (Linux Support Services Inc.) y necesitaba una centralita.
- Decidió hacerla él mismo.
- Se juntó con Jim Nixon, originalmente soportaba las tarjetas de Zapata Telephony
- Liberó el código bajo licencia GPL.



## Historia (2)

---

- Viendo la evolución de Asterisk, la empresa cambió de nombre a Digium.
- Actualmente Mark no es el CEO, es el CTO.
- Recientemente adquirieron Sokol & Associates, empresa creadora del AstriCon y de la formación de Asterisk.



# Versiones de Asterisk

---

- Principalmente hay 2 versiones
- La versión **Open Source.**
- Asterisk Business Edition: edición comercial.
  - Se basa en la Open Source, pero se eliminan todos aquellos elementos susceptibles de causar problemas.
  - Fuertemente probada.
  - Al comprarla incluye soporte durante un año.
  - Licencias.



# Ramas de Asterisk

---

- Actualmente conviven 2 ramas de Asterisk: la 1.2 y la 1.4.
- En Agosto de 2007, la rama 1.2 pasará al estado de “mantenimiento de seguridad”, y solo se corregirán bugs relativos a la seguridad.
- El curso esta basado en la última versión estable de la rama 1.4
- Asterisk 1.6: mucho esfuerzo en cuanto a su escalabilidad. Se espera que sea la versión que posicione en la cima Asterisk.





# Distintos usos de Asterisk



# Funcionalidades de Asterisk

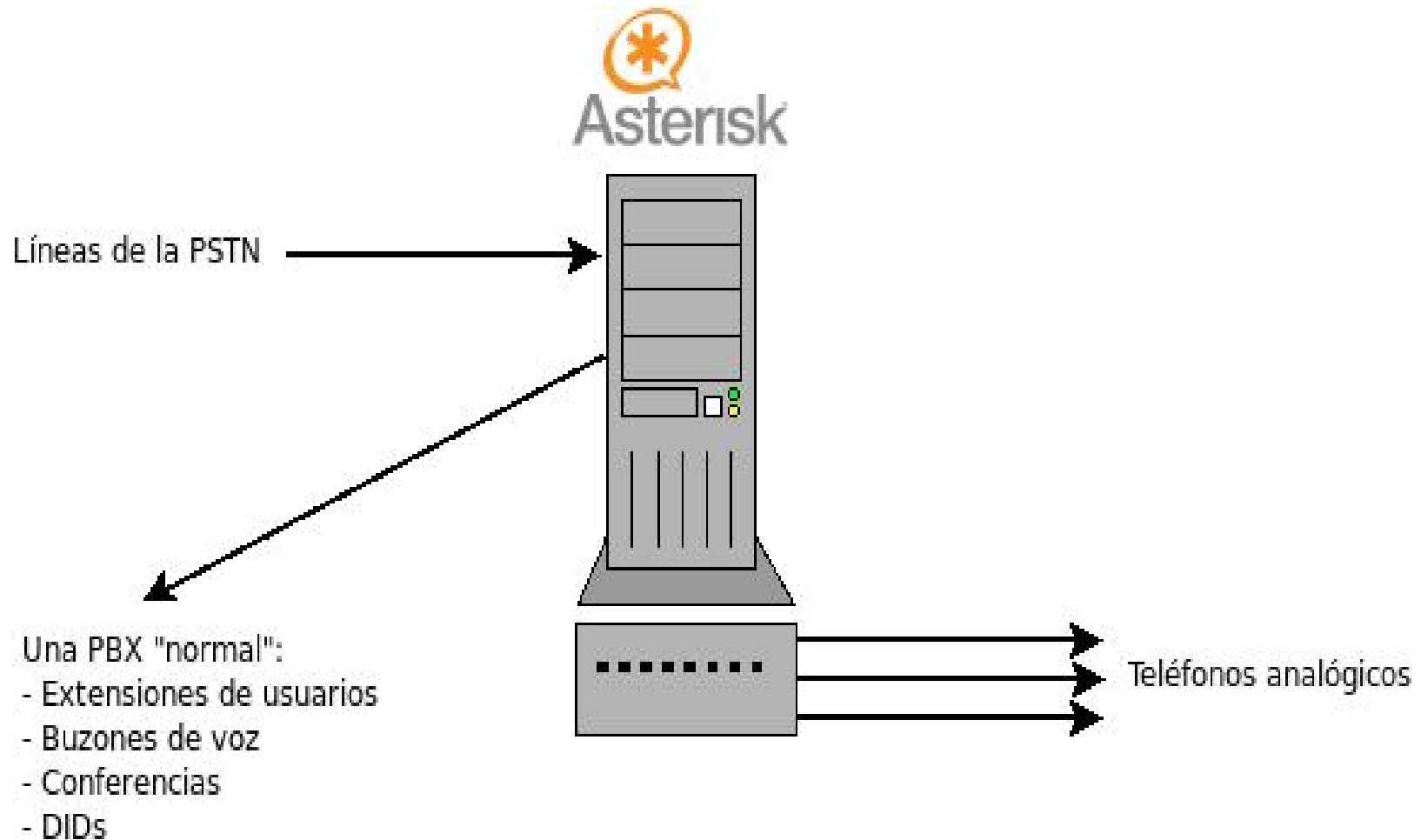
---

- Características típicas de centralita:
  - Transferencias
  - Música en espera
  - ...
- **Multiprotocolo y OpenSource!!**
- Extensiones, DIDs para usuarios.
- Buzones de voz, desvíos de llamada, follow-me, ...
- Colas y agentes.
- Menús IVR.
- Protocolos: SIP, IAX2, H.323, MGCP, ...
- PSTN: T1/E1, ISDN BRI, FXO/FXS.
- Soporte de vídeo (Asterisk 1.6: “the video release”).



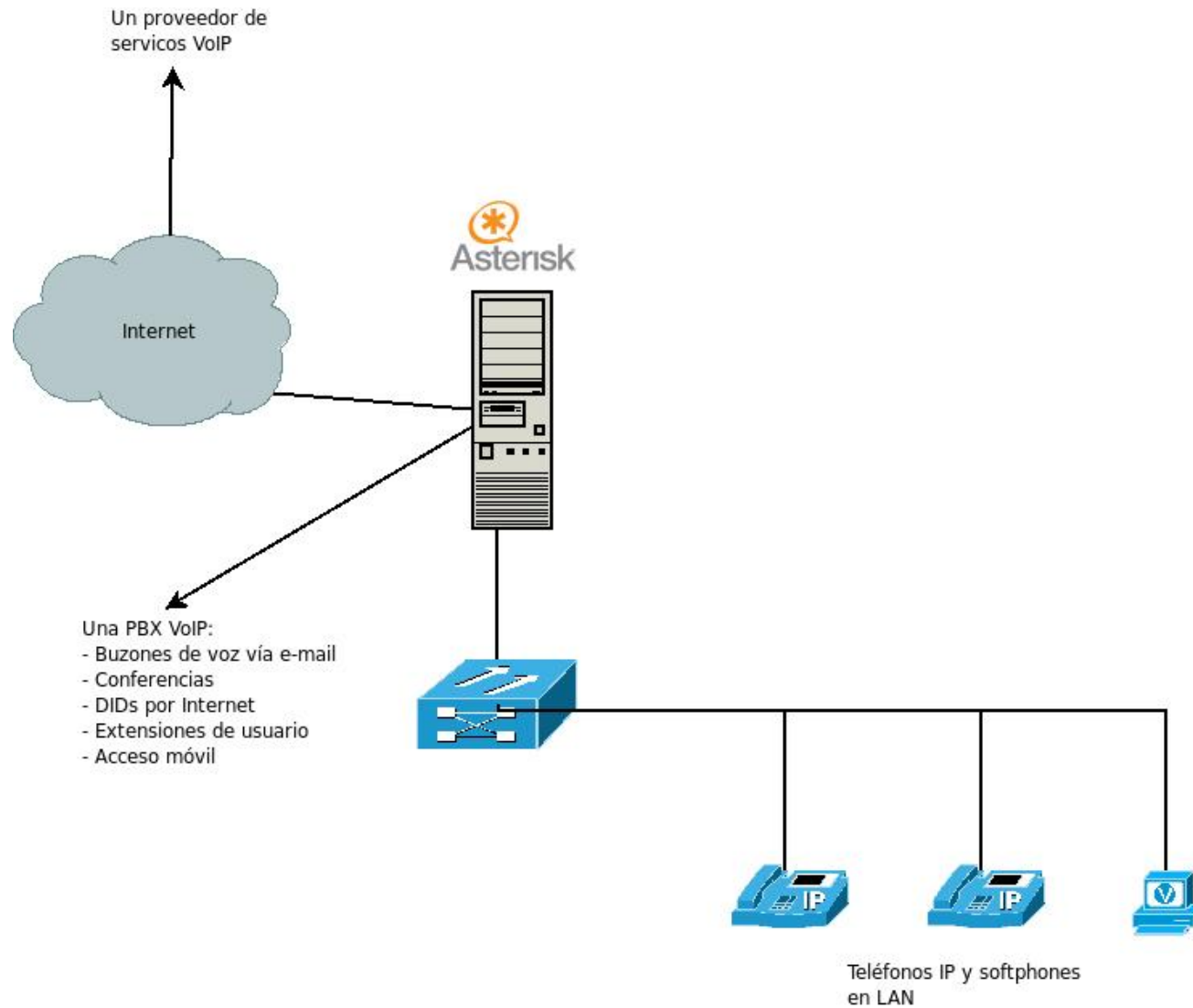
# Asterisk como PBX "normal"

---



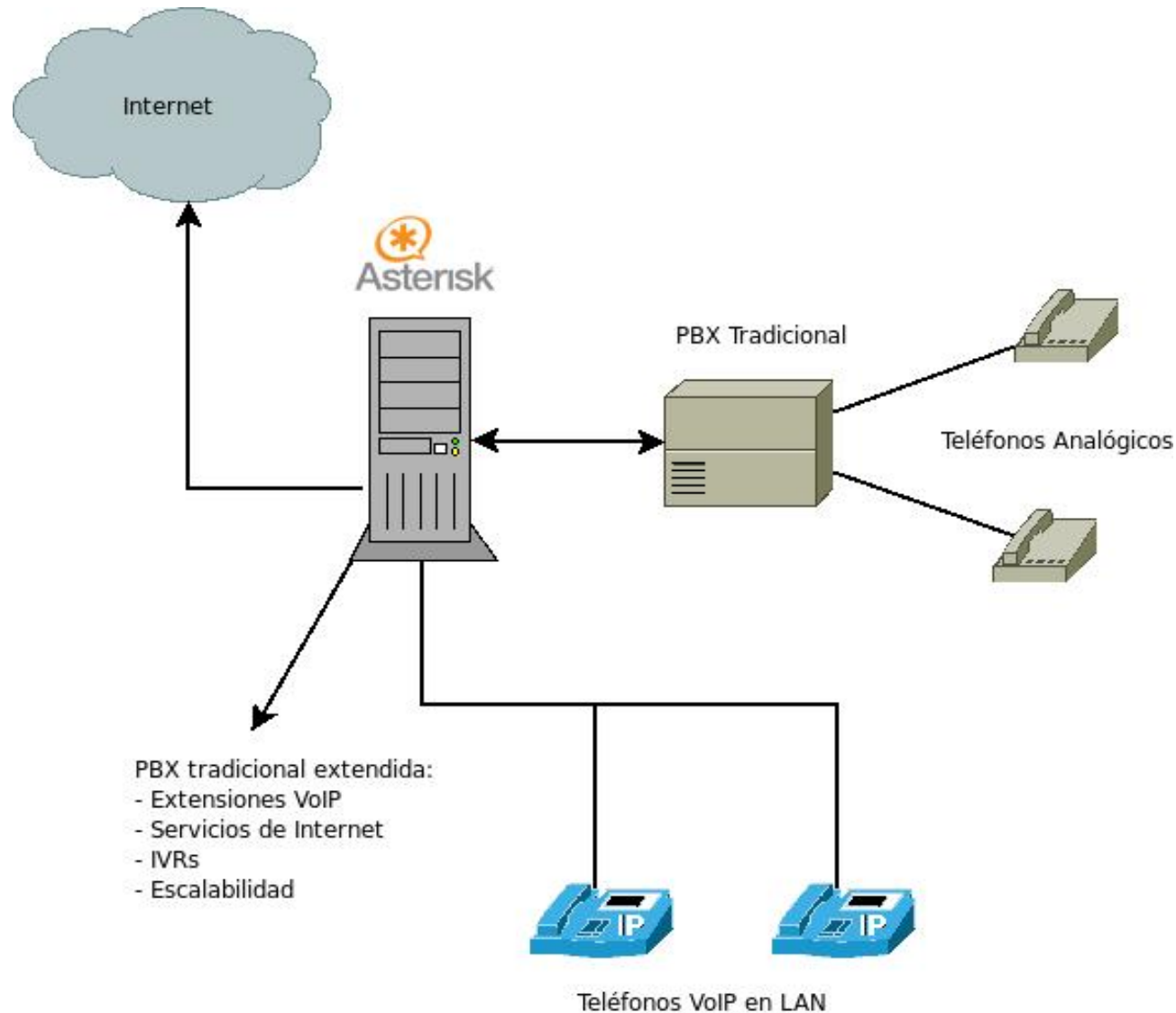


# Una PBX VoIP





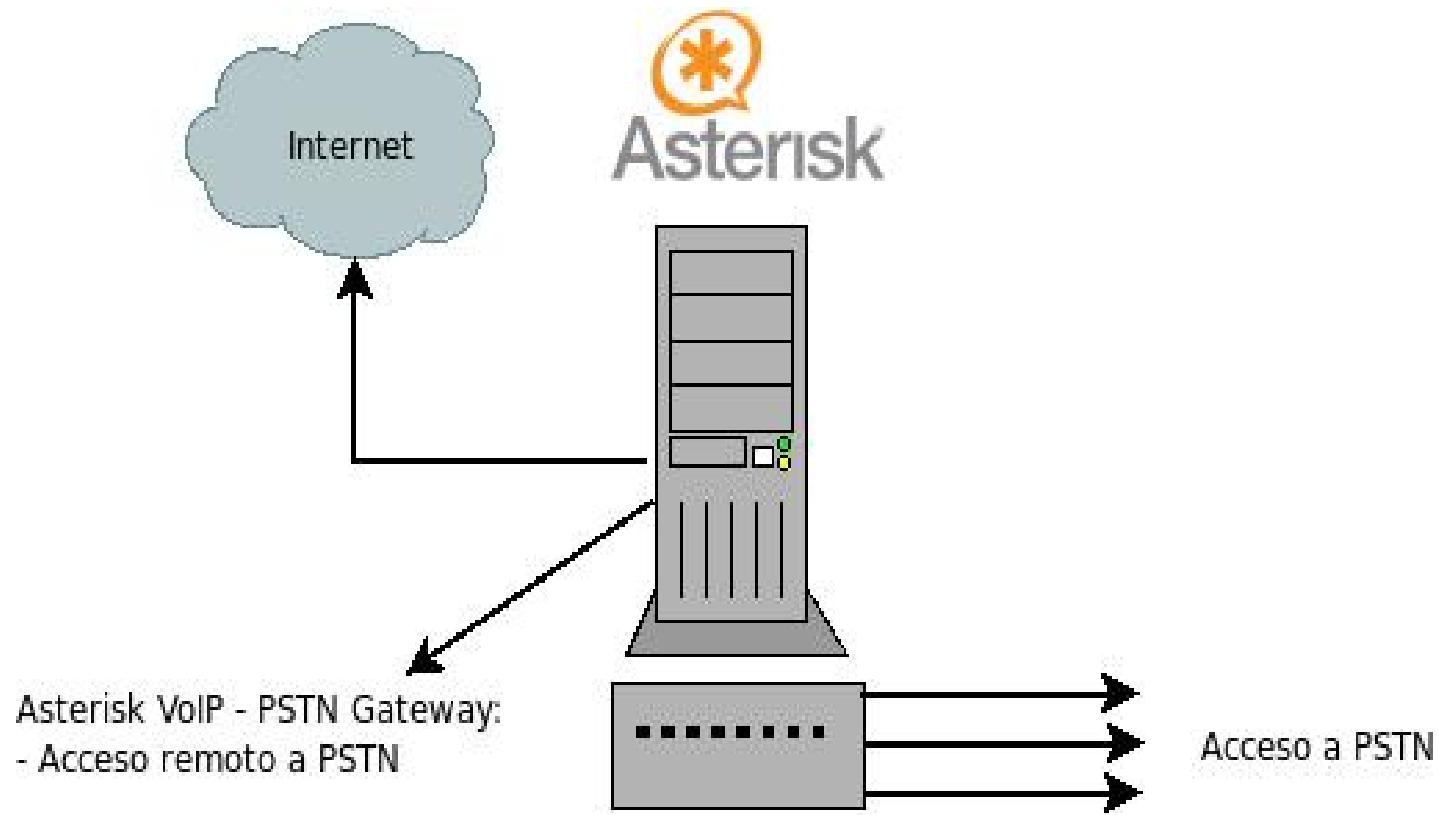
# Complemento/ampliación de una PBX tradicional





# Gateway VoIP - PSTN

---





# Recursos para Asterisk



# Para iniciarse

---

- Este curso! :P
- <http://www.voip-info.org> (Todo sobre Asterisk y VoIP)
- Libro: “Asterisk: The Future of Telephony”-> libre!!
- San Google! :P





# Asterisk “fácil-version”

---

- Distribuciones con Asterisk:
  - AsteriskNow: Basada en rPath Linux, incluye Asterisk-GUI.
  - TrixBox (antes Asterisk@Home): Basado en CentOS, incluye FreePBX (en futuras versiones Asterisk-GUI).
- GUIs para Asterisk:
  - Asterisk-GUI: Desarrollada por Digium. Asterisk Appliance.
  - FreePBX: Desarrollada por terceros.



# Asterisk “fácil-version” (2)

```
Asterisk Console on 'txiki-itx' (pid 10765) - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
== Registered application 'QueueLog'
== Manager registered action Queues
== Manager registered action QueueStatus
== Manager registered action QueueSummary
== Manager registered action QueueAdd
== Manager registered action QueueRemove
== Manager registered action QueuePause
== Manager registered action QueueLog
== Registered custom function QUEUE_VARIABLES
== Registered custom function QUEUE_MEMBER_COUNT
== Registered custom function QUEUE_MEMBER_LIST
== Registered custom function QUEUE_WAITING_COUNT
app_queue.so => (True Call Queueing)
== Registered application 'Read'
app_read.so => (Read Variable Application)
== Registered application 'SetCallerPres'
app_setcallerid.so => (Set CallerID Presentation Application)
== Registered application 'TrySystem'
== Registered application 'System'
app_system.so => (Generic System() application)
== Registered application 'Log'
== Registered application 'Verbose'
app_verbose.so => (Send verbose output)
Asterisk Ready.
*CLI> █
```

## Console roolz!!



# Asterisk-GUI

**digium | Asterisk** System Configuration | About Digium | Report a Bug | Help

**Conference Bridge Extensions Configuration** Logout

**Home** | **Users** | **Conferencing** | **Voicemail** | **Call Queues** | **Service Providers** | **Calling Rules** | **Incoming Calls** | **Voice Menus** | **Record a Menu** | **Active Channels** | **System Info** | **Backup** | **Options**

**Conferencing**  
MeetMe conference bridging allow quick, ad-hoc conferences with or without security.

**Extensions:**

- 6000 -- Test User1
- 6001 -- Conference Bridge**
- 8500 -- Check Voicemail

Extension:   
PIN Code:   
Administrator PIN Code:   
Play hold music for first caller   
Enable caller menu   
Announce callers

*Configuration saved!*

**Advanced**

- Room Override:
- Record conference
- Quiet Mode
- Wait for marked user
- Set marked user

**Tooltips:**  
Move the mouse over to a field to see tooltips

Copyright ©2006-2007 Digium, Inc. Digium® and Asterisk® are registered trademarks of Digium, Inc. All Rights Reserved. Legal Information



# FreePBX

FreePBX administration - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://100.100.100.10/admin/config.php?display=extensions&tech=sp

freePBX™ • Setup • Tools • Reports • Panel • Recordings

Logged in: admin (logout) :: Setup

Module Admin	<h2>Add SIP Extension</h2> <p><b>Add Extension</b></p> <p>Extension Number: <input type="text" value="299"/></p> <p>Display Name: <input type="text" value="Test"/></p> <p><b>Extension Options</b></p> <p>Outbound CID: <input type="text"/></p> <p>Emergency CID: <input type="text"/></p> <p>Record Incoming: <input type="text" value="On Demand"/></p> <p>Record Outgoing: <input type="text" value="On Demand"/></p> <p><b>Device Options</b></p> <p>secret <input type="text" value="1111"/></p> <p>dtmfmode <input type="text" value="rfc2833"/></p> <p><b>Voicemail &amp; Directory:</b> <input type="text" value="Disabled"/></p> <p><input type="button" value="Submit"/></p>	<b>Add Extension</b>
Conferences		Demo <399>
On Hold Music		
Digital Receptionist		
<b>Extensions</b>		
Inbound Routes		
Outbound Routes		
Trunks		
General Settings		
Administrators		



# Terminología Básica



# Telefonía

---

- PSTN: Public Switched Telephony Network.
- FXS: Foreign eXchange Station.
  - Lo que hay en nuestra pared.
  - **Alimenta** dispositivos.
  - Utiliza señalización FXO.
- FXO: Foreign eXchange Office.
  - El conector de nuestro teléfono donde entra la línea.
  - Recibe alimentación.
  - Utiliza señalización FXS.



# Redes y protocolos

---

- LAN, WAN, ...
- Protocolos:
  - Capa de Transporte:
    - TCP
    - UDP
  - Capa de Aplicación:
    - SIP: Session Initiation Protocol
    - IAX2: Inter Asterisk eXchange



# Arquitectura de Asterisk





# Arquitectura de Asterisk

---

- Asterisk es muy modular.
- Nos permite ajustarlo a nuestras necesidades (escalabilidad).
  - Asterisk en dispositivos empotrados.



# Arquitectura de Asterisk (2)

---

Canal SIP

Canal IAX2

Canal ZAP

## Canales

CSV

MySQL

## CDR



## PBX Core

Dial

Voicemail

## Aplicaciones y funciones

AGI

config

## Recursos

.gsm

.wav

## Formatos

G.729

G.711

## Codecs



# Codecs de audio

---

- Sirven para digitalizar la voz humana.
- Cuando dos dispositivos van a establecer una comunicación, acuerdan el codec a utilizar.
- Factores importantes a la hora de seleccionar un codec:
  - Calidad de audio.
  - Coste computacional.
  - Gasto de ancho de banda.
- Comparación de ancho de banda utilizado por los codecs. Fuente: [Cisco](#)



# Codecs de audio (2)

---

<b>Codec &amp; Bit Rate (Kbps)</b>	<b>Bandwidth Ethernet (Kbps)</b>
G.711 (64 Kbps)	87.2 Kbps
G.729 (8 Kbps)	31.2 Kbps
G.723.1 (6.3 Kbps)	21.9 Kbps
G.723.1 (5.3 Kbps)	20.8 Kbps
G.726 (32 Kbps)	55.2 Kbps
G.726 (24 Kbps)	47.2 Kbps
G.728 (16 Kbps)	31.5 Kbps



# Codecs de audio (3)

---

- Los proveedores VoIP suelen trabajar en g729, que consume muy poco ancho de banda y no sería viable utilizar g711 por ejemplo.
- PERO **g729 no es libre** :(
- Cada licencia de g729 para un canal = 10\$
- Hay una implementación que puede utilizarse con fines no lucrativos/educacionales  
<http://asterisk.hosting.lv/>
- Para comprobar los costes del transcoding:  
\*CLI>core show translation



# Codecs de vídeo

---

- Asterisk soporta los siguientes codecs de vídeo en modo passthrough:
  - H.261
  - H.263
  - H.264
- Solo son soportados en modo passthrough, por lo que no es posible hacer transcoding.
- Disponen de su formato asociado, por lo que se pueden usar para IVRs visuales o video voicemail.



# Formatos

---

- Asterisk soporta multitud de formatos: wav, mp3, gsm, h.264, etc ...
- Los necesitamos para reproducir ficheros, p.e. en un IVR, Playback, Voicemail, ...

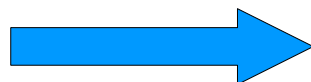


# Llamadas en Asterisk

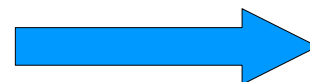
---



Dispositivo que origina la llamada: SIP, ZAP, IAX2, ...



Asterisk recoge la llamada y la procesa **su lógica de enrutado (dialplan)**. Después se encamina al destino.



Receptor de la llamada.

**En una llamada hay 2 canales:** el que origina la llamada y el que la recibe.





# AstChannels: La magia

---

- Si asterisk es multiprotocolo, ¿como se realiza una llamada entre 2 dispositivos de distinta tecnología?
- **Hay 4 canales!!** 2 dependientes de la tecnología utilizada y 2 AstChannels.
- Asterisk crea los AstChannels para poder llevar a cabo el bridging.
  - Transferencias.
  - Parking.



# Bridging

---



Canal **dependiente**  
de la tecnología. SIP,  
ZAP, IAX2,...



Canales  
**independientes** de  
la tecnología.  
Asterisk los genera y  
lleva a cabo el  
mecanismo de  
“bridging”.



Canal **dependiente**  
de la tecnología. SIP,  
ZAP, IAX2,...



# Instalando Asterisk



# Antes de instalar...

---

- Factores a tener en cuenta antes de instalar Asterisk:
  - Número de llamadas simultáneas.
  - Interrupciones.
  - Transcoding.
- Necesitamos que el servidor este **completamente dedicado** a Asterisk
  - Nada de X.
  - Ningún servicio que no sea estrictamente necesario.



# Librerías necesarias

---

- Para poder compilar Asterisk tendremos que instalar lo siguiente **como mínimo**:

```
#apt-get install bison libncurses5-dev  
libnewt-dev zlib1g-dev libssl-dev initrd-  
tools procps linux-headers-$(uname -r)  
build-essential
```

- Algunos módulos requieren librerías especiales:
  - El módulo chan\_gtalk necesita la librería iksemel.



# Compilando Asterisk

---

- Primero descargamos los ficheros necesarios:

```
#mkdir asterisk
```

```
#cd asterisk
```

```
#wget ftp://ftp.digium.com/pub/zaptel/zaptel-1.4.4.tar.gz
```

```
#wget ftp://ftp.digium.com/pub/libpri/libpri-1.4.1.tar.gz
```

```
#wget ftp://ftp.digium.com/pub/asterisk/asterisk-1.4.7.1.tar.gz
```

```
#wget ftp://ftp.digium.com/pub/asterisk/asterisk-addons-1.4.2.tar.gz
```



# Compilando Asterisk (2)

---

- Descomprimimos los ficheros y comenzamos a compilar:

```
#tar -zxvf zaptel-1.4.4.tar.gz
#cd zaptel-1.4.4
#./configure
#make menuselect
#make
#make install
```

```
#tar -zxvf libpri-1.4.1.tar.gz
#cd libpri-1.4.1.tar.gz
#make
#make install
```



# Compilando Asterisk (3)

---

```
#tar -zxvf asterisk-1.4.7.1.tar.gz
```

```
#cd asterisk-1.4.7.1
```

```
#./configure
```

```
#make menuselect
```

```
#make
```

```
#make install
```

```
#tar -zxvf asterisk-addons-1.4.2.tar.gz
```

```
#cd asterisk-addons-1.4.2.tar.gz
```

```
#./configure
```

```
#make menuselect
```

```
#make
```

```
#make install
```





# Estructura de directorios

---

- `/etc/asterisk`: Contiene los ficheros de configuración. Si al compilar ejecutamos “make samples”, tendremos ejemplos en este directorio.
- `/usr/lib/asterisk/modules`: Contiene los módulos de Asterisk que hemos compilado.
- `/var/lib/asterisk`: Contiene diferentes “librerías” de Asterisk.
- `/var/lib/asterisk/agi-bin`: Directorio para contener los AGI.
- `/var/spool/asterisk`: Directorio para archivos que genera Asterisk (voicemail, etc.)
- `/var/log/asterisk`: Aquí se guardan los log de Asterisk.



# Ejecutando Asterisk

---

- Hay que ser **root**.
- Modificadores de arranque:
  - -c: Modo consola. Asterisk se inicia en primer plano.
  - -v: Verbose. Cuantas más uves se añadan más mensajes descriptivos veremos en la consola.
  - -d: Debug. Cuantas más des, más mensajes de debug.
  - -r: Remote console. Si hemos ejecutado Asterisk en segundo plano, nos permite conectarnos al CLI.



## Ejecutando Asterisk (2)

---

- Para lanzar Asterisk en segundo plano:  
`#asterisk`
- Para lanzar Asterisk en primer plano:  
`#asterisk -vvvvvvvvvvc`
- Para conectarnos a la consola si Asterisk  
esta en segundo plano:  
`#asterisk -vvvvvvvvvvvr`



# El CLI

---

- El CLI (Command Line Interpreter) es la consola de Asterisk.
- Nos permite interactuar con Asterisk.
- Auto-completa los comandos pulsando TAB.
- Múltiples comandos nos resultan útiles para “ver lo que esta pasando”
  - \*CLI> sip show channels
  - \*CLI> core show application Dial



# Reload/Restart

---

- Ambos sirven para reiniciar Asterisk, pero...
- Reload:
  - No todos los módulos se reconfiguran con reload.
  - No corta las llamadas en curso.
- Restart:
  - Se cortan las llamadas en curso.
  - Reconfigura todos los módulos.



# Configuración de Asterisk



# Ejercicio 1

---

- Este primer ejercicio se configurarán 2 extensiones y un sencillo dialplan para poder realizar llamadas entre ellos.
- Receta:
  - Crear entradas en el fichero sip.conf
  - Crear un dialplan sencillo: extensions.conf
  - Configurar el SoftPhone X-Lite
  - Configurar el hardphone
  - Hacer un test de eco, un playback de demo y llamadas entre ellos.
  - Hacer un enlace con Digium:  
exten =>  
1234,1,Dial(IAX2/guest@misery.digium.com/s@default)



# Ejercicio 1: sip.conf

---

[general]

bindport=5060

bindaddr=0.0.0.0

context=default

realm=asterisk-saghul

disallow=all

allow=alaw

language=es





# Ejercicio 1: sip.conf (2)

---

```
[softphone]
type=friend
secret=1234
host=dynamic

[hardphone]
type=friend
secret=1234
host=dynamic
```



# Ejercicio 1: extensions.conf

---

[general]

language=es

[globals]

[default]

exten => 1234,1,Answer()

exten => 1234,n,Playback(beep)

exten => 1234,n,Echo()

exten => 12345,1,Answer()

exten => 12345,n,Playback(demo-congrats)

exten => 12345,n,Hangup

exten => 200,1,Dial(SIP/softphone,30)

exten => 201,1,Dial(SIP/hardphone,30)



# Ejercicio 1: X-Lite



```
*CLI> -- Registered SIP '200' at 192.168.1.25 port 5061 expires 1800
-- Saved useragent "X-Lite release 1105d" for peer 200
```



# Ejercicio 1: Fin

---

- Probad a hacer llamadas.
- Probad el test de eco.
- ¿Todo OK?
  - ¿Y si llamamos a un número que no existe?
- **iEres libre de investigar!**



# Más Información

---

- Los ficheros de ejemplo.
- <http://www.asterisk.org/doxygen/1.4/>
- <http://www.voip-info.org>



# El Dialplan: El corazón de Asterisk



# El Dialplan

---

- Como una “tabla de enrutado”.
- Cada número que marca una extensión va al dialplan, y ahí decidimos que hacer.
- Se organiza por contextos, extensiones y prioridades.

**exten => extensión, prioridad, aplicación**



# El Dialplan (2)

---

- La “**extensión**” puede ser cualquier cosa, no solo valen números:  
exten => saul,1,Dial(SIP/saghul)
- Se utilizan “patrones de marcado” para las extensiones (ino vamos a meter todos los números del mundo!)
  - \_ : Comienzo del patrón.
  - X: Cualquier dígito del 0 al 9.
  - Z: Cualquier dígito del 1 al 9.
  - N: Cualquier dígito del 2 al 9.
  - [12-4]: Cualquier dígito entre los corchetes, 1,2,3,4.
  - .(punto): Cualquier cosa.





# El Dialplan (3)

---

- En el ejercicio 1 no hemos utilizado patrones:  
exten => 200,1,Dial(SIP/softphone)
- Ejemplos de patrones:
  - Llamadas nacionales (empiezan por 8 o 9):  
\_[89]XXXXXXXX
  - Llamadas a móviles (empiezan por 6):  
\_6XXXXXXXX
  - Llamadas Internacionales:  
\_00X.
  - Llamadas a números especiales SIN números de pago (806, 803, 906, 903)  
\_[89]0[0-2457-9]XXXXXXXX



# El Dialplan (4)

---

- Las **prioridades** sirven para decidir **el orden de las acciones** al entrar en el dialplan.
- Pueden utilizarse números:  
exten => 1234,1,Answer()  
exten => 1234,2,Playback(demo-congrats)
- O la prioridad “n”, que va sumando 1 automáticamente:  
exten => 1234,1,Answer()  
exten => 1234,n,Playback(demo-congrats)
- Si utilizamos números es fácil saltar:  
exten => 123,1,Goto(1234,1)
- Si usamos la prioridad “n” podemos hacer “marcas” en el dialplan, para luego poder saltar a ellas:  
exten => 1234,n(mi-kosa),...
- Y podemos hacer:  
exten => 123,1,Goto(1234,mi-kosa)



# El Dialplan (5)

---

- Las extensiones se agrupan en **contextos**.
- Cada dispositivo SIP, IAX o entrada de ZAP se define para un contexto.
- Un dispositivo solo puede “llamar” a los números que tiene definidos en el contexto en el que esta.
- En el ejemplo 1, solo podíamos marcar los números definidos en el contexto “default”.
- Un contexto puede incluir a otro:  
[micontexto]  
include => default
- Ahora aunque mi dispositivo este definido en el contexto “micontexto” tiene acceso a todo el contexto “default”.



# El Dialplan (6)

---

- Hay “extensiones especiales”, que no son para que nadie marque:
  - s: Extensión start. Cuando entramos en un contexto SIN extensión. Macros, IVRs, entrantes zaptel.
  - h: Extensión hangup. Se llega a esta extensión al finalizar la llamada.
  - i: Extensión inválida. Opciones incorrectas en IVRs.
  - T: Timeout absoluto en Dial.
  - t: Timeout en IVRs
  - fax: Detección de fax en canales ZAP.



# Ejercicio 2

---

- Este ejercicio consiste en mejorar nuestro dialplan, añadiendo nuevas funciones y creando varios contextos:
- Receta:
  - Crear un contexto “internas” donde se indica como llamar a las extensiones internas.
  - Crear un contexto “servicios” con los servicios de eco y demo.
  - Crear el contexto “telf-internos” que incluya los 2 contextos anteriores y hacer que los dispositivos SIP utilicen este contexto.
  - Incluir el contexto “internas” en el contexto default, para que si permitieramos llamadas entrantes, solo pudieran llamar a los teléfonos, pero no usar los servicios.



# La aplicación Dial

---

- La aplicación Dial permite realizar una llamada a un dispositivo. Su formato más sencillo es:  
Dial(Tecnología/dispositivo,tiemout,opciones)
- Donde opciones puede ser:
  - t: Permitir que el usuario **llamado** transfiera la llamada.
  - T: Permitir que el **llamante** transfiera la llamada.
  - r: Generar un tono de ring artificial.
  - w: Permite al usuario llamado grabar la llamada.
  - W: Permite que el llamante grabe la llamada.



# La aplicación Dial (2)

---

- Tras un Dial correcto (se contesta la llamada), se termina la ejecución de acciones en el dialplan y se pasa a la extensión h (si existe):

exten => 1234,1,Dial(SIP/saghul)

exten => 1234,n,NoOp(Algo ha ido mal...)

- La variable DIALSTATUS contiene información acerca del último Dial realizado. Puede tomar los siguientes valores, entre otros: BUSY, NOANSWER, CHANUNAVAIL, ...
- Nos puede servir para hacer distintas cosas dependiendo del estado de la última llamada.



# La aplicación Dial (3)

---

Más información sobre Dial:

- `core show application dial` (en el CLI de Asterisk)
- <http://www.voip-info.org/wiki-Asterisk+cmd+Dial>
- <http://www.voip-info.org/wiki/view/Asterisk+variable+DIALSTATUS>





# La aplicación Goto

---

- Permite realizar saltos dentro del dialplan.
- Se pueden realizar saltos en base a:
  - Prioridad o etiqueta.
  - Extensión y prioridad (o etiqueta).
  - Contexto, extensión y prioridad (o etiqueta).
- Por ejemplo:  
exten => 1234,1,Goto(servicios,1234,1)



# Ejercicio 2 (2)

---

- Pistas:
  - El Dial carga la variable DIALSTATUS.
  - El Goto hace saltos.
  - Necesitamos saltar dependiendo del estado de la última llamada.
- Observaciones:
  - Los números de extensión serán de la forma 2XX
  - Los servicios serán de la forma 4XX



# Ejercicio 2 (solución)

---

## **extensions.conf**

```
[default]  
include => internas
```

```
[servicios]  
exten => 400,1,Answer()  
exten => 400,n,Playback(beep)  
exten => 400,n,Echo()  
exten => 401,1,Answer()  
exten => 401,n,Playback(demo-congrats)  
exten => 401,n,Hangup
```

```
[internas]  
exten => 200,1,Dial(SIP/softphone,45,Tt)  
exten => 200,n,Goto(200- $\${DIALSTATUS}$ ,1)  
exten => 200-BUSY,1,Busy(5)  
exten => _200-.,1,Congestion(5)
```

```
exten => 201,1,Dial(SIP/hardphone,45,Tt)  
exten => 201,n,Goto(201- $\${DIALSTATUS}$ ,1)  
exten => 201-BUSY,1,Busy(5)  
exten => _201-.,1,Congestion(5)
```

```
[telf-internos]  
include => internas  
include => servicios
```



# Ejercicio 2 (solución) (2)

---

**sip.conf**

```
[general]  
context=default
```

```
[softphone]  
type=friend  
secret=1234  
context=telf-internos  
host=dynamic
```

```
[hardphone]  
type=friend  
secret=1234  
context=telf-internos  
host=dynamic
```



# Aplicaciones

---

- Ya hemos visto algunas: Dial, Goto, ...
- Se ejecutan directamente desde el dialplan.
- Para una lista completa de las aplicaciones disponibles:  
\*CLI>core show applications



# Variables

---

- Al igual que la variable DIALSTATUS, hay muchas variables en el entorno de Asterisk:
  - EXTEN: Contiene el número marcado.
  - CONTEXT: Contiene el contexto actual.
- Una lista completa de las variables:  
<http://www.voip-info.org/wiki-Asterisk+variables>
- Las variables se pueden “recortar”:
  - `${variable:desplazamiento:longitud}` por ejemplo:
  - `${variable}=saghul -> ${variable:1} => aghul`  
`${variable:0:2}=sa`
- Se soportan expresiones básicas, encerrándolas entre corchetes:  
`exten => 1234,1,Set(kosa=${6*5})`



# Funciones

---

- Se utilizan en el dialplan, pero dentro de las aplicaciones, sirven para trabajar con datos:
  - LEN: Devuelve la longitud de una cadena  
Noop(LEN(kosa)) //devuelve 4
  - CALLERID: Fija o lee el valor del callerid:  
Set(numero=CALLERID(number))
- Para obtener una lista completa de funciones podemos ejecutar:  
\*CLI>core show functions



# Transferencias y Call Parking





# Transferencias

---

- Para configurar las teclas a utilizar en las transferencias, editar el fichero `/etc/asterisk/features.conf`
  - `blindxfer => *1` //transferencia ciega
  - `atxfer => *2` //transferencia atendida



# Call Parking

---

- Podemos configurar un “parking” para las llamadas.
- Transferimos una llamada (ciegamente) a la extensión de parking, y escucharemos una alocución diciéndonos donde está aparcada la llamada.
- Mientras la llamada esta aparcada, el usuario escuchará música.
- Si llamamos (desde cualquier teléfono) a la extensión que nos han anunciado, recuperamos la llamada.

[general]

parkext => 700

parkpos => 701-720

context => parkedcalls

parkingtime => 240



# Dialpan: Macros e IVRs



# Macros

---

- Son contextos con un comportamiento distinto.
- Admiten parámetros.
- Empiezan **siempre** con la extensión “s”.
- Son como subprogramas, se utilizan para no repetir código en el dialplan.
- Ejemplo de macro que muestra en el CLI el parámetro que se le pasa:

```
[macro-prueba]
exten => s,1,Answer()
exten => s,n,Playback(demo-congrats)
exten => s,n,Noop(${ARG1})
exten => s,n,Hangup
```

- Para llamar a la macro:  
exten => 1234,1,Macro(prueba,\${variable})



# Ejercicio 3

---

- Implementar la funcionalidad de los saltos descrita en el ejercicio 2 utilizando una macro en lugar de repetir el código.
- Ya de paso, si esta ocupado le mandaremos al buzón dando un mensaje de ocupado y si no responde le mandaremos al buzón dando un mensaje de no disponible.
- Si hay algún error (ni BUSY ni NOANSWER) le damos tonos de ocupado.



# Ejercicio 3 (pistas)

---

- <http://www.voip-info.org/tiki-index.php?page=Asterisk+cmd+VoiceMail>
- Mirar la aplicación Playtones.
- Acordaros del “pattern matching”.



# Ejercicio 3 (solución)

---

[macro-prueba]

exten => s,1,Dial(\${ARG1},30,Tt)

exten => s,n,Goto(s-\${DIALSTATUS},1)

exten => s-BUSY,1,Voicemail(\${MACRO\_EXTEN}|b)

exten => s-BUSY,n,Hangup

exten => s-NOANSWER,1,Voicemail(\${MACRO\_EXTEN}|u)

exten => s-NOANSWER,n,Hangup

exten => \_s-.,1,Answer()

exten => \_s-.,n,Playtones(busy)

exten => \_s-.,n,Busy(5)



# IVRs

---

- Un IVR (Interactive Voice Response) es un menú con el que el usuario puede interactuar mediante pulsaciones DTMF.
- Comienza con la extensión “s”.
- Se comprueba la hora para decidir la siguiente acción.
- Se reproduce un mensaje de bienvenida y se esperan las pulsaciones del usuario.
- Se controlan las extensiones “t” e “i”, por si el usuario no pulsa nada en el tiempo fijado o realiza una pulsación incorrecta.





## IVRs (2)

---

- Para que no se entre en un bucle infinito, se suelen fijar 2 tipos de retardo: tiempo inter-digito y el tiempo de respuesta.

Set(TIMEOUT(digit)=3)

Set(TIMEOUT(response)=9)



# GotolfTime

---

- Esta aplicación realiza un salto condicional, dependiendo de la hora.
- Resulta muy útil en IVRs, para determinar si nos encontramos en horario de oficina o no, por ejemplo.
- Sintaxis:  
`GotolfTime(hora|dias_semana|dias_mes|año?si_cierto:si_falso)`
- Ejemplos:  
`GotolfTime(09:00-19:30|mon-fri|*|*?7:8)`  
`GotolfTime(09:00-19:30|mon-fri|*|*?dentro_horario)`



# Background

---

- Esta aplicación es muy similar a Playback, pero puede ser interrumpida si el usuario marca un dígito.
- Muchas veces nos sabemos el IVR de la oficina de memoria y no queremos escucharlo entero. xD
- Sintaxis  
Background(nombre-del-fichero)



# WaitExten

---

- Espera a que el usuario introduzca un número marcándolo en su teléfono.
- ¡En un IVR tenemos que esperar a que el usuario introduzca la opción deseada!
- Sintaxis:  
`WaitExten(10) //espera 10 segundos.`



# IVRs: Ejercicio

---

- Un IVR es algo que puede ser bastante complejo.
- Abarca muchos conceptos.
- Antes de realizar un IVR completo, necesitamos algunos conceptos más:
  - Sonidos en Asterisk.
  - **Colas y agentes.**



# Sonidos en Asterisk



# Localización

---

- Para ajustar la localización de Asterisk a nuestro idioma debemos realizar las siguientes modificaciones:
  - Fijar “country” a “es” en el fichero `indications.conf`
  - Poner “language=es” en la sección “general” de todos los ficheros (`sip.conf`, `extensions.conf`)
  - Añadir lo siguiente al fichero `asterisk.conf`:  
[options]  
languageprefix = yes



# Sonidos personalizados

---

- Si queremos grabar nuestros propios sonidos, para utilizarlos en el IVR, por ejemplo, podemos grabarlos con cualquier aplicación y convertirlos al formato utilizado.
- Pero Asterisk dispone de la **aplicación Record**, que nos permite grabar un fichero en el formato que queramos.





# Record

---

- Permite grabar un fichero de sonido.
- Sintaxis:  
Record(nombre.formato)
- Podemos utilizar %d en el nombre del fichero, así se utilizará un índice auto-incremental, para no sobrescribir grabaciones. El nombre del fichero queda guardado en la variable `#{RECORDED_FILE}`
- Ejemplo:  
Record(sonido-asterisk%d.alaw)



# Ejercicio 4

---

- Incluir un nuevo servicio para la grabación de mensajes personalizados y grabar unos cuantos mensajes para el IVR.



# Ejercicio 4 (solución)

---

[servicios]

exten => 402,1,Answer()

exten => 402,n,Wait(2)

exten => 402,n,Playback(beep)

exten => 402,n,Record(asterisk-sonido%d.alaw)

exten => 402,n,Wait(2)

exten => 402,n,Playback(beep)

exten => 402,n,Playback(\${RECORDED\_FILE})

exten => 402,n,Hangup



# Colas y Agentes



# Colas

---

- Las colas nos permiten manejar de una manera cómoda y eficiente las llamadas entrantes.
- Las llamadas se distribuyen entre los agentes disponibles (que hayan iniciado sesión)
- Consisten básicamente en **miembros** que contestan las llamadas.



# Configuración de colas

---

## queues.conf

[general]

language=es

persistentmembers = yes

autofill = yes

//si reiniciamos...

//varias conexiones a la vez

[ventas]

musiconhold = default

strategy = ringall

//ringall-roundrobin-leastrecent

//fewestcalls-random-rrmemory

timeout = 15

retry = 5

//tiempo para reintentar

wrapuptime = 0

//tiempo para volver a llamarle

maxlen = 0

announce-holdtime = no

periodic-announce = queue-periodic-announce

periodic-announce-frequency=20

member => SIP/201

member => Agent/@1



# Aplicación Queue

---

- Muy similar a la aplicación Dial.
- Toma parámetros parecidos, pero en lugar del dispositivo se especifica el nombre de la cola.
- Ejemplo:
  - exten => 1234,n,Queue(ventas|t||45|)



# Agentes

---

- Los agentes son “extensiones virtuales”.
- En realidad son extensiones normales que automáticamente o tras un proceso de autenticación se convierten en agentes y pasan a atender las llamadas de una o varias colas.





# Configuración de agentes

---

## agents.conf

[agents]

autologoff= 15

wrapuptime = 0

musiconhold = default

//tiempo para volver a llamarle

group = 1

agent => 3001,1234,Comercial 1

agent => 3002,1234,Comercial 2



# Login de Agentes

---

- Cualquier extensión puede ser agente.
- Basta con que inicie sesión con un número de agente y contraseña válidos y automáticamente comenzará a recibir llamadas.
- En lugar de utilizarse agentes, utilizaremos **“miembros dinámicos”**.
- Ejemplo:
  - exten => 1234,1,AddQueueMember(ventas)
  - exten => 1235,1,RemoveQueueMember(ventas)



# Música en espera

---

- Mientras los usuarios esperan a ser atendidos escuchan música.
- Se configura con el fichero musiconhold.conf
- Es conveniente utilizar formato nativo.
  - mp3 carga mucho la CPU!!



# Música en espera (2)

---

```
[default]
mode=files
directory=/var/lib/asterisk/moh-native
random=yes
```



# Buzones de Voz



# Buzones de voz

---

- Muy útiles, son una mejora sustancial frente a las PBX tradicionales.
- Número “ilimitado” de buzones.
- Posibilidad de enviar los mensajes por e-mail ...



# Configuración de voicemail

---

## **voicemail.conf**

[general]

format=wav

maxmsg=100

//máximo número de mensajes

maxmessage=180

skipms=3000

//al rebobinar

maxsilence=10

maxlogins=3

[default]

200 => 1234,Pepe Perez

201 => 1234,Antonio Monje



# Aplicaciones

---

- Voicemail:
  - Sirve para enviar la llamada a un buzón.
  - Los modificadores u y b permiten modificar la alocución que escuchará el usuario.
- VoicemailMain:
  - IVR para la configuración del buzón de voz.





# Dialplan Avanzado



# Dialplan Avanzado

---

- La base de datos interna de Asterisk: AstDB
- Macros:
  - Do Not Disturb.
  - Call Forward.
- IVR:
  - Un IVR de oficina.



# AstDB

---

- Asterisk incorpora una BD tipo Berkley DB v2.
- En ella el propio Asterisk guarda diversos valores:
  - Registros SIP, IAX.
  - Información sobre colas.
- Nosotros también podemos guardar información en ella y consultarla desde el dialplan.
- Se organiza en familias, y dentro de una familia puedes tener diversas claves, y para cada clave un solo valor.



# AstDB (2)

---

- Ejemplo desde el CLI:

```
*CLI> database put mifamilia miclave mivalor  
Updated database successfully
```

```
*CLI> database show mifamilia  
/mifamilia/miclave           : mivalor
```

```
*CLI> database put mifamilia miclave mivalor2  
Updated database successfully
```

```
*CLI> database show mifamilia  
/mifamilia/miclave           : mivalor2
```



# AstDB (3)

---

- Desde el dialplan:
  - Set(var=\${DB(familia/clave)})
  - Set(DB(familia/clave)=\${var})



# Ejercicio 5

---

- Desarrollar una macro que implemente DND (Do Not Disturb)
- Si la extensión a la que llamamos tiene el DND activado reproducimos un mensaje de “extensión no disponible”.
- Si no lo tiene activado, le llamamos.
- Si no contesta o esta ocupado, le mandamos al buzón.
- Pistas:
  - Función DB\_EXISTS.
  - Aplicación Gotolf.
  - Variable MACRO\_EXTEN.



# Ejercicio 5 (solución)

---

[macro-llamar]

exten => s,1,Gotof(\$[\${DB\_EXISTS(DND/\${MACRO\_EXTEN})} = 0]?s,llamar)

exten => s,n,Playback(vm-extension)

exten => s,n,Playback(vm-isunavail)

exten => s,n,Hangup

exten => s,n(llamar),Dial(\${ARG1},45,Tt)

exten => s,n,Goto(s-\${DIALSTATUS},1)

exten => s-BUSY,1,Voicemail(\${MACRO\_EXTEN}|b)

exten => s-BUSY,n,Hangup

exten => s-NOANSWER,1,Voicemail(\${MACRO\_EXTEN}|u)

exten => s-NOANSWER,n,Hangup

exten => \_s-.,1,Answer()

exten => \_s-.,n,Playtones(busy)

exten => \_s-.,n,Busy(5)



# Ejercicio 6

---

- Ampliar la macro anterior para que implemente CF (Call Forward)
- Primero se comprueba el DND, y luego el CF.
- El desvío puede realizarse a cualquier número que los teléfonos internos puedan marcar (contexto telf-internos)
- Pistas:
  - Función DB\_EXISTS.
  - Variable DB\_RESULT.





# Ejercicio 6 (solución)

---

```
[macro-llamar]
exten => s,1,Gotolf([$${DB_EXISTS(DND/${MACRO_EXTEN})} = 0]?s,comp-cf)
exten => s,n,Playback(vm-extension)
exten => s,n,Playback(vm-isunavail)
exten => s,n,Hangup
exten => s,n(comp-cf),Gotolf([$${DB_EXISTS(CF/${MACRO_EXTEN})} = 0]?s,llamar)
exten => s,n,Goto(telf-internos,${DB_RESULT},1)
exten => s,n(llamar),Dial(${ARG1},45,Tt)
exten => s,n,Goto(s-${DIALSTATUS},1)
exten => s-BUSY,1,Voicemail(${MACRO_EXTEN}|b)
exten => s-BUSY,n,Hangup
exten => s-NOANSWER,1,Voicemail(${MACRO_EXTEN}|u)
exten => s-NOANSWER,n,Hangup
exten => _s-. ,1,Answer()
exten => _s-. ,n,Playtones(busy)
exten => _s-. ,n,Busy(5)
```



# Ejercicio 7

---

- Crear un IVR tipo oficina para vuestra centralita.
- Características:
  - Horarios.
  - Colas y miembros dinámicos.
  - Mensaje de bienvenida personalizado.
  - Buzones de voz.
  - Control de extensiones “i” y “t”.
  - Directorio de empresa.
  - Implementar 3 colas: operadora, comercial e i+d.
    - Pulse 1 para departamento comercial.
    - Pulse 2 para departamento i+d.
    - Sino espere y será atendido por una operadora ...



# Aplicación Directory

---

- Aplicación de directorio de empresa.
- Se solicita al usuario que introduzca por teclado las 3 primeras letras del nombre del empleado.
- A continuación se leen las opciones encontradas.
- Ejemplo:  
exten => 1234,1,Directory(default|internas|ef)
  - default: El contexto de voicemail.
  - internas: El contexto donde se marcará.
  - ef: Opciones para que lea el nombre (en lugar del apellido) y también la extensión.



# Ejercicio 7 (solución)

---

```
[ivr]
exten => s,1,GotolfTime(09:00-19:30|mon-fri|*|*?s,dentro)
;;fuera de horario
exten => s,n,Playback(fuera-de-horario)
exten => s,n,Voicemail(200|u)
exten => s,n,Hangup

;;dentro del horario
exten => s,n(dentro),Set(TIMEOUT(digit)=3)
exten => s,n,Set(TIMEOUT(response)=9)
exten => s,n,Answer
exten => s,n,Background(dentro-de-horario)
exten => s,n,WaitExten(2)
exten => s,n(cola),Queue(operadora|t||45|)
exten => s,n,Hangup

exten => i,1,Goto(s,cola)
exten => t,1,Goto(s,1)

exten => 1,1,Queue(comercial|t||45|)
exten => 1,n,Hangup
exten => 2,1,Queue(imasd|t||45|)
exten => 2,n,Hangup
```



# VoIP y Asterisk



# VoIP y Asterisk

---

- Ya hemos comentado que **Asterisk es una PBX Multiprotocolo Open Source.**
- Protocolos que Asterisk maneja:
  - SIP
  - IAX2
  - MGCP
  - SCCP
  - H.323
- Vamos a ver con más detalle la VoIP en general, el protocolo SIP, IAX2 y los canales analógicos/digitales.



# VoIP en general

---

## VoIP: Características Principales

- Se utiliza una única red. Si dos empresas están unidas a través de Internet, ¿Porqué no aprovecharlo ?
- Se administra una única red.
- **Estándares abiertos e internacionales.** Inter-operabilidad, Bajada de precios en proveedores y fabricantes de hardware VoIP.
- **Calidad:** Es posible conseguir la misma calidad, de hecho hoy por hoy, el 40% de las llamadas de las grandes operadoras se encaminan por VoIP.
- **Fiabilidad:** En LAN, se puede lograr una gran fiabilidad. En Internet también, pero existen quizás demasiados factores. (QoS, NAT)
- **Coste:** Instalación sobre red existente, precios de llamadas.



# VoIP en general (2)

---

## VoIP: Problemas

- **NAT:** El cáncer de la VoIP. Distintos tipos, no es fácilmente manejable.
- **QoS:** Necesidad de ofrecer calidad de servicio, al ser la voz crítica en tiempo real.
- **Latencia:** Tiempo que tarda la voz en llegar al destino.
- **Jitter:** Variación de la latencia.
- **Ancho de banda:** En España conexiones asimétricas. El ancho de banda es muy caro.





# Canales VoIP en Asterisk

---

- Antes de profundizar en el protocolo SIP o IAX2, definiremos unos conceptos comunes que maneja Asterisk.
- Peers, users y friends:
  - Asterisk HACE una llamada A UN “peer”.
  - Asterisk RECIBE una llamada DE un “user”.
  - Un “friend” es ambas cosas.
  - Concepto de “user” en desuso. Se utilizan “peers” y “friends”.



# Canales VoIP en Asterisk (2)

---

- Proceso de registro:
  - Un dispositivo informa a Asterisk sobre su ubicación (dirección IP y puerto)
  - Solo es necesario registrarse con Asterisk si Asterisk no sabe donde esta el dispositivo (host=dynamic).
  - El registro expira, y se vuelve a informar de la ubicación.
- Autenticación:
  - Necesaria para realizar una llamada. (¿o no?)
  - Pueden realizarse llamadas sin registrarse pero no sin autenticarse.
  - El registro implica autenticación, para que Asterisk mande las llamadas a la IP que se le ha dicho.



# Canales VoIP en Asterisk (3)

---

- Puertos y protocolos:
  - En IAX2 todo viaja por el mismo puerto: 4569 UDP.
  - En SIP, la señalización por el puerto 5060 UDP y los streams de audio y video con el protocolo RTP por puertos aleatorios (rtp.conf) UDP.



# El protocolo SIP



# El protocolo SIP

---

- Protocolo de Inicio de Sesión (Session Initiation Protocol)
- Un protocolo de **señalización**.
- Legible por el ser humano, similar a HTTP.
- Es principalmente un protocolo de señalización de capa de aplicación para iniciación, modificación y terminación de sesiones de comunicación multimedia entre usuarios.
- Principales elementos implicados:
  - User Agent (Usuario)
  - Registrar y SIP Proxy
- El sector de la VoIP tiende globalmente hacia SIP.



# El protocolo SIP (2)

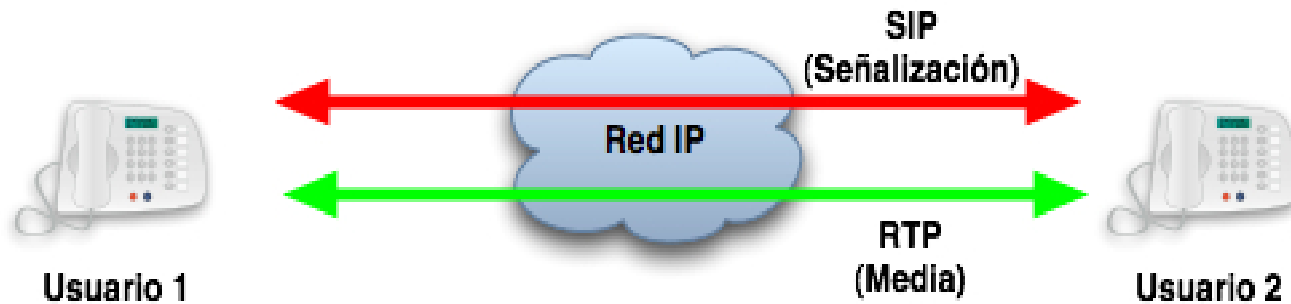
---

- Capacidades de SIP:
  - Localización del usuario.
  - Disponibilidad del usuario: determinación de la voluntad del receptor de la llamada de participar en las comunicaciones.
  - Capacidad del usuario: Determinación del medio y de sus parámetros.
  - Gestión de la sesión: transferencia, terminación de sesiones, modificación de los parámetros de la sesión desde el propio 'User Agent'.
- En SIP, el usuario es el 'dueño' de su sesión.
- Contrasta con el concepto de “centralita”.



# El protocolo SIP (3)

- El protocolo SIP es de forma nativa “peer to peer”: 2 User Agents pueden establecer una sesión entre sí:



- Dos Canales:
  - Señalización(UDP 5060): Establecimiento, Negociación,Fin...)
  - Streaming RTP (UDP 10000-20000 normalmente) y control RTCP.



# El protocolo SIP (4)

---

## **Proxy Server:**

- Aplicación intermedia que actúa tanto como servidor y cliente, generando mensajes SIP a nombre del cliente que generó el mensaje original.
- Los mensajes pueden ser respondidos o encaminados a otros servidores.
- Interpreta, re-escribe o traduce los mensajes antes de encaminarlos.
- Dos tipos de Proxy Server:
  - Outbound Proxy
  - Inbound Proxy

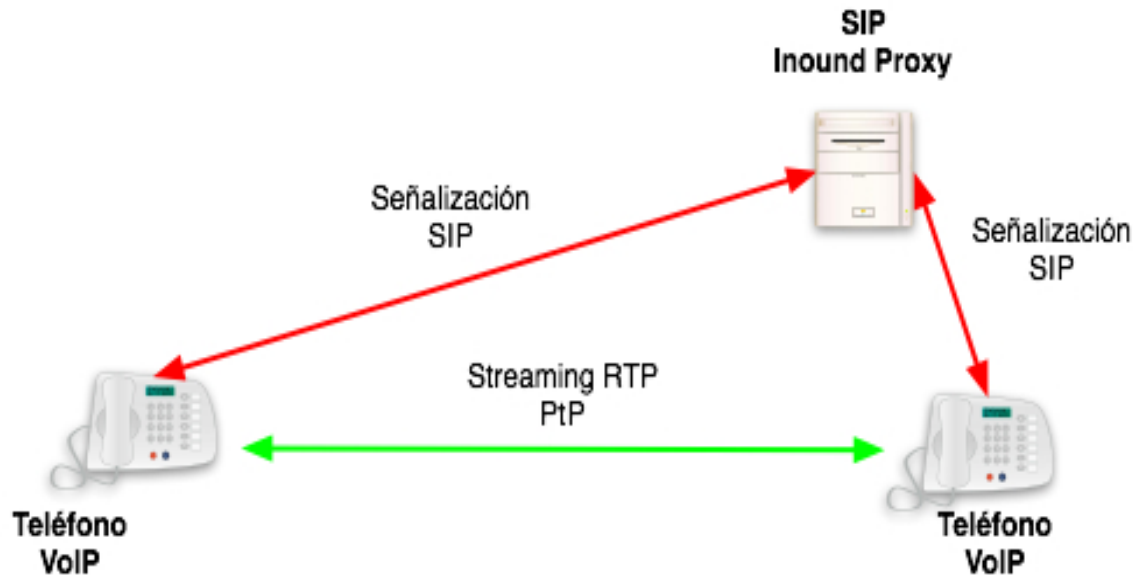




# El protocolo SIP (5)

## Inbound Proxy:

- El Inbound Proxy permite independizar al usuario del dispositivo que utiliza y de su localización:

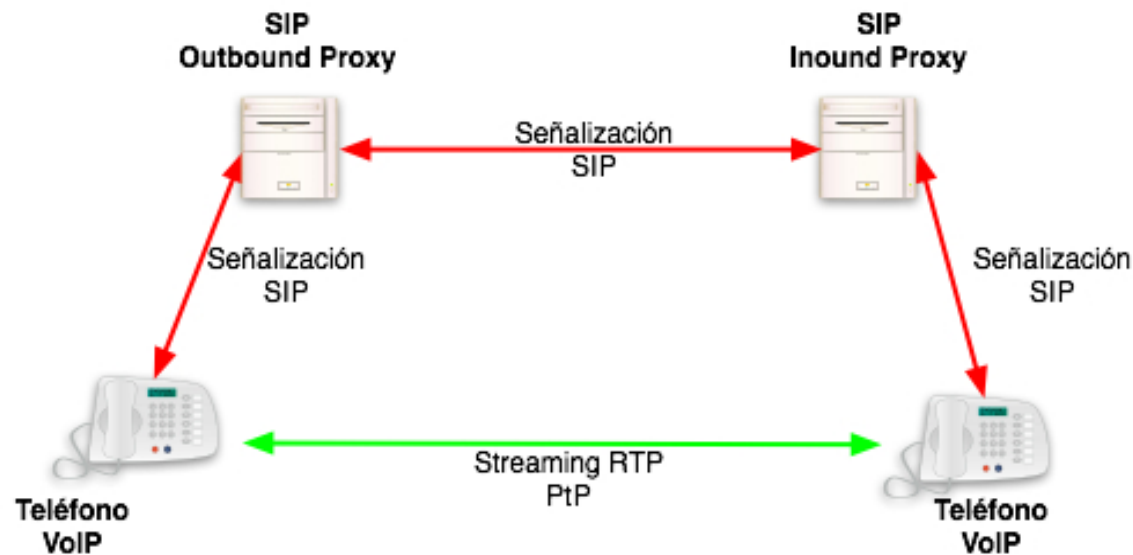




# El protocolo SIP (6)

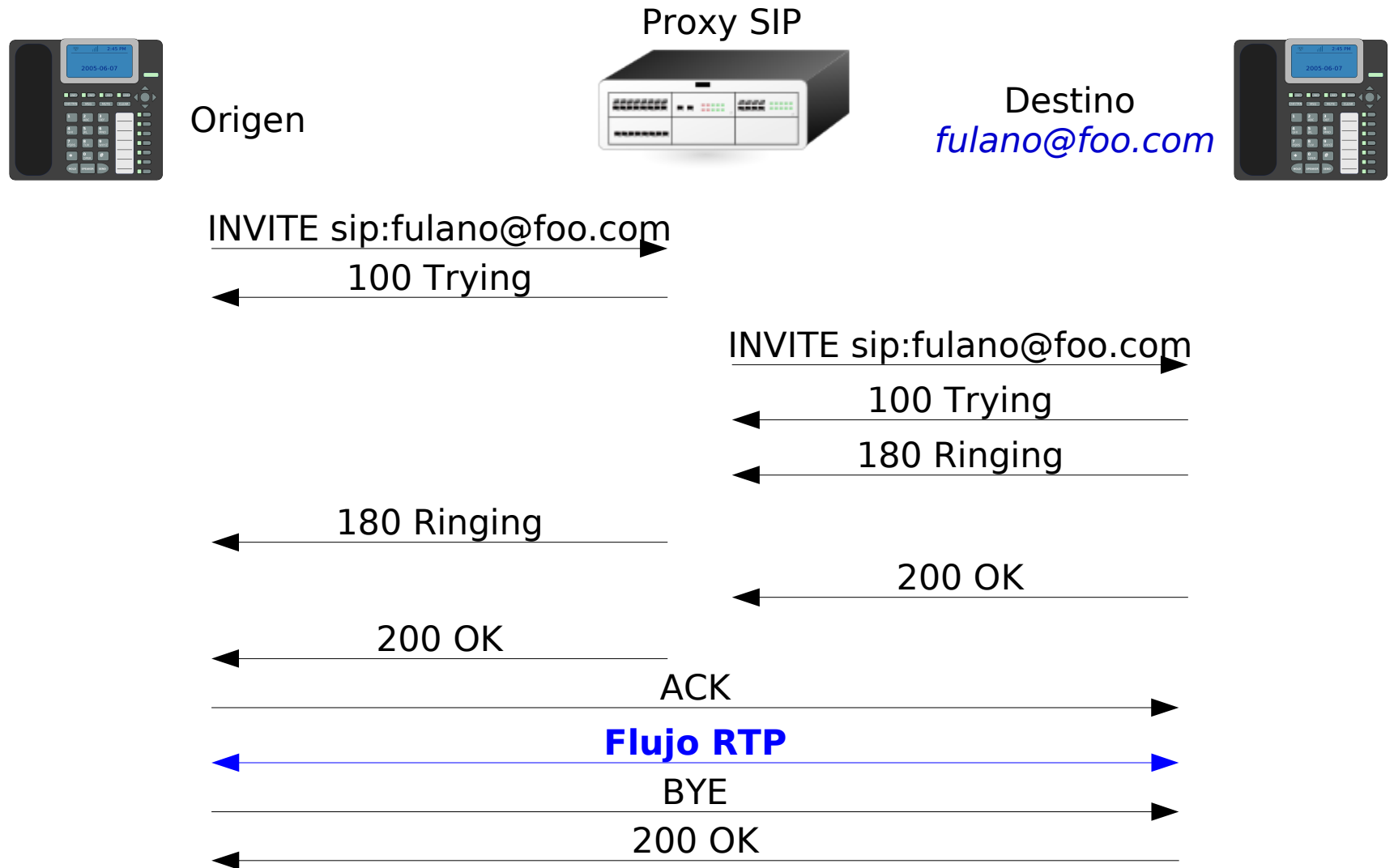
## Outbound Proxy:

- El proxy saliente simplifica la administración de los usuarios de un dominio, aplica políticas, tarifica, etc ...
- Un mismo servidor puede funcionar como proxy entrante y saliente de un dominio





# El protocolo SIP (7)

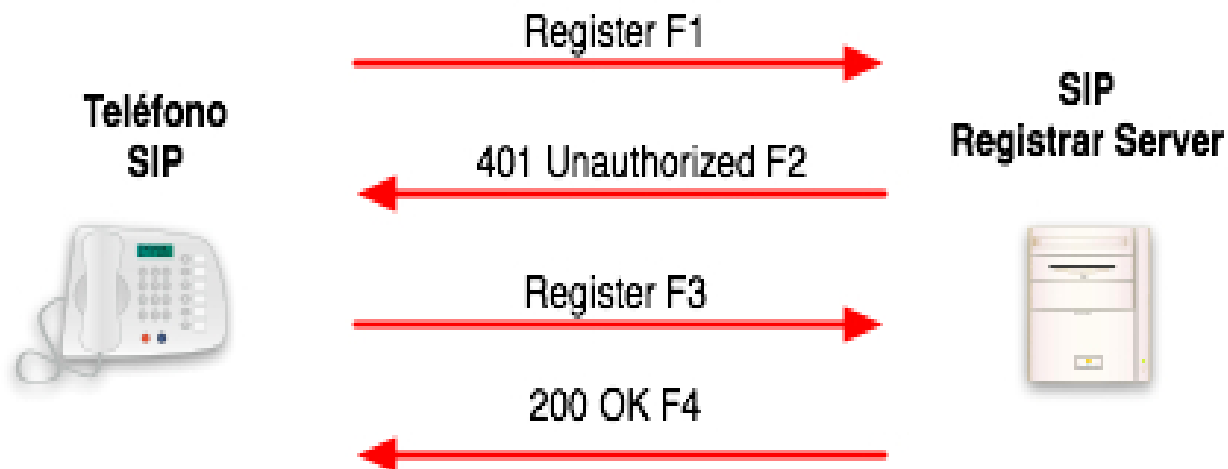




# El protocolo SIP (8)

## Registrar Server:

- Es un servidor que acepta mensajes de tipo REGISTER.
- De esa forma, se conoce la localización (IP + Puerto) de cada usuario (posibles múltiples localizaciones). No en Asterisk :(
- Normalmente: mismo servidor que el SIP Proxy.





# El protocolo SIP (9)

---

## **Movilidad:**

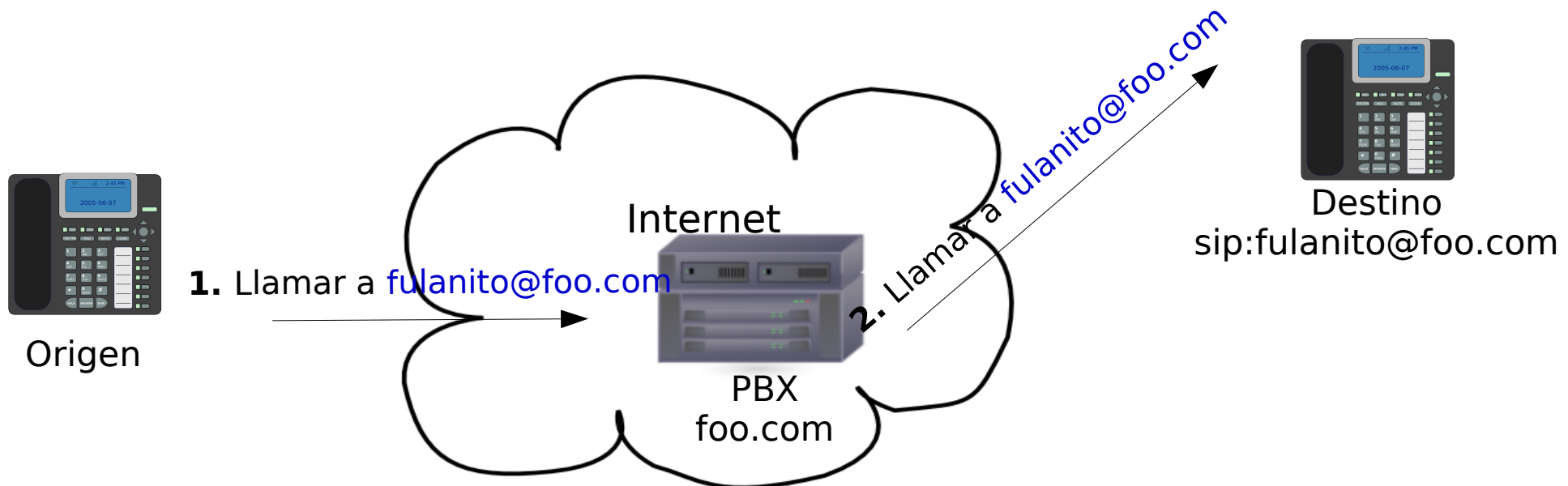
- SIP permite implementar dos tipos de movilidad diferentes:
  - La movilidad personal, el usuario puede ser alcanzado en un dispositivo cualquiera, registrándose en el SIP Registrar server.
  - La movilidad propia al protocolo IP (VPN).
- El registro permite mantener las localizaciones actuales del usuario de manera dinámica.
- Basado en la localización actual, el proxy server encaminará las llamadas al destino.



# El protocolo SIP (10)

## Direcciones SIP:

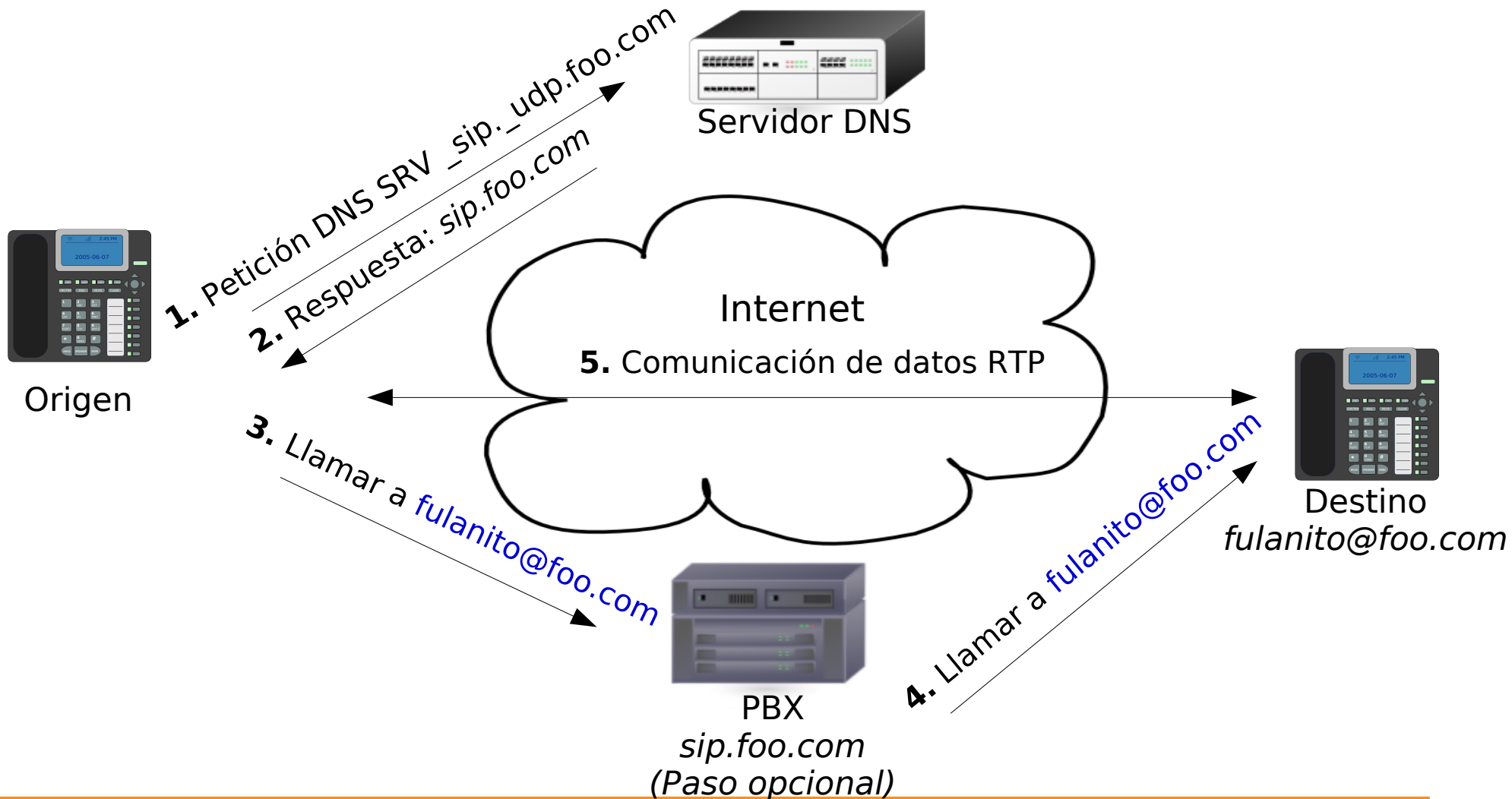
- Para llamadas entre teléfonos IP o softphones.
- Registros SRV en servidores DNS.
- Similar a las direcciones de correo: sip:fulanito@foo.com
- El cliente (teléfono SIP, softphone, PBX) se conecta a la centralita o punto destino responsable del dominio/IP llamado.





# El protocolo SIP (11)

## Registros SRV en servidores DNS:





# SIP vs. NAT

---

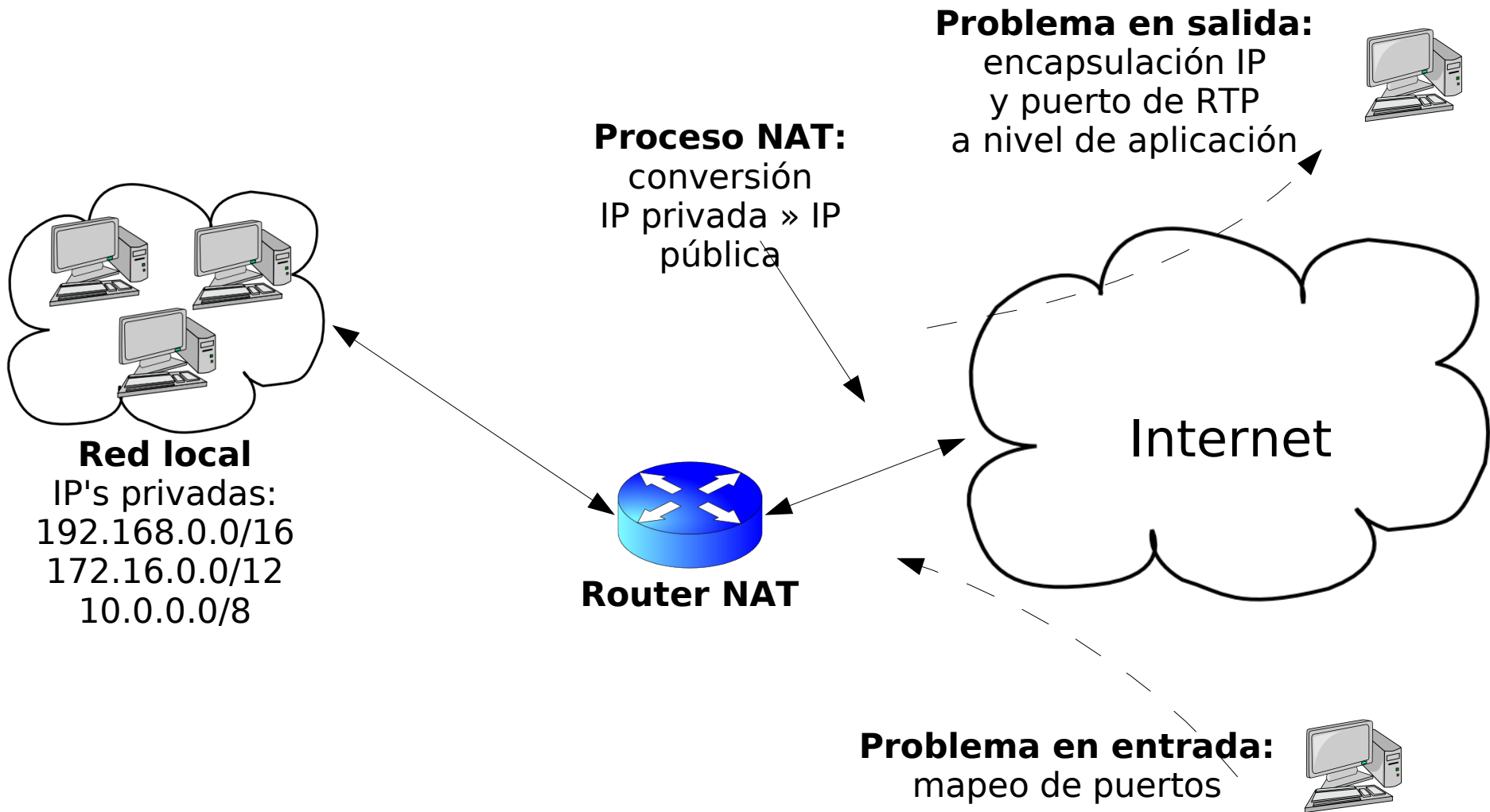
- El auténtico enemigo de SIP es el NAT.
- El “supuesto” agotamiento de los rangos de direcciones IP utilizables en Internet ha obligado a utilizar direcciones IP privadas dentro de las redes de las empresas y usuarios domésticos.
- Un equipo IP para ser alcanzado en Internet debe utilizar una IP pública para sus comunicaciones. Es necesario por tanto “enmascarar” la red interna en una o varias IPs públicas (Source NAT).
- El proceso de NAT no es nada sencillo: varios tipos de NAT, varios tipos de soluciones, **infinidad de problemas.**





# SIP vs. NAT (2)

## Escenario





# SIP vs. NAT (3)

---

- Problema difícil de solucionar.
- Soluciones en el servidor:
  - NAT Helpers.
  - Keep alive.
  - Media Proxys.
- Soluciones en el cliente:
  - Keep alive.
  - STUN.



# STUN

---

- Simple Traversal of UDP through NAT.
- Permite a clientes detrás de NAT averiguar su IP pública, tipo de NAT y puerto externo.
- NO soluciona el problema de NAT SIMÉTRICO!



# Asterisk chan\_sip

---

- Asterisk NO es un SIP Proxy.
- Si están activadas las transferencias, hay que hacer transcoding, ... el audio pasa a través de Asterisk.
- No implementa el RFC 3261 al 100%.
- chan\_sipv3???



# Asterisk chan\_sip (2)

---

```
[general]
bindport=5060
bindaddr=0.0.0.0
srvlookup=yes
language=es
canreinvite=yes
context=default
realm=asterisk-saghul
;pedantic=yes
disallow=all
allow=alaw
```

```
[telefono]
type=friend
context=internas
disallow=all
allow=alaw
dtmfmode=rfc2833
secret=1234
callerid="Saghul" <666>
host=dynamic
```



# chan\_sip y NAT

---

```
extenip=123.123.123.123  
localnet=192.168.1.0/24  
nat=no|yes|route|never  
qualify=no|yes|numero
```



# Enlaces SIP en Asterisk

---

- A parte de para registrar teléfonos podemos utilizar SIP para enlazar Asterisk con cualquier PBX, gateway, proxy, ... que hable SIP.
- Aunque para los enlaces Inter-Asterisk es mucho mejor IAX2, se pueden enlazar 2 Asterisk por SIP.
  - Configuración de un peer/user/friend.
  - Register
    - register => usuario:password@host



# Ejercicio 8

---

- Realiza un enlace por SIP con la PBX de tu compañero.
- Utiliza cuentas de “friend”.
- Haz un contexto para que las extensiones de una PBX puedan llamar a las de la otra.
- Inventa un prefijo para que al marcarlo la llamada sea enviada a la otra centralita.





# Ejercicio 8 (solución)

---

## En la centralita A:

```
[pbxB]
type=friend
username=pbxA
fromuser=pbxA
secret=1234
context=entrantes-sip
host=123.123.123.123
```

```
exten => _666X.,1,Dial(SIP/${EXTEN:3}@pbxB,45)
```

## En la centralita B:

```
[pbxA]
type=friend
username=pbxB
fromuser=pbxB
secret=1234
context=entrantes-sip
host=123.123.123.123
```

```
exten => _666X.,1,Dial(SIP/${EXTEN:3}@pbxA,45)
```

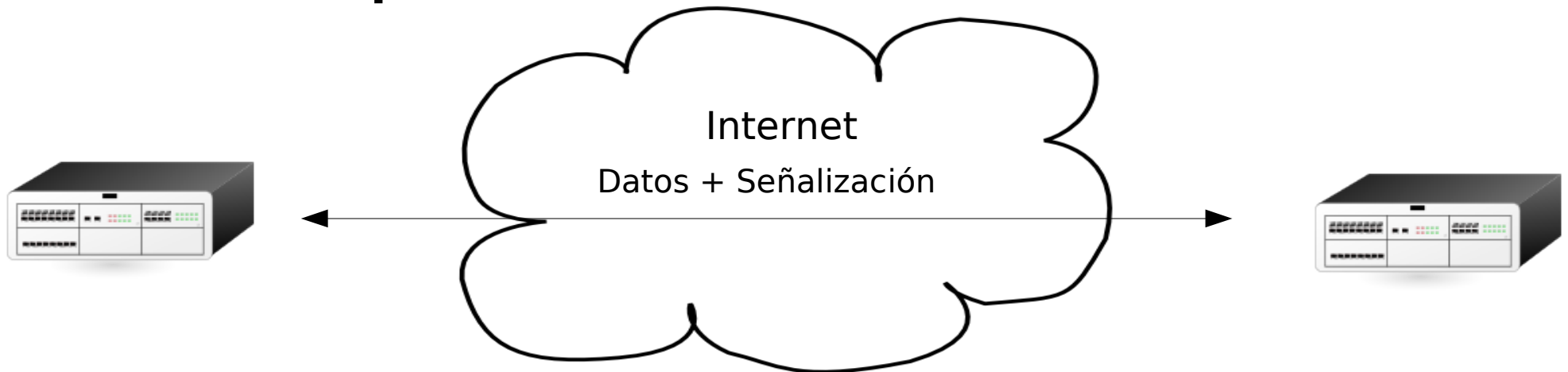


# El protocolo IAX2



# IAX2

- Inter Asterisk eXchange v2.
- Protocolo exclusivo para enlaces entre Asterisk, aunque hay algunos hardphones IAX2 (Atcom).
- Los streams de audio y la señalización viajan por el mismo puerto: 4569 UDP.
- Protocolo binario, NO basado en texto.
- **No tiene problemas con NAT!!**





# IAX2 (2)

---

- IAX2 trunking:
  - Es posible enviar varias conversaciones por el mismo flujo, lo cual supone un importante ahorro de ancho de banda (overhead de la capas IP y transporte UDP).
- Encriptación:
  - AES 128 ??
- Autenticación
  - MD5
  - Texto en claro (plaintext)
  - RSA
- Necesita una fuente de tiempo: Zaptel (ztdummy)



# Ejercicio 9

---

- Realiza un enlace por IAX2 con la PBX de tu compañero.
- Utiliza cuentas de “friend”.
- Haz un contexto para que las extensiones de una PBX puedan llamar a las de la otra.
- Inventa un prefijo para que al marcarlo la llamada sea enviada a la otra centralita.



# Ejercicio 9 (solución)

---

## En la centralita A:

```
[pbxB]  
type=friend  
username=pbxA  
secret=1234  
context=entrantes-iax  
host=123.123.123.123
```

```
exten => _666X.,1,Dial(IAX2/pbxA@pbxB/${EXTEN:3},45)
```

## En la centralita B:

```
[pbxA]  
type=friend  
username=pbxB  
secret=1234  
context=entrantes-iax  
host=123.123.123.123
```

```
exten => _666X.,1,Dial(IAX2/pbxB@pbxA/${EXTEN:3},45)
```



# Canales analógicos/digitales



# 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





# Telefonía Tradicional (2)

---

- 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 accesos básicos, Digium dispone de un hardware para 4 RDSI.



# Líneas Analógicas

---

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



# Tarjetas Analógicas

---



**TDM11B**



# Configuración TDM

---

## **/etc/zaptel.conf**

```
loadzone=es  
defaultzone=es  
fxoks=1  
fxsks=2
```

## **/etc/asterisk/zapata.conf**

```
[channels]  
language=es  
hidecallerid=no  
callwaiting=yes  
echocancel=yes  
echocancelwhenbridged=no  
echotraining=yes  
transfer=yes  
usecallerid=yes  
callerid=asreceived  
rxgain=0.0  
txgain=0.0  
busydetect=no  
busycount=5  
answeronpolarityswitch=yes  
hanguponpolarityswitch=yes  
signalling=fxs_ks  
context=entrada-zap  
group=1  
channel=2  
group=2  
signalling=fxo_ks  
context=telf-internos  
channel=1
```



# Líneas Digitales

---

- También utilizan el subsistema Zaptel.
  - La b410p de Digium de distinta manera.
- Los ficheros a configurar son los mismos.
- Las BRI utilizan mISDN en lugar de Zaptel. (Junghans no - BriStuff)



# Tarjetas PRI

---



**TE120P**



# Configuración PRI

---

## **/etc/zaptel.conf**

```
loadzone=es
defaultzone=es
span=1,1,0,ccs,hdb3,crc4
bchan=1-15
dchan=16
bchan=17-31
```

## **/etc/asterisk/zapata.conf**

```
[channels]
calwaiting=yes
transfer=yes
echocancel=yes
echocancelwhenbridged=no
language=es
switchtype=euroisdn
signalling=pri_cpe
usercallerid=yes
callerid=asreceived
rxgain=0.0
txgain=0.0

group=1
context=944123123
channel=1-15,17-31
```



# Tarjetas BRI

---



**B410P**





# Configuración BRI

---

- Para instalar una tarjeta B410P, hay que compilar Zaptel, mISDN y mIDSNUser.
- Al compilar Zaptel, en lugar de hacer make, ejecutamos:
  - make b410p
  - y luego como siempre: make install
- Compilamos mISDN:
  - wget <http://www.misdn.org/downloads/mISDN.tar.gz>
  - make
  - make install
- Compilamos mISDNUser
  - wget <http://www.misdn.org/downloads/mISDNuser.tar.gz>
  - make
  - make install



# Configuración BRI (2)

---

- Tras compilar Zaptel, mISDN y mISDNUser, hay que recompilar Asterisk.
  - Comprobar que esta habilitado **chan\_misdn**
- Ejecutamos: `/etc/init.d/misdn-init scan`

```
asterisk:/etc# /etc/init.d/misdn-init scan
[OK] found the following devices:
card=1,0x4
[ii] run "/etc/init.d/misdn-init config" to store this information to /etc/misdn-init.conf
```

- Ejecutamos: `/etc/init.d/misdn-init config` y se genera el fichero `/etc/misdn-init.conf`

```
asterisk:/etc# /etc/init.d/misdn-init config
[OK] /etc/misdn-init.conf created. It's now safe to run "/etc/init.d/misdn-init start"
[ii] make your ports (1-4) available in asterisk by editing "/etc/asterisk/misdn.conf"
asterisk:/etc#
```

- Retocamos el fichero `/etc/asterisk/misdn.conf`
- Ejecutamos `/etc/init.d/misdn-init start`

```
asterisk:/etc# /etc/init.d/misdn-init start
-----
Loading module(s) for your misdn-cards:
-----
modprobe --ignore-install hfcmulti type=0x4 protocol=0x12,0x12,0x12,0x12 layermask=0x3,0x3,0x3,0x3 poll=128 debug=0xf
[ii] creating device node: /dev/mISDN
asterisk:/etc#
```



# Configuración BRI (3)

---

## **/etc/misdn-init.conf**

```
card=1,0x4
te_ptp=1,2,3
nt_ptmp=4
poll=128
dsp_options=0
debug=0xf
```

## **/etc/asterisk/misdn.conf**

```
[general]
misdn_init=/etc/misdn-init.conf
debug=0
ntdebugflags=0
ntdebugfile=/var/log/misdn-nt.log
bridging=no
stop_tone_after_first_digit=yes
append_digits2exten=yes
dynamic_crypt=no
crypt_prefix=**
crypt_keys=test,muh
```

```
[default]
context=default
musicclass=default
senddtmf=yes
far_alerting=yes
allowed_bearers=all
nationalprefix=
internationalprefix=00
rxgain=0
txgain=0
te_choose_channel=no
need_more_infos=no
```

```
method=standard
dialplan=0
localdialplan=0
cpndialplan=0
early_bconnect=yes
incoming_early_audio=no
nodialtone=no
immediate=yes
callgroup=1
pickupgroup=1
presentation=1
screen=1
echocancel=yes
```

```
[telefonica]
context=default
ports=1,2,3
msns=*
```

```
[centralita]
context=outgoing
ports=4
msns=*
```



# Informes de llamadas: CDRs



# CDRs

---

- En centralitas, proveedores de servicios, etc. es importante tener un control de las llamadas.
- Asterisk puede generar CDRs (Call Detail Record) en distintos formatos
  - CSV
  - MySQL
  - SQLite
  - ...
- Nos dan **todos** los detalles de las llamadas.
  - Billing.
- Para utilizar el CDR en MySQL es necesario compilar asterisk-addons. (instalar paquete libmysqlclient15-dev)



# CDRs (2)

---

- Por defecto el CDR se genera en CSV en /var/log/asterisk/
- Posibilidad para desarrollar aplicaciones de estadística para CallCenters, ...





# Ejercicio 10

---

- Crear la estructura de tablas necesaria y poner en funcionamiento el sistema de CDRs en base de datos MySQL.
- Receta:
  - Si no lo esta, compilar asterisk-addons.
  - Configurar el fichero `cdr_mysql.conf` para que apunte a nuestra base de datos.
  - Crear la estructura de la base de datos.
  - Cargar el módulo editando el fichero `modules.conf`  
`load => cdr_addon_mysql.so`



# Aspectos avanzados de Asterisk





# AEL2

---

- AEL (Asterisk Extension Language) es una forma diferente de escribir el dialplan.
- Más orientado a programadores:
  - switch
  - while/for
  - ...
- Utiliza el módulo pbx\_ael.so (pbx\_config.so carga el dialplan normal)
- Internamente Asterisk trabaja con el dialplan clasico, “parsea” y convierte el AEL.
- Muy útil, y mucho más legible.
- Más información:  
<http://www.voip-info.org/wiki/view/Asterisk+AEL2>



# AEL2 (2)

---

- **Ejemplo:**

```
context prueba {
    1234 => {
        Dial(SIP/saghul,45);
        switch (${DIALSTATUS}) {
            case BUSY:
                Voicemail(b200);
                break;
            case NOANSWER:
                Voicemail(u200);
                break;
            default:
                Noop(Algo raro ha pasado);
                Hangup;
        }
    }
}
```



# AGI

---

- AGI (Asterisk Gateway Interface) nos permite ejecutar en Asterisk software de terceros escrito en casi cualquier lenguaje
- Permite **extender al infinito** las posibilidades de Asterisk, juntando su potencia, con las posibilidades que ofrece un lenguaje de programación.
- Muchos lenguajes soportados: Python, PHP, Perl, Bash, Java, ...
- Conviene utilizar un lenguaje que no resulte demasiado lento, para no demorar demasiado la ejecución.
- Opinión personal:
  - PHP es un buen lenguaje para AGI.
  - Se ejecuta rápido.
  - No es difícil programar en PHP.
  - La librería `phpagi` nos hace la vida más fácil.  
<http://phpagi.sourceforge.net/>



# AGI (2)

---

- Ejemplo de AGI con phpagi:

```
#!/usr/bin/php
<?php
require_once("phpagi.php");
$myagi = new AGI();
$myagi->set_variable("VAR","hoola");
$myagi ->exec_dial("SIP","saghul",45);
?>
```

- En el dialplan pondríamos:

```
exten => 1234,1,AGI(prueba.php)
exten => 1234,n,noop(${VAR})           //nos saldría hoola
```



# AMI

---

- **¿Qué es Asterisk AMI?**

Asterisk AMI permite que programas cliente se conecten a Asterisk mediante TCP/IP y sean capaces de ejecutar comandos y leer eventos. Por cada cosa que Asterisk realiza se generan eventos que pueden ser leídos mediante una sesión de manager, y el usuario puede tratarlos a su gusto. Además, AMI permite la ejecución de comandos, lo que proporciona la posibilidad de alterar el comportamiento de Asterisk desde un programa hecho a medida.

- **Funcionamiento:**

Para trabajar con AMI es necesario tener un usuario definido en el fichero manager.conf. A partir de aquí hay que establecer una comunicación TCP/IP con el servidor de Asterisk en el puerto 5038, y una vez conectado y autenticado, se puede comenzar a leer los eventos o ejecutar comandos.



# AMI (2)

---

- **¿AJAM?**

De la mano de Asterisk 1.4 viene AJAM (Asynchronous Javascript Asterisk Manager), un nuevo manager, que permite conectar con Asterisk por medio de HTTP. Para poder trabajar con AJAM es necesario configurar los ficheros manager.conf y httpd.conf. Adivina como funciona el nuevo GUI de Asterisk..... premio!

- **Ejemplos de funcionamiento:**

`http://IP_de_Asterisk:8088/asterisk/manager?action=login&username=nombre_de_usuario&secret=contraseña`

Esto abrirá una sesión de Asterisk Manager. Si ahora ejecutas:

`http://IP_de_Asterisk:8088/asterisk/rawman?action=status`

Verás la salida del comando.



# AMI (3)

---

- **Más información:**

<http://www.voip-info.org/wiki-Asterisk+manager+API>

<http://www.voip-info.org/wiki/view/Aynchronous+Javascript+Asterisk+Manager+%28AJAM%29>



# ARA

---

- Asterisk permite ser configurado (algunos ficheros) en una Base de Datos, mediante **Asterisk Realtime Architecture**.
- Soporta MySQL, y es posible migrar a Base de Datos lo siguiente:
  - Configuración de dispositivos SIP e IAX.
  - Configuración de buzones de voz.
  - Configuración de colas.
- Las estructuras de la bases de datos están en <http://www.voip.info.org>
- El fichero a configurar es **extconfig.conf**
- El fichero `res_mysql.conf` contiene la configuración de la base de datos.





# ARA (2)

---

- ARA dispone de 2 tipos de Realtime: **estático y dinámico**:
  - **Estático**: La configuración esta almacenada en la BD, pero Asterisk la carga al arrancar como si fuera un fichero. Si se realiza algún cambio, es necesario hacer un reload.
  - **Dinámico**: La configuración esta almacenada en la BD y Asterisk realiza una consulta a esa BD cada vez que necesita un dato. No es necesario hacer reload si se han realizado cambios. Mucha carga para el servidor...
- Todos los ficheros **no soportan** Realtime Dinámico.
  - SIP, IAX y los buzones de voz en Realtime Dinámico.
  - Colas en Realtime Dinámico ???



# ARA (3)

---

- Ejemplo de configuración (extconfig.conf):

```
[settings]
;;RealTime Dynamic
;sipusers => mysql,asterisk,dispositivos_sip
;sippeers => mysql,asterisk,dispositivos_sip
;iaxusers => mysql,asterisk,dispositivos_iax
;iaxpeers => mysql,asterisk,dispositivos_iax
;voicemail => mysql,asterisk,buzones

;;RealTime Static
;sip.conf => mysql,asterisk,sip_conf
;extensions.conf => mysql,asterisk,extensions_conf
;iax.conf => mysql,asterisk,iax_conf
;queues.conf => mysql,asterisk,queues_conf
;voicemail.conf => mysql,asterisk,voicemail_conf
```



# Asterisk-GUI



# Asterisk-GUI

---

- Interfaz gráfica para Asterisk creada por Digium.
- Se lanzó tras el lanzamiento del Asterisk Appliance.





# Asterisk-GUI (2)

---

- Esta en desarrollo... todavía no es muy estable.
- Para instalarla, la descargamos del repositorio SVN de Digium:

```
#svn co http://svn.digium.com/svn/asterisk-gui/trunk asterisk-gui  
#cd asterisk-gui  
#./configure  
#make  
#make install
```

- Hay que configurar 2 ficheros:
  - manager.conf
  - http.conf



# Asterisk-GUI (3)

---

- Configuración de **manager.conf**

```
[general]
displaysystemname = yes
enabled = yes
webenabled = yes
port = 5038
bindaddr = 0.0.0.0
```

```
[saghul]
secret = 1234
read = system,call,log,verbose,command,agent,user,config
write = system,call,log,verbose,command,agent,user,config
```

- Configuración de **http.conf**

```
[general]
enabled=yes
enablestatic=yes
bindaddr=0.0.0.0
bindport=8088
```



# Asterisk-GUI (4)

---

- Una vez Asterisk-GUI esta compilado, podemos comprobar si todo esta correcto ejecutando (en el directorio asterisk-gui):

```
make checkconfig
```

- Si todo hay ido bien, se nos mostrará un mensaje de que ya podemos acceder a finalizar la instalación, entrando en la dirección:

```
http://ip_de_asterisk:8088/asterisk/static/config/setup/install.html
```



# Asterisk-GUI (5)

Asterisk GUI Setup Wizard - Iceweasel

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://192.168.1.123:8088/asterisk/static/config/setup/install.html

Google

digium | Asterisk

Welcome to Asterisk GUI setup wizard

Please enter your admin Username and Password

Username:

Password:

Login

Back Next

Copyright ©2006-2007 Digium, Inc. Digium® and Asterisk® are registered trademarks of Digium, Inc. All Rights Reserved. [Legal Information](#)

Terminado





# Asterisk-GUI (6)

Asterisk GUI Setup Wizard - Iceweasel

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://192.168.1.123:8088/asterisk/static/config/setup/install.html

Google

digium Asterisk

Welcome to Asterisk GUI setup wizard

Start

**Verify Analog Ports**

Local Extension Settings

Service Providers

Calling Rules

VoiceMail Settings

User Extensions

Incoming Calls

Finish

**Step 1 of 7 - Verify Analog Ports**

**No Analog ports have been detected on your system**

If you do not have any analog hardware please click Next to proceed

Back Next

Copyright ©2006-2007 Digium, Inc. Digium® and Asterisk® are registered trademarks of Digium, Inc. All Rights Reserved. [Legal Information](#)

Terminado



# Asterisk-GUI (7)

---

- Cuando se termine la configuración inicial, el usuario será redireccionado a:

`http://192.168.1.123:8088/asterisk/static/config/cfgbasic.html`

- Desde esa ventana es posible configurar Asterisk de manera bastante intuitiva.
- Muestra mucha información: gráficos, logs, CDR ???
- **Editor de ficheros y acceso al CLI!**
- Aún le queda por mejorar, pero será una herramienta muy útil...



# Asterisk-GUI (8)

Asterisk GUI (Beta) -- User Extensions - Iceweasel

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://192.168.1.123:8088/asterisk/static/config/cfgbasic.html

digium Asterisk

About Digium | Report a Bug | Help

Home

User and Phone Configuration **Activate Changes** **Logout**

Users

Users is a short cut for quickly adding and removing all the necessary configuration components for any new phone.

2000 -- Saghul

2001 -- Lolailo

2002 -- Jose Antonio

Extension: 2000

Name: Saghul

Password: 1234

VM Password:

E-mail: saghul@gmail.com

Caller ID:

Analog Phone: No Analog lines detected.

Dial Plan: DialPlan1

Extension Options:

Voicemail  In Directory

SIP  IAX

CTI  Is Agent

Call Waiting  3-Way Calling

Can Reinvite  NAT

rfc2833 DTMFMode

Codecs

disallow:

allow:

New Delete Save Cancel

Call Waiting: Check this option if the User or Phone should have Call-Waiting capability.

Copyright 2006-2007 Digium, Inc. Digium and Asterisk are registered trademarks of Digium, Inc. All Rights Reserved. Legal Information

Terminado



# Anexo A: Softphones



# X-Lite

- Protocolo SIP
- Soporte STUN
- g711a/u, gsm, ilbc, speex
- Windows, GNU/Linux, MacOSX





# Zoiper (antes Idefisk)

- Protocolo SIP e IAX2
- Soporte STUN
- g711a/u, gsm, ilbc, speex, g729 (version BIZ)
- Windows, GNU/Linux, MacOSX





# Cubix

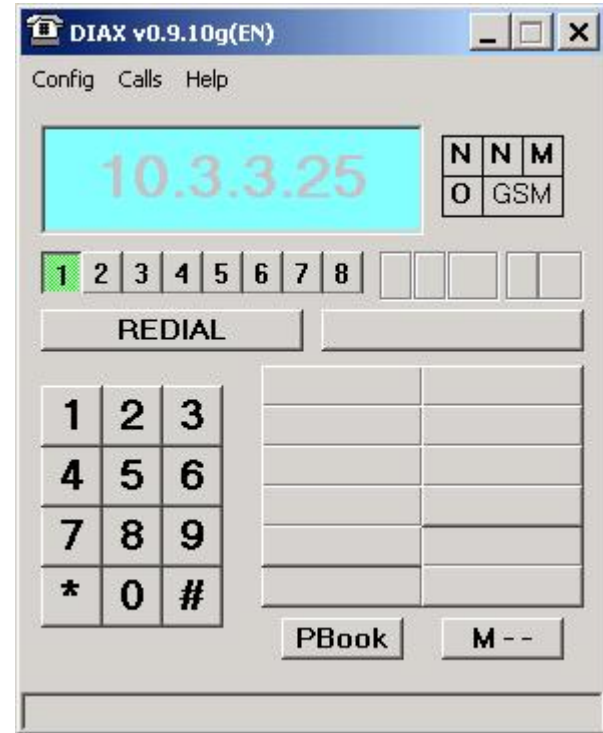
- Protocolo SIP e IAX2
- Soporte STUN
- g711a/u, gsm, ilbc, speex, g729
- Windows





# DIAX

- Protocolo IAX2
- g711a/u, gsm, ilbc, speex
- Windows







# KIAX

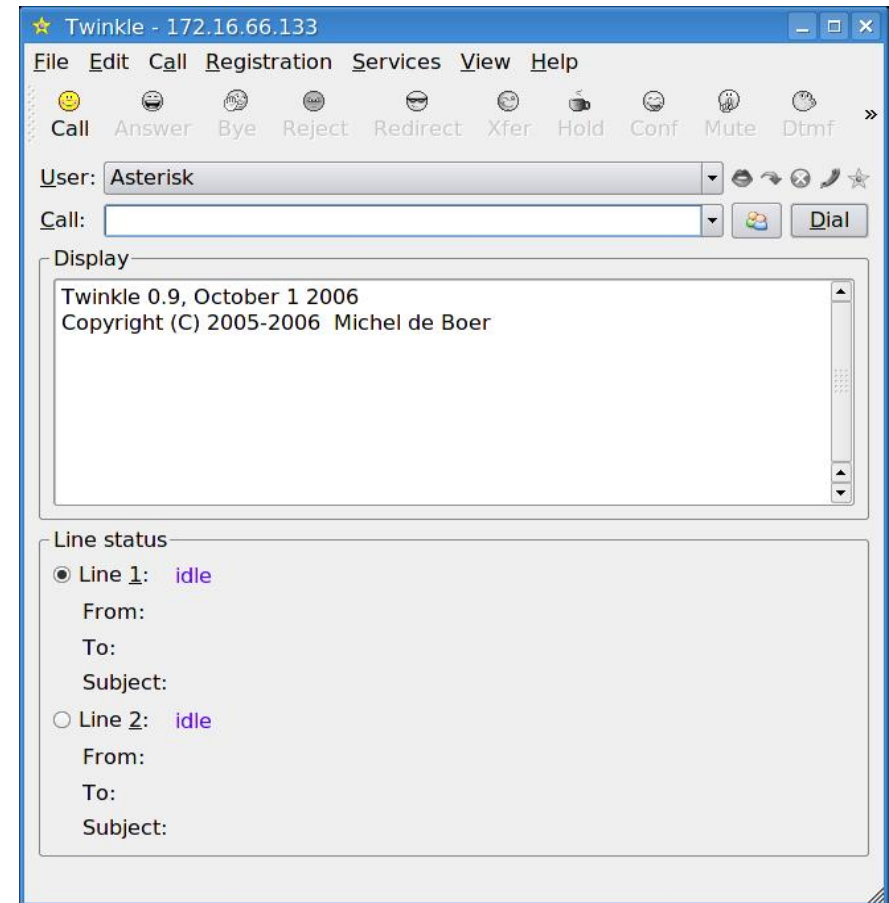
- Protocolo IAX2
- g711u, gsm, speex
- Windows, Linux, \*BSD





# Twinkle

- Protocolo SIP
- g711a/u, gsm, speex, g726
- GNU/Linux (Qt)





# Ekiga

- Protocolo SIP e IAX2.
- g711a/u, gsm, speex, g726.
- GNU/Linux.





# Anexo B: Links de interés



# Links

---

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

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

<http://planetasterisk.org/>

<http://planet.voz-ip.com/>

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

<http://www.asterisk.org/doxygen/>

<http://www.asterisk-es.org/>

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

<http://asteriskcounter.sinologic.net/>



# Anexo C: Configuración de terminales



# Thomson ST2030

---

- Pulsar la tecla de “menú”.
- Pulsar el botón central, que dice “admin” e introducir la contraseña: 784518.
- Comprobar la dirección IP y entrar a ella con el navegador poniendo /admin.html al final
- Con en navegador abierto, reiniciar el teléfono para que se activen todas las opciones.





# Thomson ST2030 (2)

Iceweasel

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://192.168.6.190/main.html

thomson st2030 admin password

THOMSON

HOME SETUP ADVANCED UTILITY STATUS LOGOUT

Network Interface  
Network Setup

VoIP Service  
Basic Setup

Auto Provisioning  
Basic Setup  
APS Log

**Basic Setup**

Profile Name : Profile 1

Transfer to Voice Mail  
Telephone Number :

**Primary SIP Server :**

SIP Unregister  
SIP Transport  UDP  TCP  Connect Reuse

Service Domain: 192.168.6.50

Registrar Server Address: 192.168.6.50 port : 5060 (1025~49151)

Proxy Server Address: 192.168.6.50 port : 5060 (1025~49151)

SIP Local Port: 5060 (1025~49151)

Registration Timer: 3600 (60~200,000)

Register Frequency: 600 (60~1800 sec)

Ring Tone: Default

**Backup SIP Server :**

SIP Unregister  
SIP Transport  UDP  TCP  Connect Reuse

Service Domain:

Registrar Server Address: port : 5060 (1025~49151)

Proxy Server Address: port : 5060 (1025~49151)

SIP Local Port: 5060 (1025~49151)

Registration Timer: 3600 (60~200,000)

Register Frequency: 600 (60~1800 sec)

**User Accounts :**

Phone Number	Phone Name	Authentication ID	Password
200	200	200	****

Apply Cancel Back

Terminado





# Thomson ST2020

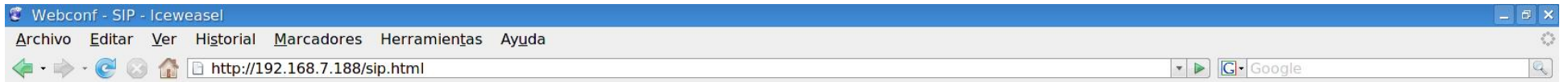
---

- Comprobar la dirección IP en la pantalla y entrar a ella con el navegador, poniendo /admin.html al final.
- El usuario y la contraseña son los mismos que en el ST2030.





# Thomson ST2020 (2)



[Admin](#) / [Network](#) / [SIP](#) / [Media](#) / [Phone](#) / [Speeddial](#) / [Advanced](#) / [Info](#) / [Upgrade](#)

## SIP configuration

On this page you can configure the SIP settings of this SIP device and the proxy / registrar server.

Display name:	<input type="text" value="200"/>
User name:	<input type="text" value="200"/>
Authentication user:	<input type="text" value="200"/>
Authentication password:	<input type="password" value="****"/>
SIP registrar/Domain name:	<input type="text" value="192.168.6.50"/>
SIP registrar port:	<input type="text" value="5060"/>
SIP outbound proxy server:	<input type="text" value="192.168.6.50"/>
SIP outbound proxy port:	<input type="text" value="5060"/>
Register interval(sec):	<input type="text" value="3600"/>
Local SIP port (0=random):	<input type="text" value="5060"/>
Local RTP port (0=random):	<input type="text" value="41000"/>

Terminado



# Linksys SPA-9xx

---

- Comprobar la dirección IP:
  - Tecla “menú” -> 9.
- Acceder vía web.
- Pinchar en “admin login”.
- Pinchar en “advanced”.
- Pestaña “Ext1”.





# Linksys SPA-9xx (2)

<b>Proxy and Registration</b>			
Proxy:	192.168.6.50	Use Outbound Proxy:	no
Outbound Proxy:		Use OB Proxy In Dialog:	yes
Register:	yes	Make Call Without Reg:	no
Register Expires:	3600	Ans Call Without Reg:	no
Use DNS SRV:	no	DNS SRV Auto Prefix:	no
Proxy Fallback Intvl:	3600	Proxy Redundancy Method:	Normal
<b>Subscriber Information</b>			
Display Name:	hardphone	User ID:	hardphone
Password:	*****	Use Auth ID:	no
Auth ID:			
Mini Certificate:			
SRTP Private Key:			
<b>Audio Configuration</b>			
Preferred Codec:	G711a	Use Pref Codec Only:	no
G729a Enable:	yes	G723 Enable:	yes
G726-16 Enable:	yes	G726-24 Enable:	yes
G726-32 Enable:	yes	G726-40 Enable:	yes
Release Unused Codec:	yes	DTMF Process AVT:	yes
Silence Supp Enable:	no	DTMF Tx Method:	Auto
<b>Dial Plan</b>			
Dial Plan:	(*xx [3469]11 0 00 [2-9]xxxxxx 1xxx[2-9]xxxxxx50 xxxxxxxxxxxxx.)		
Enable IP Dialing:	yes		



# Anexo D: Hardware VoIP



# Gama Alta

---



Cisco Systems

Polycom





# Gama Media

---



Linksys

Thomson



Snom



# Gama Baja

---

Grandstream







# WiFi

---



UT-Starcom:  
F1000 y F3000



Linksys:  
WIP300



ZyXel:  
Prestige 2000W

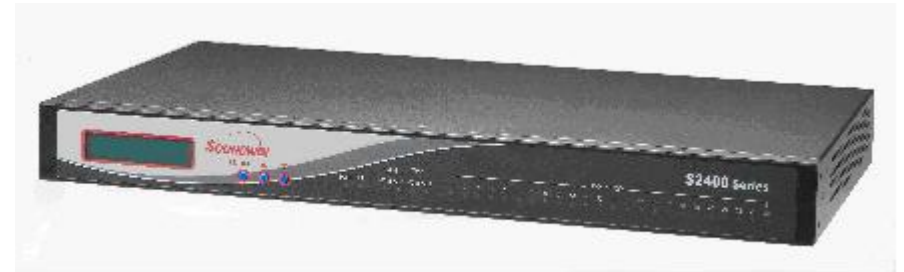


# ATAs

---



Grandstream



Soundwin



Linksys



# PBX “in-a-box”

---



Digium:  
Asterisk Appliance



Linksys:  
SPA-9000



Fonality:  
Trixbox Appliance



# Licencia



# Licencia

---



<http://creativecommons.org/licenses/by-nc/2.5/es/>



¡¡GRACIAS A TODOS!!