

Introdución a VoIP

Historia de VoIP

VoIP comezou como o resultado do traballo dun grupo de xóvenes en Israel durante 1995. Naquela época a única comunicación posible era de PC-a-PC. Pouco máis tarde Vocaltec, Inc. anunciou o lanzamento do primeiro softphone que chamaron "Internet Phone Software". Este softphone fixeráse para ser usado nun PC caseiro que tiña tarxeta de son, micrófono, parlantes e modem. O software funcionaba comprimindo o sinal de voz, convertíndoa en paquetes de voz que eran enviados por Internet (exactamente igual que hoxe). O software só funcionaba se vos dous PCs tiñan o mesmo software e o mesmo hardware. E foi comercialmente un fracaso principalmente porque as comunicacións de banda ancha aínda non estaban dispoñibles.

En 1997 Jeff Pulver decide xuntar por primeira vez aos poucos usuarios, fabricantes, e interesados nesta tecnoloxía en VON ^[1], a primeira feira/congreso que actualmente segue sendo o maior evento de VoIP. Agora Pulver organiza VON dúas veces por ano en EEUU, e agora tamén unha vez por ano en varios países de Europa. Tamén formou unha compañía prestadora do servizo VoIP chamada FreeWorldDialup ^[2] comúnmente chamada FWD (que pode confundirse co termo FWD = transferencia de chamadas) e é co-fundador de Vonage ^[3], ou proveedor de VoIP máis grande de EEUU. Pulver ten varias empresas relacionadas coa VoIP entre elas PulverMedia, a súa empresa encargada de organizar VON e publicar medios en todo o mundo.

En 1998 VoIP deu outro gran salto. Un grupo de emprendedores comezou a fabricar os primeiros ATA/gateways ^[4] para permitir as primeiras comunicacións PC-a-teléfono convencional e finalmente as primeiras comunicacións teléfono-convencional - a - teléfono-convencional (con Atas en cada extremo). Algúns destes emprendedores inicialmente daban o servizo sen cargo a vos seus clientes para que poideran probar a calidade e a tecnoloxía. Estas chamadas contiñan publicidade non inicio e ao final de cada comunicación. Lestes servizos só prestábanse en EEUU e funcionaban grazas a esta publicidade. A miúdo debía comezarse a comunicación a través dunha PC para logo pasar a un teléfono convencional. Neste punto VoIP sumaba o 1% do total do tráfico de voz. Durante 1998 tres fabricantes comezaron a fabricar switches de Layer 3 con QoS.

En 1999 Cisco vende as súas primeiras plataformas corporativas para VoIP. Utilizábase principalmente o protocolo H323 de sinalización.

No ano 2000 VoIP representaba máis do 3% do tráfico de voz. O mesmo ano **Mark Spencer** un estudante da Universidade de Auburn crea Asterisk ^[5], a primeira central telefónica / conmutador baseada en Linux cun PC simple cun código fonte aberto. Asterisk hoxe ofrece unha solución freeware para fogares/pequenas empresas e solucións IP-PBX corporativas. Mark Spencer é o CEO de Digium ^[6].

En 2002 o protocolo SIP comeza a desplazar al H323 ^[7].

En 2003 dous mozos universitarios - Jan Friis e Niklas Zenntrom - crean un softphone gratuito facilmente instalable en calquera PC que pode atravesar todos os firewalls e routers inclusive os corporativos. Ese produto é Skype, que se propaga cunha velocidade increíble, e chega en decembro de 2005 a contar con 50 millóns de usuarios.

En 2005 Skype é comprada en U\$ 2600 millóns por eBay. Skype ten un só problema: os usuarios de Skype non usan un protocolo standard senón un propio, que Skype non divulga. Iso non permite a un usuario de Skype comunicarse con outro usuario que non sexa de Skype. Pero Skype esta cambiando.....

Enlaces externos

- [1] <http://www.von.com/>
- [2] <http://www.siptosip.net/>
- [3] <http://www.vonage.com/index.php?ic=1>
- [4] <http://voipex.blogspot.com/2006/03/adaptadores-telefonicos-analogicos.html>
- [5] <http://www.asterisk.org>
- [6] <http://www.digium.com/en/>
- [7] <http://voipex.blogspot.com/2006/03/protocolos-de-sealizacion.html>

Fuente: http://informatica/manuais/index.php?title=Introducción_a_VoIP&oldid=11579

Autores principales: Cristina,

O sistema Asterisk

ASTERISK

Asterisk é una aplicación de software libre (baixo licencia GPL) que proporciona funcionalidades dunha central telefónica (PBX)(1)e é completamente configurable.. Como calquer PBX, pódese conectar un número determinado de teléfonos para facer chamadas entre sí e incluso conectar a un proveedor de VoIP ou ben a unha RDSI tanto básicos como primarios.

- **(1)O concepto PBX**

PBX Literalmente a sigla quere decir Private Branch Exchange, e con este término conócense aos diferentes tipos de centrais telefónicas de uso privado, ou centrais telefónicas utilizadas nas empresas, sen importar a tecnoloxía que utilicen, ou a súa riqueza de funcionalidades. Este termo é usado del lado dos operadores para definir un grupo de líneas análogas que están agrupadas baixo un so número.

Mark Spencer, de Digium, inicialmente creou Asterisk e actualmente é o seu principal desenvolvedor, xunto con outros programadores que contribuen a correxir erros e engadir novidades e funcionalidades. Orixinalmente creado para o sistema operativo GNU/Linux, Asterisk actualmente tamén distribúese en versións para os sistemas operativos BSD, MacOSX, Solaris e Microsoft Windows, aínda que a plataforma nativa (GNU/Linux) é a mellor soportada de todas.

Asterisk inclúe moitas características anteriormente sólo dispoñibles en costosos sistemas propietarios PBX como buzón de voz, conferencias, IVR, distribución automática de chamadas, e outras moitas máis. Os usuarios poden crear novas funcionalidades escribindo un dialplan na linguaxe de script de Asterisk ou engadindo módulos escritos na linguaxe C ou en calquera outra linguaxe de programación soportada por Linux.

Funcionalidades Central Asterisk

- Desde 2 até 150 extensións por cada sistema
- Máis de 150 extensións con sistemas DV-Asterisk traballando en paralelo (clustering)
- Desde 1 até 120 liñas dixitais RDSI por cada sistema]
- Desde 1 até 72 liñas analóxicas por cada sistema en formato barebon
- Desde 1 até 24 liñas dixitais RDSI a diario en formato barebone (48 liñas de voz)
- Desde 1 até 12 liñas de primario a diario en formato barebone (360 liñas de voz)
- Soporta liñas de telefonía analóxica, liñas RDSI, accesos primarios, VoIP (voz a través de Internet)
- Transferencia de chamadas, internas e externas
- Desvío de chamadas se está ocupado ou non contesta
- Opción Non molestar (Do Not Disturb)
- Parking de chamadas (Call Parking)
- Chamada en espera (Hold)
- Extensións remotas fóra de oficina (empregados que traballan en casa ou outras sedes)
- Grupos de chamada (Ring groups)
- Identificador de llamante (CallerID)
- Sistema DISA. é un método polo cal unha persoa externa á oficina pode realizar chamadas a través da central:

exemplo: unha comercial chama á central, teclea a extensión que lle dá acceso exterior, o sistema pídlle un pin e dálle ton para marcar un número

- Operador Dixital (menús interactivos e guiados)
- Música en espera e en transferencia (ficheiros MP3 actualizables polo usuario)
- Captura de chamadas de forma remota (remote pickup)
- Caixas de correos de voz (xeral, individuais, por grupos) protexidos por contrasinal
- Indicador visual de mensaxe en espera
- Envío de mensaxes da caixa de correos ao correo electrónico
- Grupos de caixas de correos
- Interfaz web para acceder ás Caixas de correos
- Xestión da caixa de correos de voz mediante o terminal telefónico e páxina web
- Xestión de listas negras (números telefónicos con acceso prohibido)
 - Paging/Intercom.
 - Accións a realizar segundo horarios e datas (horario laboral, días festivos, etc.)
 - Salas de conferencia (2 ou máis terminais simultaneamente)
 - Rexistro e listaxes de chamadas entrantes e saíntes, con gráficas de consumo.
 - Rutado de chamadas saíntes por parámetros
 - Programación de novas funcionalidades a medida
 - Detección automática de entrada de faxes
 - Recepción de fax desde o propio sistema (só liñas RDSI), e posterior envío por e-mail
 - Envío de faxes desde o propio sistema a través de interfaz web
 - Rutado de chamadas entrantes (Direct Inward Dialing) en liñas dixitais
 - Panel gráfico de Operadora a través de navegador web (estado de liñas e terminais)
 - Posibilidade de integrar un sistema de chamadas pre-pago (solución para locutorios telefónicos)
 - Xestión de colas de chamadas entrantes (solución para Call Centers)
 - Gravación de chamadas entrantes e saíntes

- Monitorización de chamadas en curso
- Soporta videoconferencia con protocolos SIP e IAX2

Funciones VoIP

- Pasarela VoIP <-> RTB/RDSI
- Soporta redes Ethernet 10/100/1000/10000 Mb/s
- Posibilidade de interconexión de varias sedes e oficinas por VoIP
- Módulo Wi-Fi opcional (soporta terminales WiFi y PDA)
- Soporta teléfonos IP (fijos, móviles Wi-Fi e " softphone ")
- Protocolos SIP, H.323, MGCP, IAX2)
- Codecs: GSM, Alaw, Ulaw, iLBC, ADPCM, ...

Funcionalidades Conectividad

- Firewall integrado (incluido)
- Router de saída hacia Internet (incluido)
- Proxy caché de web (opcional)
- Servidor de correo seguro con tráfico cifrado, antivirus e anti-spam (opcional)
- Servidor DHCP (opcional)
- Servidor de mensajería instantánea Jabber con pasarelas MSN, Yahoo, ICQ (opcional)
- Servidor de impresoras (opcional)
- Redes privadas virtuales (VPN) (opcional)
- Proxy HTTP y FTP (opcional)
- Servidor de correo corporativo (opcional)
 - Antispam
 - Antivirus
 - Webmail
 - Pasarela E-Mail <-> SMS

Se desea dotar ao seu negocio dun sistema integral de comunicacións de altas prestacións e baseada en Software Libre, sen dúbida Asterisk é a solución que necesita. De igual maneira, é o sistema ideal para establecer unha comunicación VoIP entre as distintas sucursais dunha mesma empresa, aforrando en chamadas telefónicas e complexidade de infraestructuras.

Fuente: http://informatica/manuais/index.php?title=O_sistema_Asterisk&oldid=11825

Autores principales: Cristina, Seco,

Instalación de Servidor Linux con CentOS

Instalación de CentOS 5.2 parte I

Arrancar có CD-ROM ou DVD do CentOS e presionar ENTER cando apareza o o prompt:



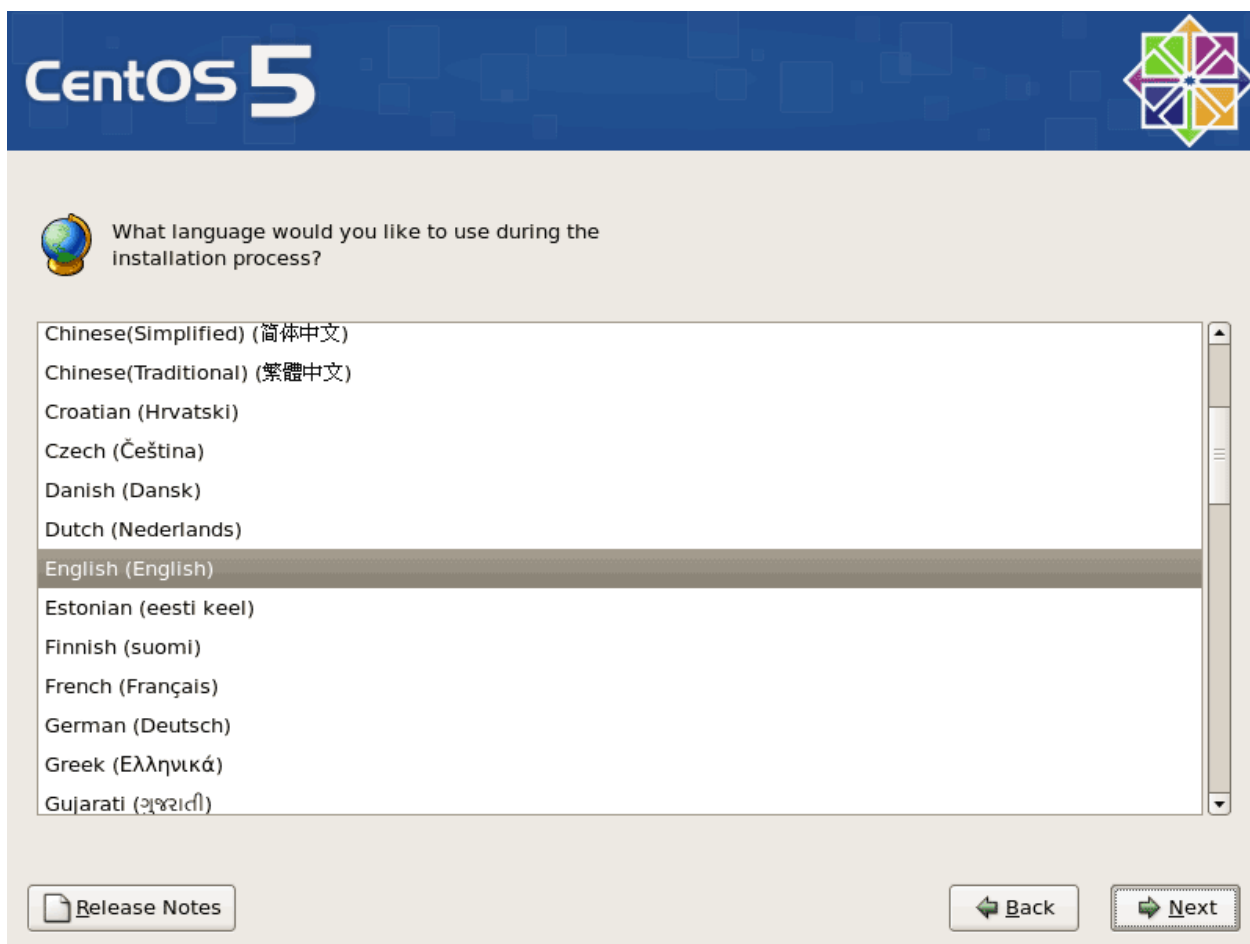
Pode levar un pouco o testear o medio onde está o CentOS polo que saltaremos esa opción:



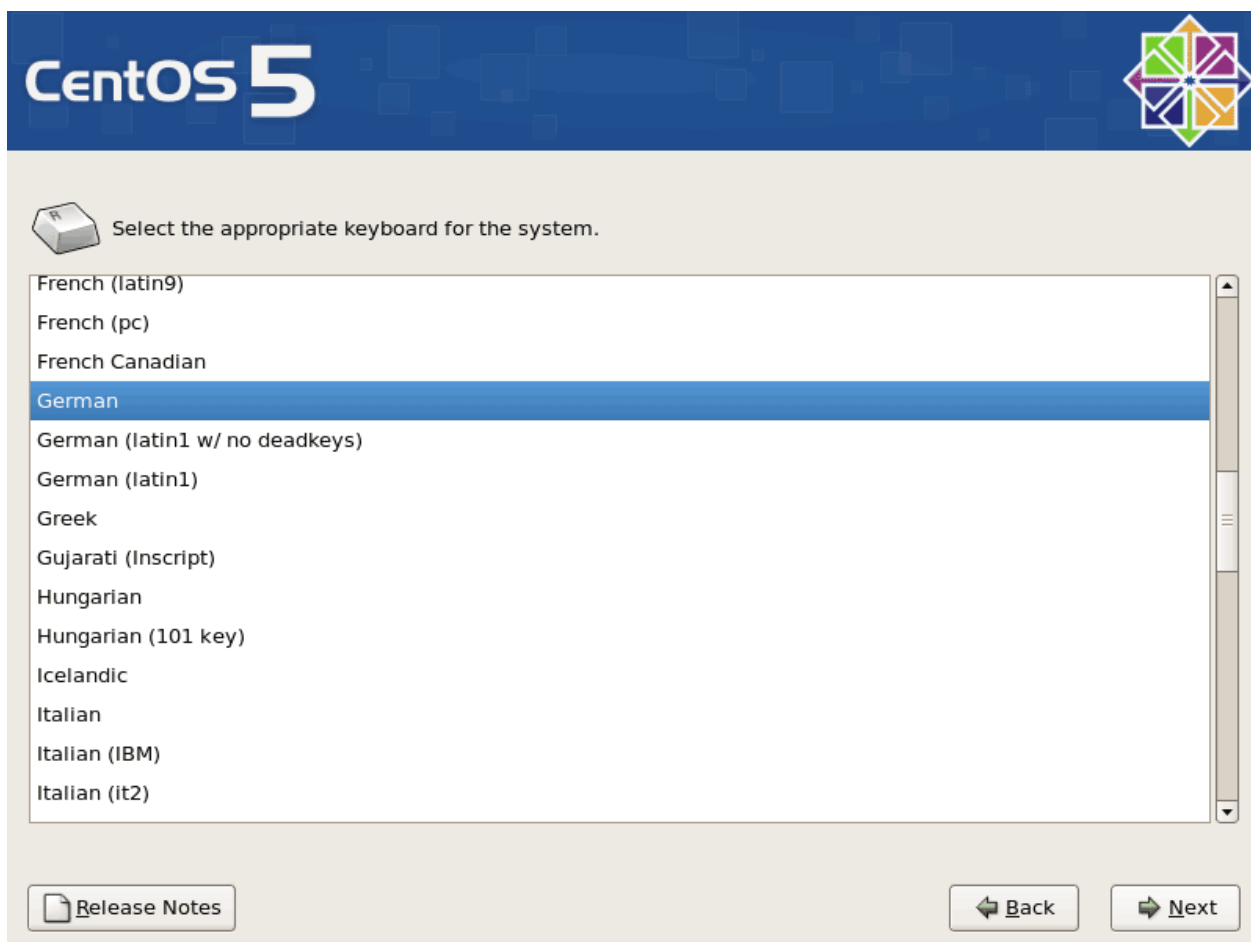
A pantalla de bienvenida do instalador aparecerá, pulsamos en Next:



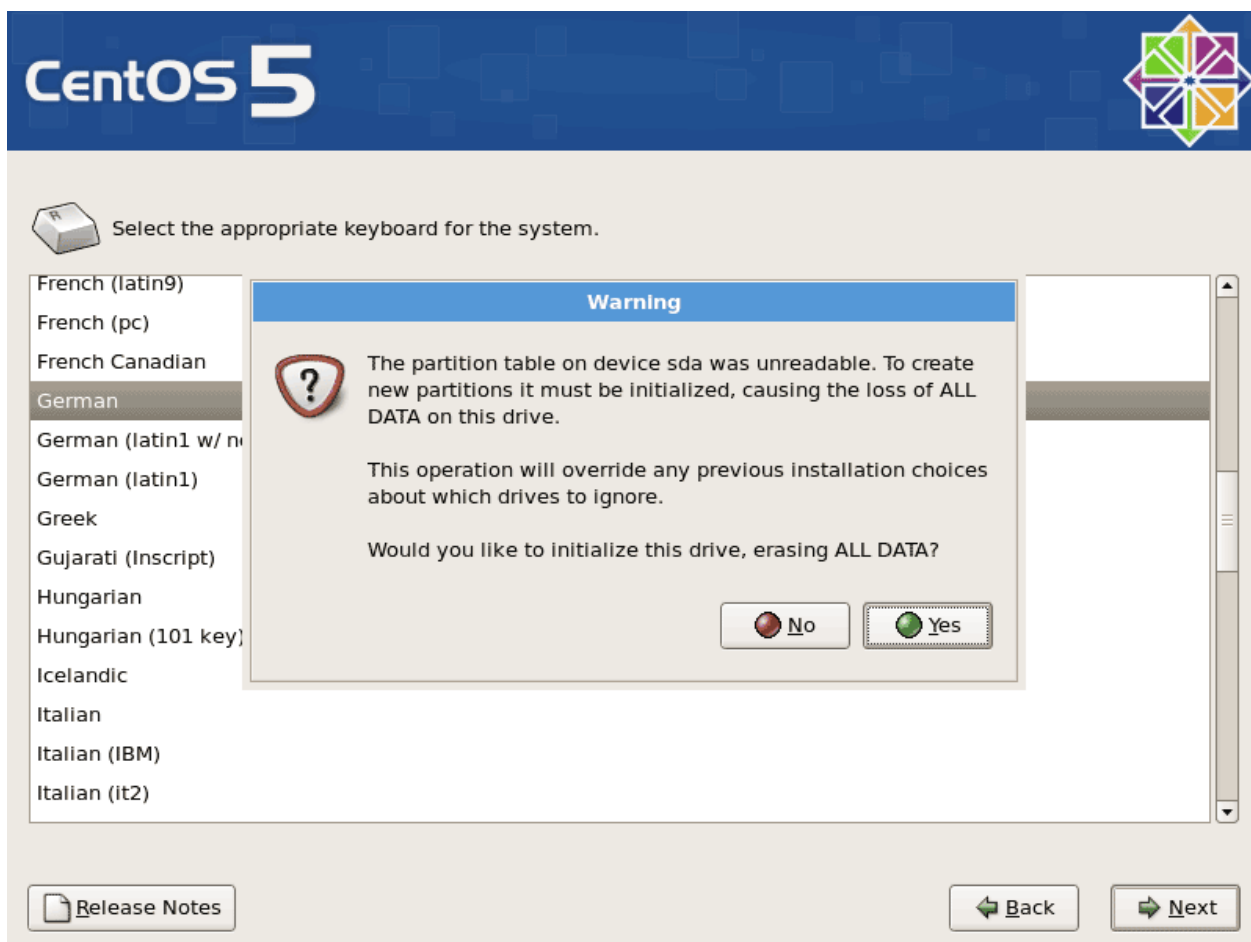
A continuación escollemos o language:





Seleccionamos o tipo de teclado:



Estamos instalando CentOS 5.2 nun sistema novo, polo que responderemos Yes a cuestión sobre si queremos inicializar o disco e tódolos seus datos.



Agora temos que seleccionar un esquema de particións para a instalación. Para simplificar o proceso seleccionamos a opción de eliminar as particións de linux e instalar coa plantilla por defecto. Isto resultará nun pequeno /boot e unha partición máis longa / así como a partición de swap. Dende logo que tamén teremos a libertade de particionar o disco duro como queiramos.



Installation requires partitioning of your hard drive. By default, a partitioning layout is chosen which is reasonable for most users. You can either choose to use this or create your own.

Remove Linux partitions on selected drives and create default layout. ▾

Select the drive(s) to use for this installation.

<input checked="" type="checkbox"/>	sda	30718 MB	VMware, VMware Virtual S
-------------------------------------	-----	----------	--------------------------

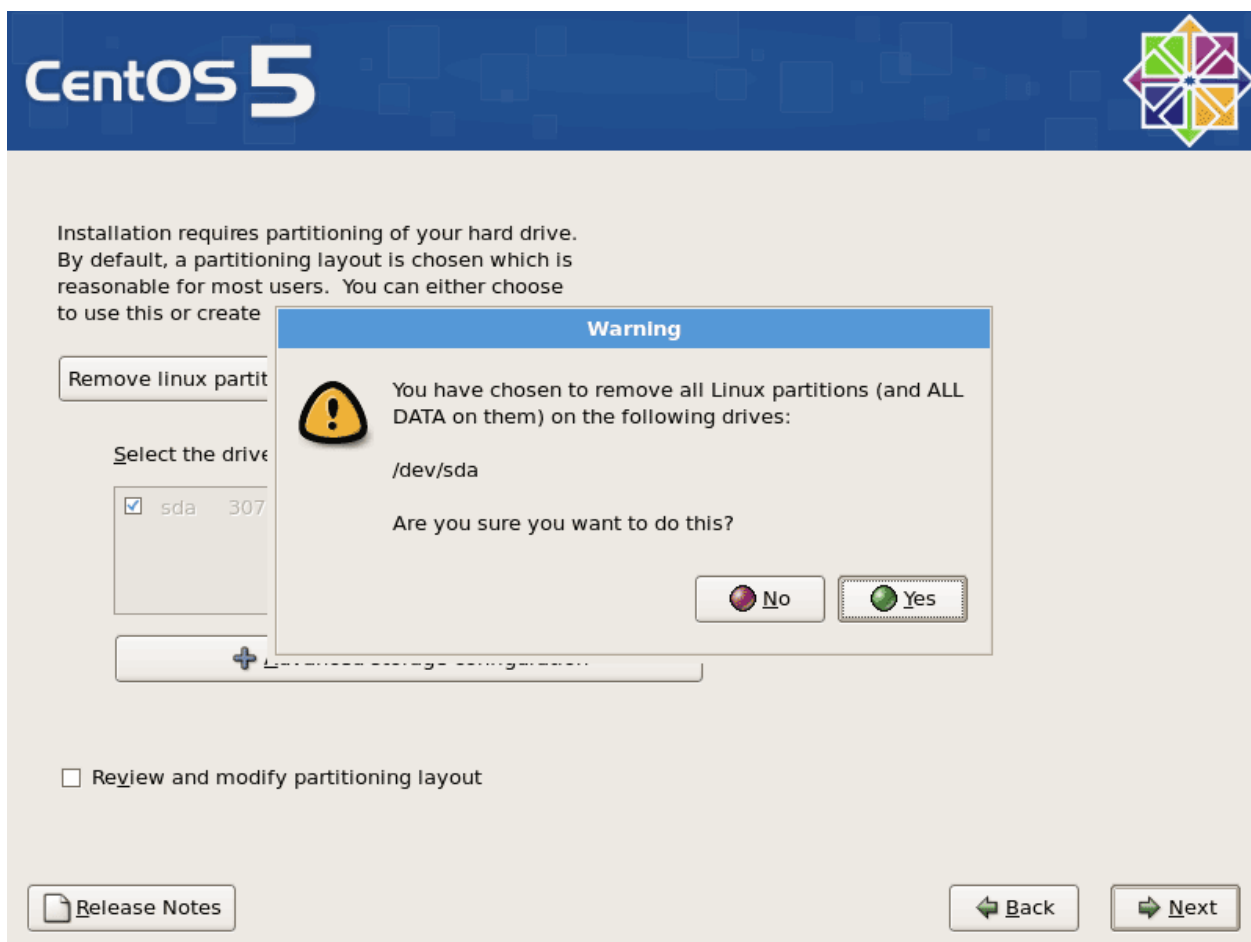
[+ Advanced storage configuration](#)

Review and modify partitioning layout

[Release Notes](#)

[Back](#) [Next](#)

Respondemos con Yes a pregunta de si queremos hacer iso:



Na configuración de rede. A configuración por defecto é configurar as interfaces de rede por DHCP, pero estamos instalando un servidor, polo que a configuración estática de direccións IP non é unha mala idea. Pulsaremos na opción Edit na parte superior dereita.



Network Devices

Active on Boot	Device	IPv4/Netmask	IPv6/Prefix
<input checked="" type="checkbox"/>	eth0	DHCP	Auto

[Edit](#)

Hostname

Set the hostname:

automatically via DHCP

manually (e.g., host.domain.com)

Miscellaneous Settings

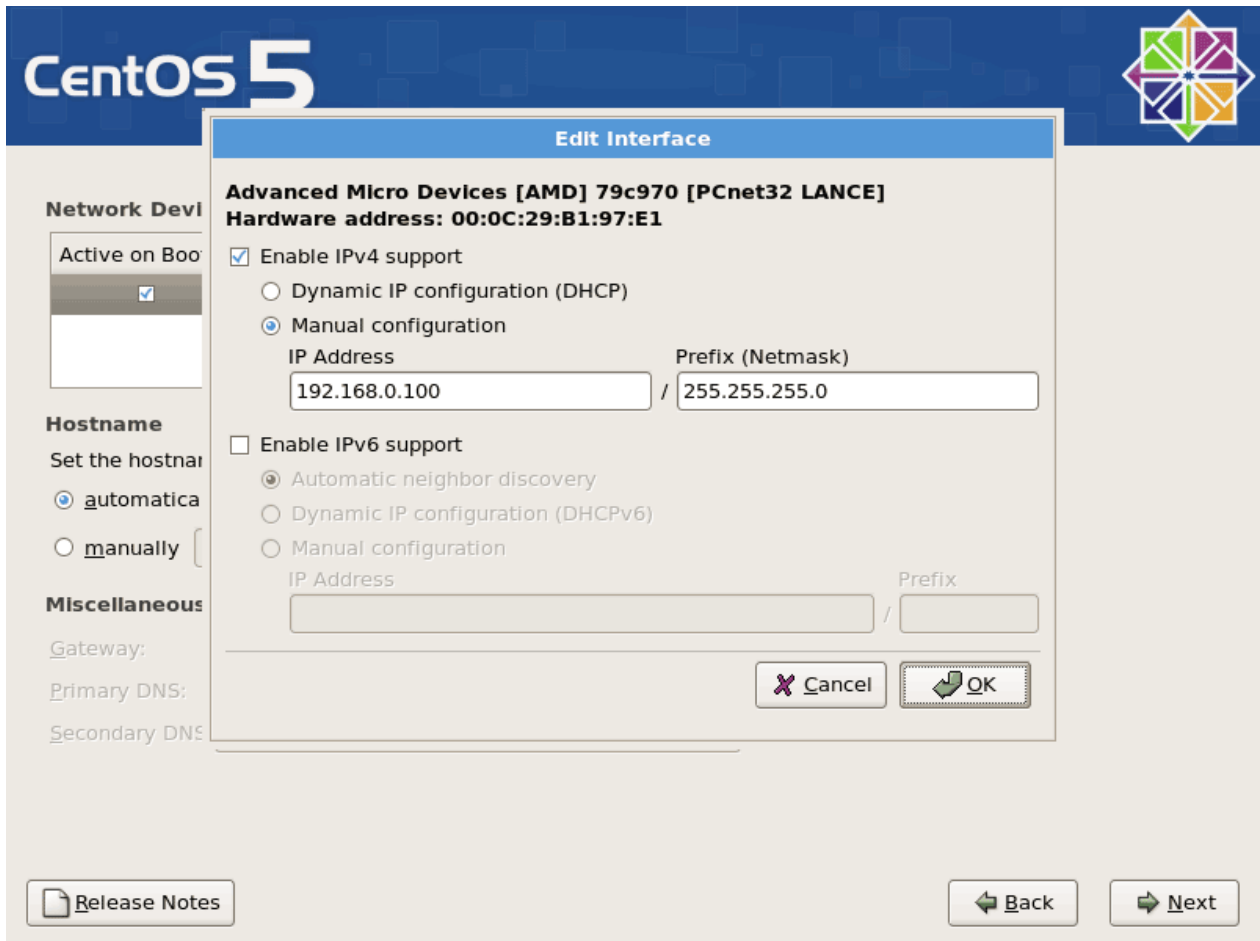
Gateway:

Primary DNS:



Secondary DNS:

[Release Notes](#) [Back](#) [Next](#)

Na ventana que se amosa desmarcamos a opción de Usar Configuración IP dinámica (DHCP) , habilitamos si o desexamos soporte IPV6 e dámoslle á tarxeta de rede unha dirección IP estática (neste caso imos a empregar a dirección 192.168.0.100) e unha máscara clase C (255.255.255.0). Si non estamos seguros da máscara a empregar na páxina <http://www.subnetmask.info> ^[1])



Pomos o nome do host manualmente, exemplo server1.example.com, e introducimos o gateway (192.168.0.1) e ata dous servidores DNS (exemplo 213.191.92.86 e 145.253.2.75):

Network Devices

Active on Boot	Device	IPv4/Netmask	IPv6/Prefix
<input checked="" type="checkbox"/>	eth0	192.168.0.100/24	Disabled

[Edit](#)

Hostname

Set the hostname:

automatically via DHCP

manually (e.g., host.domain.com)

Miscellaneous Settings

Gateway:

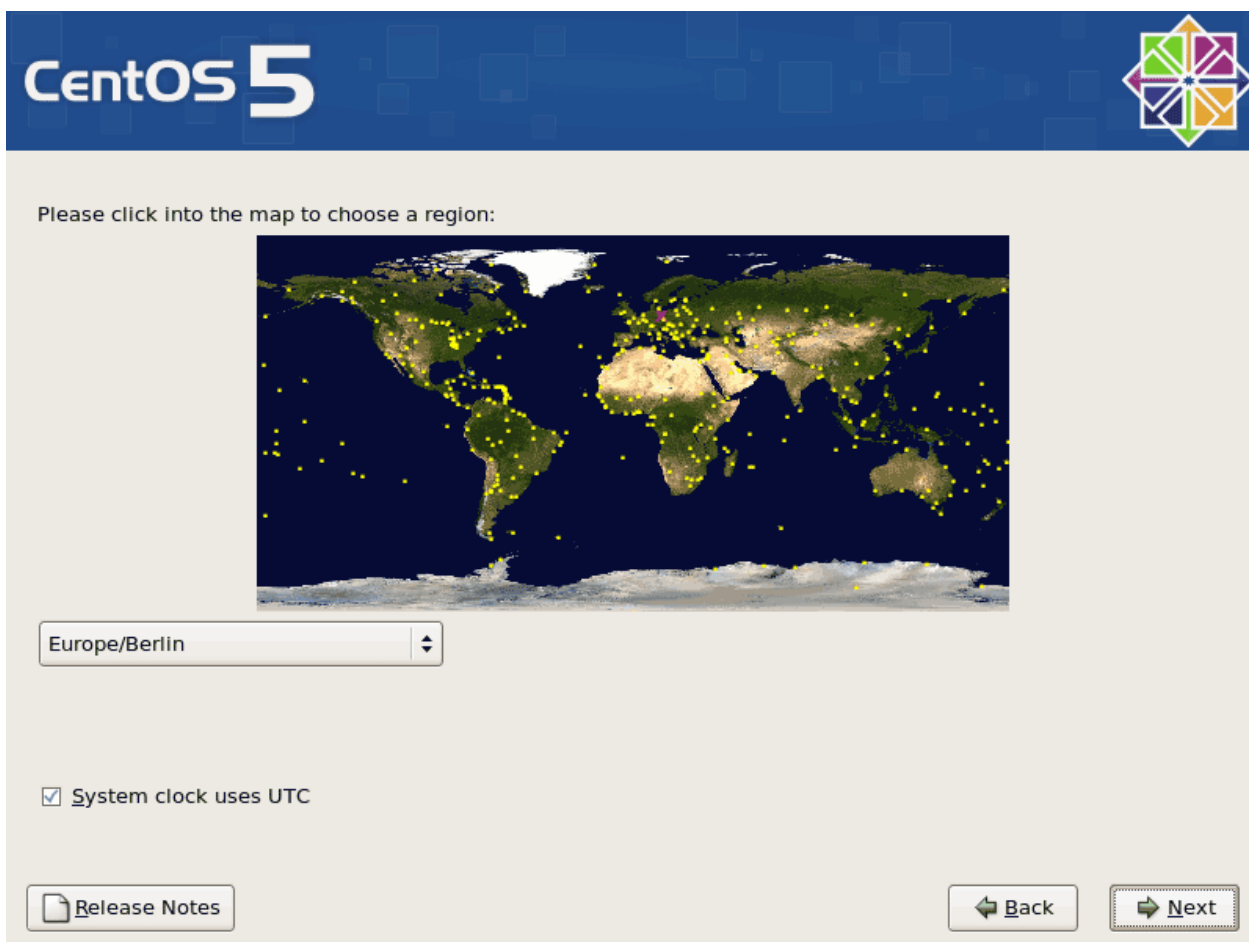
Primary DNS:

Secondary DNS:



[Release Notes](#)


[Back](#) [Next](#)

Elejimos a zona horaria:



Introducimos unha clave de root:



 The root account is used for administering the system. Enter a password for the root user.

Root Password:

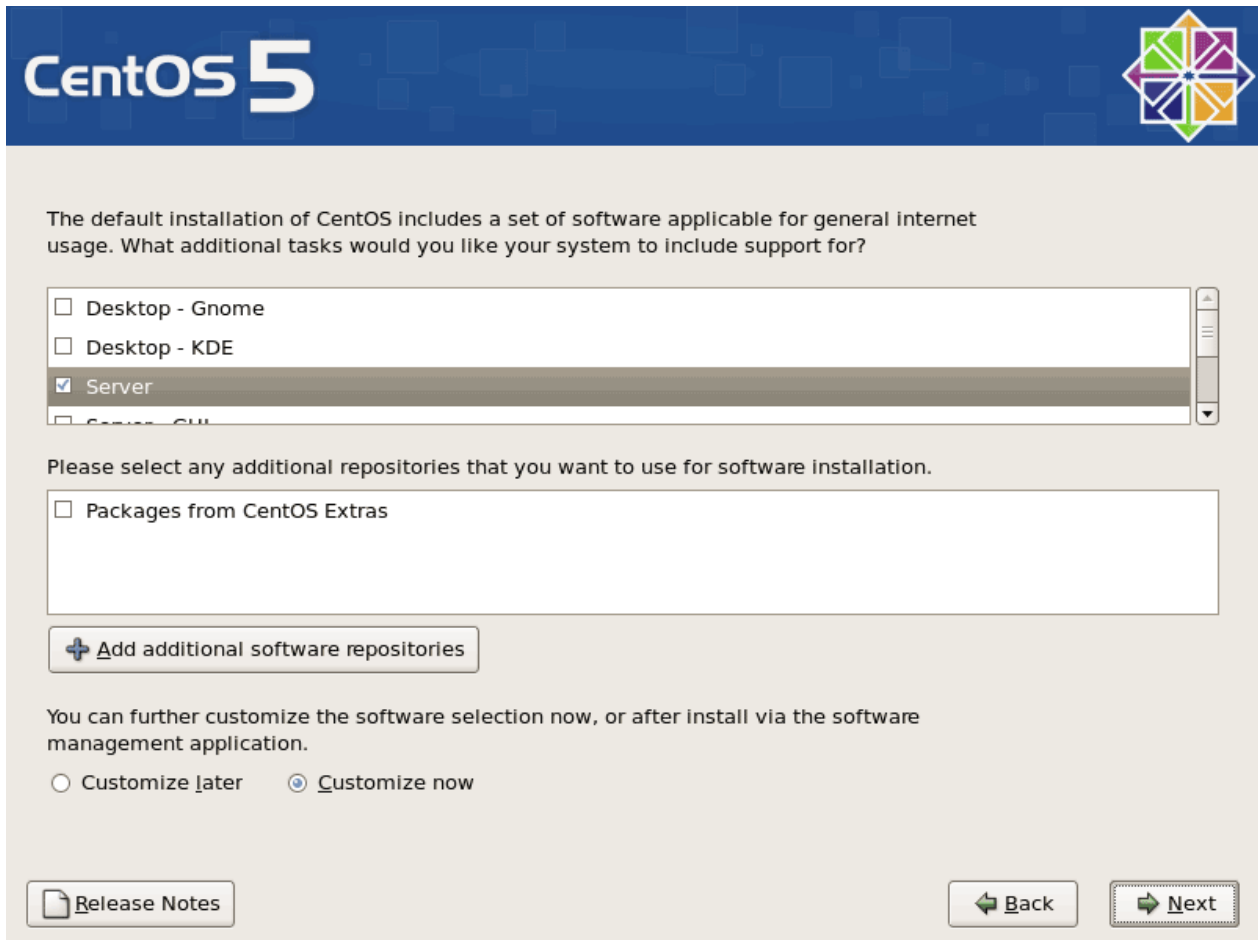
Confirm:

[Release Notes](#)

[Back](#) [Next](#)

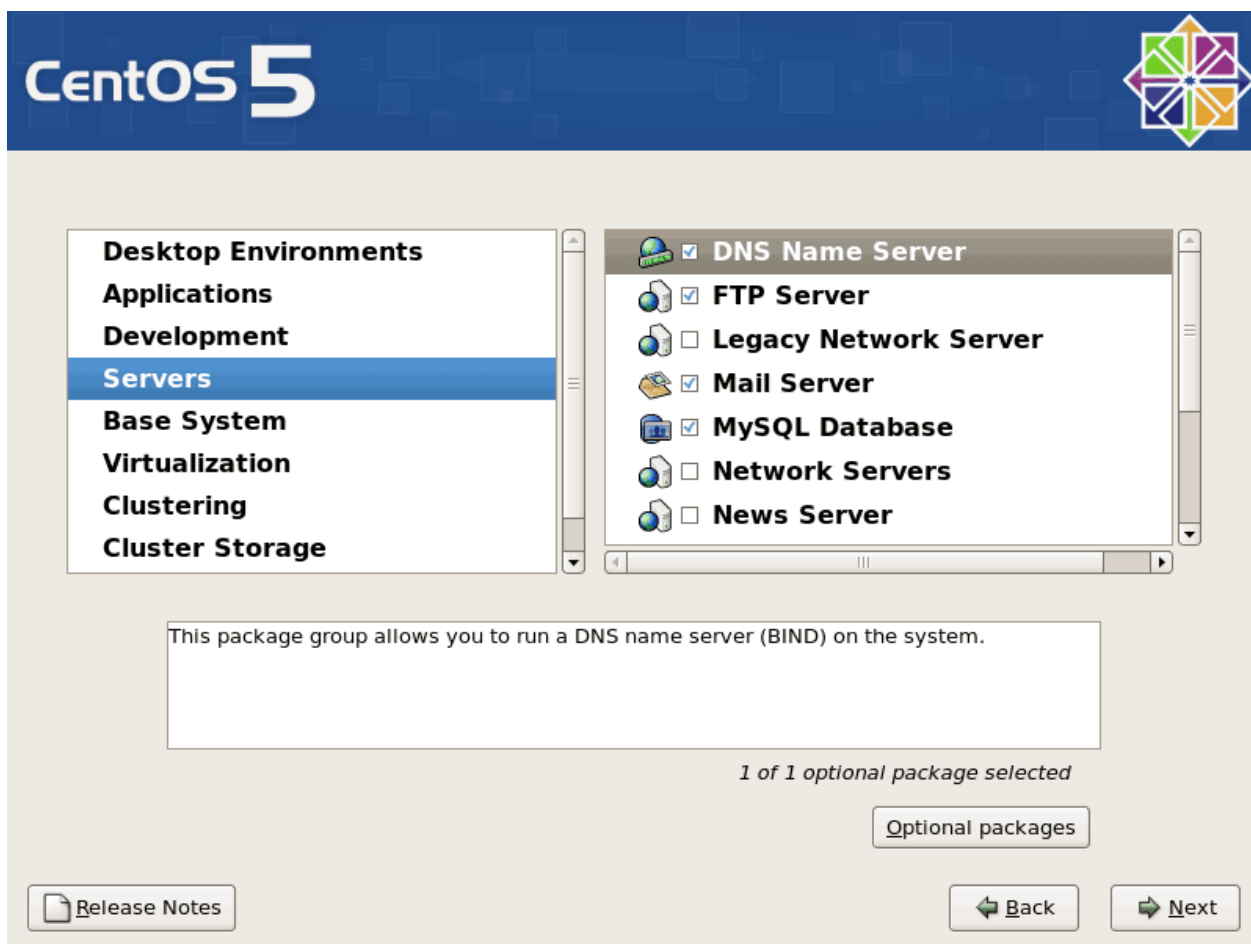
Instalación de CentOS 5.2 parte II

Ahora seleccionamos los paquetes de software que desecemos instalar. Desmarcamos todo e dejamos solamente la opción Server. Marcamos a continuación la opción de Customize Now e pulsamos en Next:



The screenshot shows the CentOS 5 installation interface. At the top, there is a blue header with the "CentOS 5" logo on the left and a colorful geometric logo on the right. Below the header, the text reads: "The default installation of CentOS includes a set of software applicable for general internet usage. What additional tasks would you like your system to include support for?". There is a list of options with checkboxes: "Desktop - Gnome", "Desktop - KDE", "Server" (which is checked and highlighted), and "Server - GUI". Below this list, the text says: "Please select any additional repositories that you want to use for software installation." There is a checkbox for "Packages from CentOS Extras" which is unchecked. Below that is a button labeled "+ Add additional software repositories". Further down, the text says: "You can further customize the software selection now, or after install via the software management application." There are two radio buttons: "Customize later" (unchecked) and "Customize now" (checked). At the bottom, there are three buttons: "Release Notes" (with a document icon), "Back" (with a left arrow icon), and "Next" (with a right arrow icon).

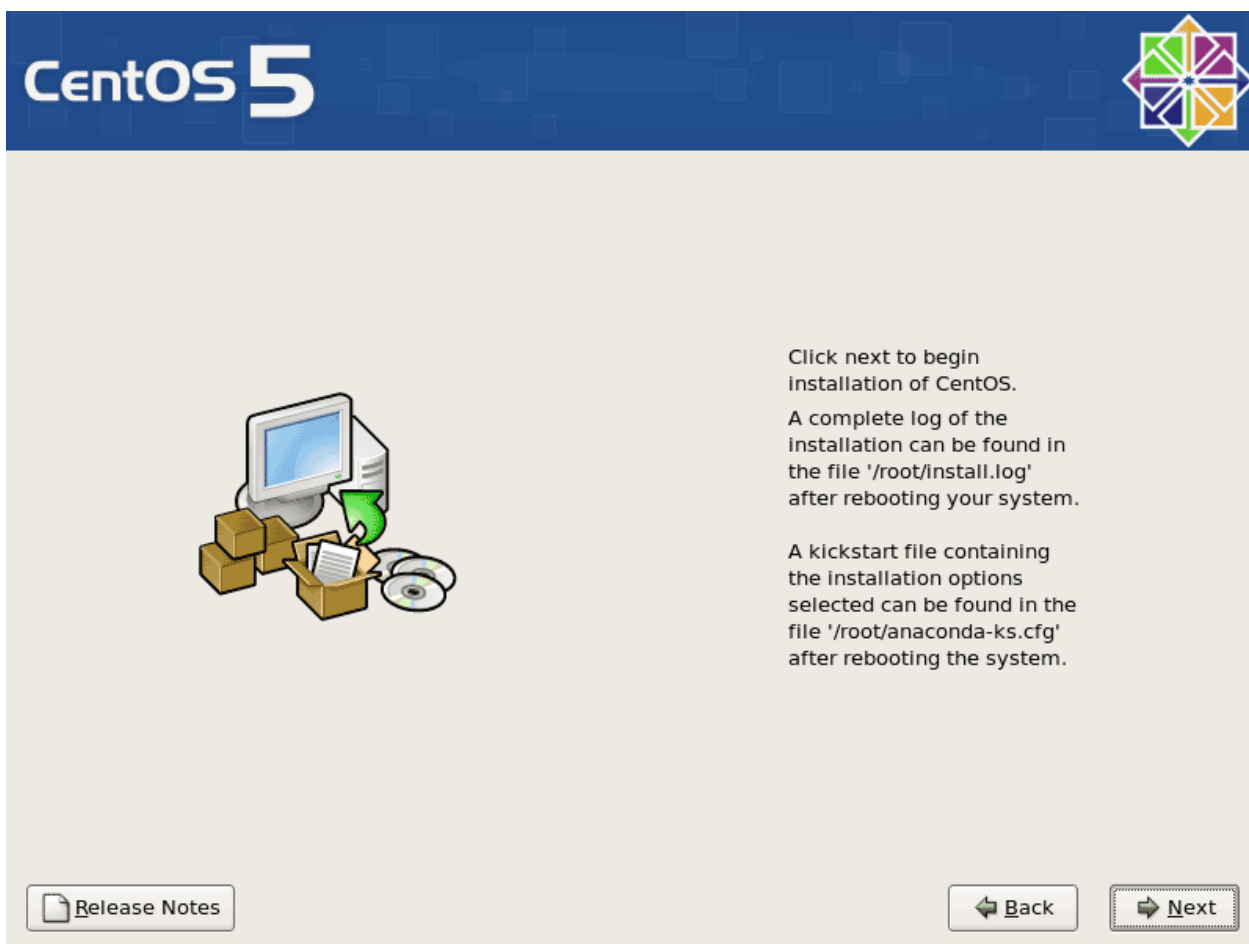
Ahora deberemos seleccionar el grupo de paquetes que desecemos instalar. Seleccionamos Editors, Text-based Internet, Development Libraries, Development Tools, DNS Name Server, FTP Server, Mail Server, MySQL Database, Server Configuration Tools, Web Server, Administration Tools, Base, e System Tools (desmarcamos el resto de paquetes) e pulsamos en Next:



O instalador chequeará as dependencias entre os diferentes paquetes:



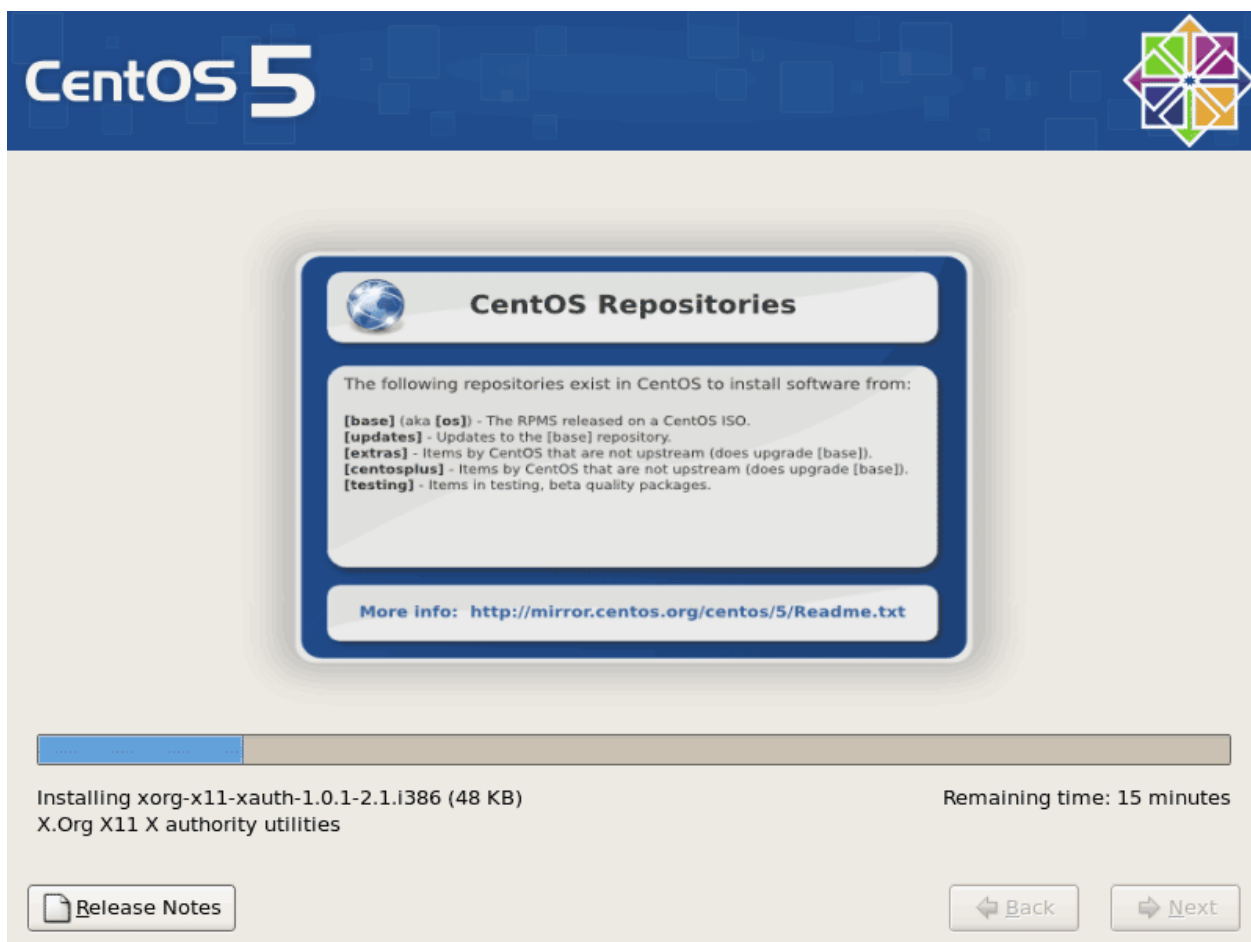
Pulsamos en Next para comenzar a instalación:



O disco duro está sendo formateado:



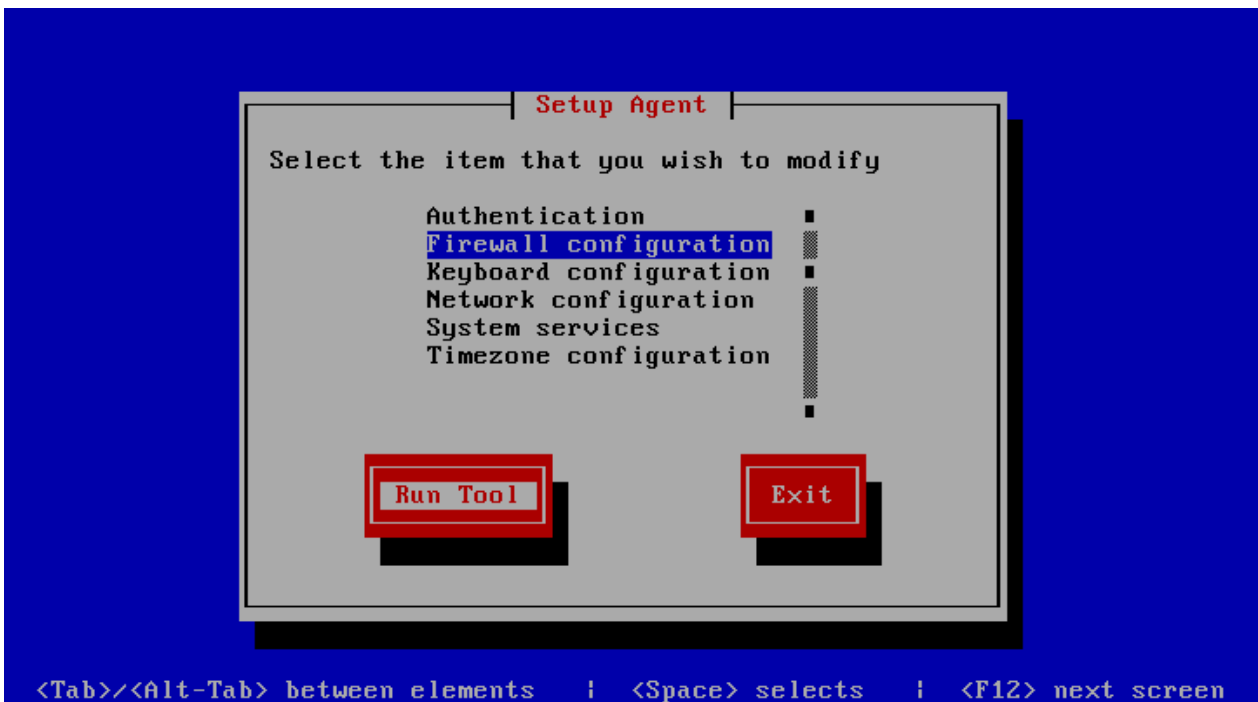
A instalación comenza, tardará uns minutos:



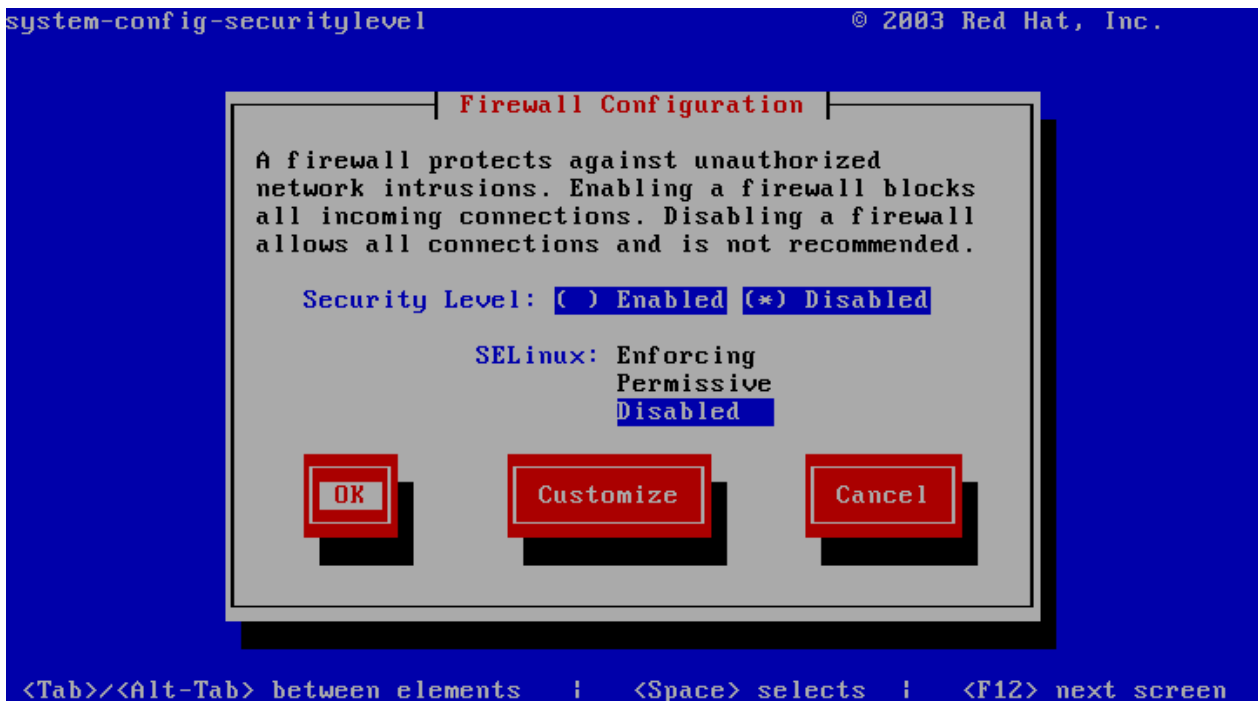
Finalmente, a instalación está completada e xa podemos sacar o CD ou DVD de ordenador e reiniciar:



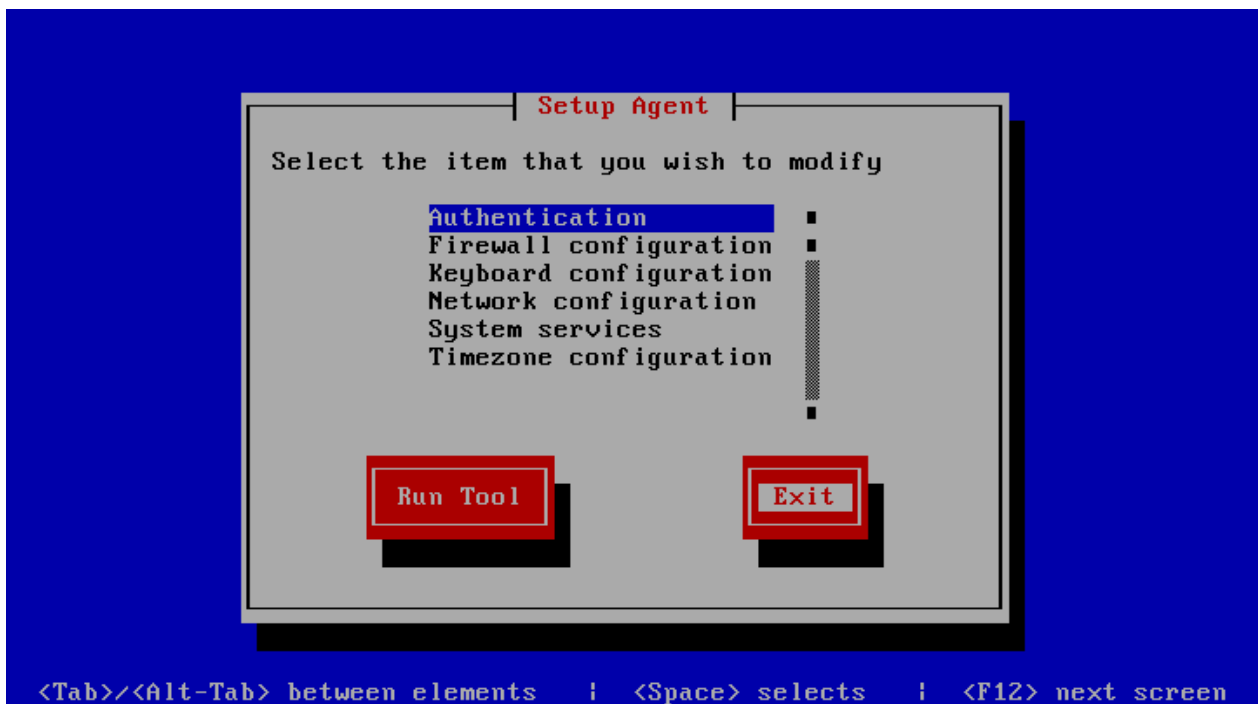
Despós do reboot veremos a seguinte pantalla. Seleccionaremos configuración do Firewall e executaremos a ferramenta:



Deshabilitaremos o firewall do CentOS e tamén SELinux o cal proporciona seguridade estendida.



Pulsaremos Exit para salir do axente de configuración:



A continuación entramos como root e reiniciaremos o sistema para que se apliquen os cambios:

```
reboot
```

Imos agora á configuración....

Instalación de CentOS 5.2 parte III

Axustamos /etc/hosts

Editamos o arquivo /etc/hosts para que sexa semellante ó seguinte:

```
vi /etc/hosts
```

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
192.168.0.100     server1.example.com server1
::1               localhost6.localdomain6 localhost6
```

Configuramos direccións IP adicionais.

(Esta sección é opcional. Soamente amosa como engadir direccións IP adicionais á interface de rede eth0 no caso que necesitamos máis de unha dirección IP. Si non necesitamos máis direccións IP podemos saltar esta sección.)

Asumimos que a nosa interface de rede é eth0. Polo tanto hai un arquivo /etc/sysconfig/network-scripts/ifcfg-eth0 o cal contén as configuracións para eth0. Podemos usar este arquivo para configurar unha nova interface de rede virtual eth0:0:

```
cp /etc/sysconfig/network-scripts/ifcfg-eth0
/etc/sysconfig/network-scripts/ifcfg-eth0:0
```

Agora queremos usar a dirección IP 192.168.0.101 na interface virtual eth0:0. Polo tanto abriremos ese arquivo /etc/sysconfig/network-scripts/ifcfg-eth0:0 e o modificamos do seguinte xeito (podemos deixar o apartado HWADDR xa que é a mesma dirección física da tarxeta):

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0:0
```

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0:0
BOOTPROTO=static
BROADCAST=192.168.0.255
IPADDR=192.168.0.101
NETMASK=255.255.255.0
NETWORK=192.168.0.0
ONBOOT=yes
```

Despois disto temos que reiniciar os servicios de rede:

```
/etc/init.d/network restart
```

Tamén podemos axustar o arquivo /etc/hosts despois de engadir a nova dirección IP, aínda que isto non é necesario.

Agora executamos

```
ifconfig
```

Deberíamos ver a nova dirección IP na saída:

```
[root@server1 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:B1:97:E1
          inet addr:192.168.0.100  Bcast:192.168.0.255
Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb1:97e1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:310 errors:0 dropped:0 overruns:0 frame:0
          TX packets:337 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:28475 (27.8 KiB)  TX bytes:72116 (70.4 KiB)
          Interrupt:177 Base address:0x1400

eth0:0    Link encap:Ethernet  HWaddr 00:0C:29:B1:97:E1
          inet addr:192.168.0.101  Bcast:192.168.0.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:177 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:560 (560.0 b)  TX bytes:560 (560.0 b)

[root@server1 ~]#
```

Instalación de algún software

Primeiro importamos as claves CPG para os paquetes de software:

```
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY*
```

A continuación actualizamos os paquetes actuais do sistema:

```
yum update
```

Instalamos os seguintes paquetes que necesitaremos máis adiante:

```
yum install fetchmail wget bzip2 unzip zip nmap openssl lynx fileutils
ncftp gcc gcc-c++
```

Install A Chrooted DNS Server (BIND9)

Para instalar BIND9 chrooted faremos o seguinte:

```
yum install bind-chroot
```

A continuación:

```

chmod 755 /var/named/
chmod 775 /var/named/chroot/
chmod 775 /var/named/chroot/var/
chmod 775 /var/named/chroot/var/named/
chmod 775 /var/named/chroot/var/run/
chmod 777 /var/named/chroot/var/run/named/
cd /var/named/chroot/var/named/
ln -s ../../ chroot
cp /usr/share/doc/bind-9.3.4/sample/var/named/named.local
/var/named/chroot/var/named/named.local
cp /usr/share/doc/bind-9.3.4/sample/var/named/named.root
/var/named/chroot/var/named/named.root
touch /var/named/chroot/etc/named.conf
chkconfig --levels 235 named on
/etc/init.d/named start

```

BIND se ejecutará a continuación bajo `/var/named/chroot/var/named/`.

MySQL (5.0)

Para instalar MySQL faremos o siguiente:

```

yum install mysql mysql-devel mysql-server

```

A continuación creamos o enlace para que arranque có sistema:

```

chkconfig --levels 235 mysqld on
/etc/init.d/mysqld start

```

Ahora chequeamos que a rede está habilitada e executamos:

```

netstat -tap | grep mysql
</source>

```

E debería aparecer algo como:

```

<source lang=bash>
[root@server1 ~]# netstat -tap | grep mysql
tcp        0      0 *:mysql      *:*          LISTEN     2584/mysqld
[root@server1 ~]#

```

Si non aparece, editaremos `/etc/my.cnf` e comentamos a opción `skip-networking`:

```

vi /etc/my.cnf

[...]
#skip-networking
[...]

```

e reiniciamos o servidor de MySQL:

```
/etc/init.d/mysqld restart
```

Executamos:

```
mysqladmin -u root password yourrootsqlpassword  
mysqladmin -h server1.example.com -u root password yourrootsqlpassword
```

para modificar a password do user root (noutro caso calqueira podería acceder á base de datos MySQL).

Instalación de CentOS 5.2 parte IV

Apache2 con PHP

Agora instalamos Apache con PHP (PHP 5.1.6):

```
yum install php php-devel php-gd php-imap php-ldap php-mysql php-odbc  
php-pear php-xml php-xmlrpc curl curl-devel perl-libwww-perl  
ImageMagick libxml2 libxml2-devel
```

A continuación editamos /etc/httpd/conf/httpd.conf:

```
vi /etc/httpd/conf/httpd.conf
```

e modificamos DirectoryIndex a

```
[...]  
DirectoryIndex index.html index.htm index.shtml index.cgi index.php  
index.php3 index.pl  
[...]
```

Configuramos a continuación o sistema para que o Apache arranque no inicio:

```
chkconfig --levels 235 httpd on
```

Arrancamos Apache:

```
/etc/init.d/httpd start
```

Sincronizamos o reloxo do sistema

Si queremos deixar o reloxo do sistema sincronizado cun servidor de hora NTP faremos o seguinte:

```
yum install ntp  
  
chkconfig --levels 235 ntpd on  
ntpdate 0.pool.ntp.org  
/etc/init.d/ntpd start
```

Instalaremos algunos módulos Perl adicionales

```
yum install perl-HTML-Parser perl-DBI perl-Net-DNS perl-Digest-SHA1
```

Enlaces externos

[1] <http://www.subnetmask.info>

Fuente: http://informatica/manuais/index.php?title=Instalación_de_Servidor_Linux_con_CentOS&oldid=11441

Autores principales: Veiga,

Instalación e Configuración de Asterisk

Instalación e Configuración de Asterisk

1. → Descarga e instalación dos módulos de Asterisk
2. → Configuración de Asterisk
3. → Consola de Asterisk
4. → Comandos en Asterisk

Consideraciones sobre archivos de configuración

1. Archivo chan_dahdi.conf
 2. → Archivo dahdi-channels.conf
 3. → Archivo extensions.conf
 4. → Archivo iax.conf
 5. → Archivo indications.conf
 6. → Archivo meetme.conf
 7. → Archivo musiconhold.conf
 8. → Archivo queues.conf
 9. Archivo res_ldap.conf
 10. → Archivo sip.conf
 11. → Archivo system.conf
 12. → Archivo voicemail.conf
1. → Comparación de CODECS de audio

Autores principais: Veiga,

Configuración de Asterisk

Carpeta configuracións de Asterisk

Os ficheiros de configuración de Asterisk atópanse na seguinte carpeta:

```
/etc/asterisk
```

Configuración de Asterisk e NAT

Configuración de SIP

Se temos instalado o Asterisk nun servidor nunha rede privada que use NAT, teremos que configurar o seguinte para o correcto funcionamento de SIP:

1. No ficheiro `/etc/asterisk/sip.conf` :

```
[general]
nat=yes
externip = X.X.X.X
localnet = 192.168.X.0/255.255.255.0
qualify=yes

# En externip podemos por a ip externa ou o seu FQDN
# (Fully Qualified Domain Name -- nome do dominio).
```

2. No router abriremos os seguintes portos á ip do servidor asterisk:

```
UDP: 5060
UDP: 10000-20000
```

Máis información en: <http://www.voip-info.org/wiki-Asterisk+SIP+NAT+solutions>

Configuración de IAX

Para que o protocolo IAX funcione correctamente simplemente teremos que redireccionar o seguinte porto no router:

```
UDP: 4569
```

Ficheiros de Configuración

1. `/etc/asterisk/indications.conf`
2. `/etc/asterisk/modules.conf`
3. `/etc/asterisk/sip.conf`
4. `/etc/asterisk/extensions.conf`

Documentación adicional

```
/usr/src/asterisk/asterisk-1.6.0.5/doc
```

Fuente: http://informatica/manuais/index.php?title=Configuración_de_Asterisk&oldid=8395

Autores principales: Veiga,

Consola de Asterisk

Acceso á consola

Poderemos acceder á consola de asterisk mediante o seguinte comando, sempre e cando o asterisk esté executándose, é dicir que arranque de forma automática có sistema operativo:

```
asterisk -rvvvv
```

Cantas máis 'v' poñamos, máis nivel de detalle obteremos nas mensaxes do Asterisk.

Arranque manual de Asterisk

No caso de non estar configurado para arranque automático, é posible executar asterisk en consola mediante o seguinte comando:

```
asterisk -c
```

ou ben

```
asterisk -cvvvv
```

#para que amose máis información acerca do funcionamento.

Recarga de Asterisk

Para recargar Asterisk dende a liña de comandos poremos:

```
asterisk -r -x reload
```

Asterisk funcionando como un servizo

Si cando arrancamos o servidor xa temos funcionando Asterisk, poderemos xestionar ese servizo cos seguintes comandos:

```
service asterisk start    Iniciar servizo.
service asterisk stop    Detener servizo.
service asterisk status   Obter estado do servizo.
service asterisk restart  Reiniciar servizo.
```

Fuente: http://informatica/manuais/index.php?title=Consola_de_Asterisk&oldid=8352

Autores principales: Veiga,

Comandos en Asterisk

Guía Rápida de Comandos de Asterisk

Descargar versión pdf aquí: [Media:Guia-Rapida-Asterisk.pdf](#)

INICIAR Y DETENER ASTERISK

asterisk	Arrancar Asterisk.
asterisk -c	Iniciar Asterisk y abrir la consola remota
asterisk -r	Ingresar a la consola remota
asterisk -rx 'comando'	Ejecutar comando sin ingresar a la consola.
stop now	Detener el servicio Asterisk desde la consola remota.
stop gracefully	Detener el servicio Asterisk cuando todos los canales activos sean completados.
stop when convenient	Detener el servicio Asterisk cuando todos los canales activos sean completados. Nuevas llamadas son aceptadas. El sistema esperará hasta que no exista actividad.
exit	Salir de la consola remota. No detiene el servicio Asterisk.

DIRECTORIOS Y ARCHIVOS

/etc/asterisk/	Contiene todos los archivos de configuración.
/etc/asterisk/extensions.conf	Archivo de configuración del dialplan.
/etc/asterisk/sip.conf	Archivo de configuración de los canales SIP.
/etc/asterisk/iax.conf	Archivo de configuración de los canales IAX.
/etc/asterisk/voicemail.conf	Archivo de configuración del sistema de correo de voz.
/usr/lib/asterisk/modules	Contiene las librerías, aplicaciones y módulos.
/var/run/asterisk	Contiene datos de los procesos.
/var/lib/asterisk/sounds/	Contiene los archivos de sonido.
/var/log/asterisk/	Contiene los archivos de bitácora.
/var/spool/asterisk	Contiene los archivos de llamadas.

COMANDOS IMPORTANTES CLI

set verbose 5	
set debug	
sip show peers	
sip show channels	

AYUDA DENTRO DEL CLI

help	
help sip	
help sip show	
help sip show peers	

REGEX EN EXTENSIONS.CONF

X	0-9
Z	1-9
N	2-9
[5-7]	5, 6 y 7
[15-7]	1, 5, 6 y 7
.	Cualquier carácter o dígito

SIP.CONF

[general]	Sección de configuración global
port=5060	Asterisk escucha puerto 5060
bindaddr=0.0.0.0	Escucha peticiones de todos los interfaces.
[200]	Configura el canal SIP 200 (puede ser alfanumérico).
type=friend	Configura el tipo de canal. Asterisk <= user, Asterisk => peer, Asterisk = friend.
username=200	Usuario
secret=123	Contraseña
host=dynamic	Define la dirección IP para el dispositivo SIP. dynamic = todos los IPs.
context=internos	Contexto el cual es iniciado cuando el dispositivo abre el canal.
qualify=yes	Monitoreo de latencia.
nat=yes	Soporte para NAT.

[general]

static=yes

writeprotec

[globals]

VAR2=23

[nombre]

setGlobal(V

set(VAR3=

dial(tech/u

answer()

hangup()

wait(n)

goto(n)

gotoif(\$[\$

gotoitime(\$

fri|1-31|*?d

saynumber

sayalpha()

saydigits()

sayunixtim

voicemail()

voicemailm

Autores principales: Veiga,

Arquivo chan dahdi.conf

Téñase en conta que Asterisk non lé todos os arquivos de configuración xerados por **DAHDI**, se non soamente o ficheiro **chan_dahdi.conf**. Para que Asterisk interprete a configuración xerada, agregar a ese ficheiro a liña:

```
#include dahdi-channels.conf
```

O arquivo *chan-dahdi.conf* que resulta despois de adaptalo é:

```
[channels]
#include dahdi-channels.conf
;General options
; Default language
language=es
usecallerid=yes
hidecallerid=no
callwaiting=yes
threewaycalling=yes
transfer=yes
echocancel=yes
echocancelwhenbridged=yes
rxgain=0.0
txgain=0.0
;FXO Modules
echocancel=yes
signalling=fxs_ks
context=from_pstn
channel=>1
```

Fuente: http://informatica/manuais/index.php?title=Arquivo_chan_dahdi.conf&oldid=12094

Autores principales: Veiga,

Arquivo dahdi-channels.conf

O ficheiro *dahdi-channels.conf* é autoxenerado polo script *dahdi_genconf* antes executado, así este ficheiro quedaría do seguinte xeito:

```
; Autogenerated by /usr/sbin/dahdi_genconf on Tue May 19 13:07:48 2009
; If you edit this file and execute /usr/sbin/dahdi_genconf again,
; your manual changes will be LOST.
; Dahdi Channels Configurations (chan_dahdi.conf)
;
```

```
; This is not intended to be a complete chan_dahdi.conf. Rather, it is
intended
; to be included by chan_dahdi.conf that will include the global
settings
;
; Span 1: WCTDM/0 "Wildcard AEX410 Board 1" (MASTER)
;;; line="1 WCTDM/0/0 FXSKS (EC: MG2)"
signalling=fxs_ks
callerid=asreceived
context=from_pstn
channel => 1
```

Fuente: http://informatica/manuais/index.php?title=Arquivo_dahdi_channels.conf&oldid=12096

Autores principales: Veiga,

Arquivo extensions.conf

```
; extensions.conf - the Asterisk dial plan
;
; Static extension configuration file, used by
; the pbx_config module. This is where you configure all your
; inbound and outbound calls in Asterisk.
;
; This configuration file is reloaded
; - With the "dialplan reload" command in the CLI
; - With the "reload" command (that reloads everything) in the CLI
;
; The "General" category is for certain variables.
;
[general]
static=yes
writeprotect=no
clearglobalvars=no

[buzonvoz]
; Acceso a buzón de voz con extensión + contraseña
exten => 8000,1,VoicemailMain

; Acceso directo a buzón de voz solicitando solamente contraseña.
exten => 8001,1,VoicemailMain(${CALLERID(num)}@default)

[latencia]
```

```
; Extension para evaluar latencias.
exten => 8100,1,Playback(demo-echotest) ; Let them know what's going on
exten => 8100,n,Echo ; Do the echo test
exten => 8100,n,Playback(demo-echodone) ; Let them know it's over
exten => 8100,n,Goto(s,6) ; Start over

[conferencias]
exten => 8500,1,MeetMe(100,iM)
exten => 8500,1,Playback(conf-thereare)
exten => 8500,n,MeetMeCount(100)
exten => 8500,n,Playback(conf-peopleinconf)

[macro-extension]
;esta macro recibira dous parámetros
;o primero indica o dispositivo ó que hai que chamar
;o segundo e o número do buzón de voz ó que desviar a chamada
exten => s,1,NoOp(chamada para ${ARG1})
exten => s,n,Dial(${ARG1},20)
exten => s,n,Goto(s-${DIALSTATUS},1)

; si esta ocupado, reproducése a mensaxe correspondente
exten => s-BUSY,1,VoiceMail(${ARG2},b)
exten => s-BUSY,n,Hangup

; si non contesta, reproducése a mensaxe de "non dispoñible"
exten => s-NOANSWER,1,VoiceMail(${ARG2},u)
exten => s-NOANSWER,n,Hangup

; en calqueira outro caso, facemos o mesmo que si non contesta.
exten => s-.,1,Goto(s-NOANSWER,1)

;-----
; Ramiro
; Grabar sonidos para IVR
;-----
[grabacion]
exten => 205,1,Wait(2)
exten => 205,2,Record(/tmp/asterisk-recording:gsm)
exten => 205,3,Wait(2)
exten => 205,4,Playback(/tmp/asterisk-recording)
exten => 205,5,wait(2)
exten => 205,6,Hangup

[deeper]
exten => s,1,Background(dipa)
exten => s,2,Goto(secretaria,s,1)
```



```
[secretaria]
exten => s,1,Background(secretaria)
exten => 1,1,Goto(deeper,s,1)
exten => 2,1,Goto(informatica,s,1)

[informatica]
exten => s,1,Background(informatica)
exten => s,2,Dial(SIP/7000)
exten => s,3,Hangup()
exten => 0,1,Goto(ivr,100)
exten => 1,1,Goto(Menu,s,1)

[Menu]
exten => s,1,Set(CHANNEL(musicclass)=default);
exten => s,2,Wait(3) ;espera 3 segundos
exten => s,3,Set(TIMEOUT(digit)=5) ; 5 segundos es el tiempo que espera
entre el primer digito y los sucesivos
exten => s,4,Set(TIMEOUT(response)=10) ; 10 segundos es el tiempo que
espera para que el llamante presione una tecla

exten => s,5,BackGround(introduccion-menu-ies) ; presenta el menu vocal y
al mismo tiempo escucha si el llamante presiona alguna tecla
exten => s,6,WaitExten() ; espera que el llamante presione alguna tecla

exten => 1,1,Goto(informatica,s,1)
exten => 2,1,Goto(secretaria,s,1)
exten => i,1,Playback(invalid)
exten => i,2,Playback(goodbye)
exten => i,3,Hangup

exten => t,1,Goto(Menu,s,2) ;si dentro de 10 segundo el llamante no
presiona ninguna tecla vuelve a presentar el menu vocal
exten => h,1,Hangup ; si el llamante cuelga ejecuta la extension h

[ivr]
exten => 100,1,Answer
exten => 100,2,Goto(Menu,s,1)

;-----

[provinciales]

[internacionales]

[moviles]
```

```
[exteriores]

[personal]
include => buzonvoz
include => latencia
include => conferencias
include => colas

include => grabacion
include => ivr

exten => 5000,1,Macro(extension,SIP/5000,5000)
exten => 5001,1,Macro(extension,SIP/5001,5001)
exten => 5002,1,Macro(extension,SIP/5002,5002)
exten => 5003,1,Macro(extension,SIP/5003,5003)
exten => 5004,1,Macro(extension,SIP/5004,5004)
exten => 5005,1,Macro(extension,SIP/5005,5005)
exten => 5006,1,Macro(extension,SIP/5006,5006)
exten => 5007,1,Macro(extension,SIP/5007,5007)

exten => 6000,1,Macro(extension,IAX2/6000,5000)
exten => 6001,1,Macro(extension,IAX2/6001,5001)
exten => 6002,1,Macro(extension,IAX2/6002,5002)
exten => 6003,1,Macro(extension,IAX2/6003,5003)
exten => 6004,1,Macro(extension,IAX2/6004,5004)
exten => 6005,1,Macro(extension,IAX2/6005,5005)
exten => 6006,1,Macro(extension,IAX2/6006,5006)
exten => 6007,1,Macro(extension,IAX2/6007,5007)

exten => 7000,1,Macro(extension,SIP/7000,7000)
exten => 0000,1,Macro(extension,SIP/0000,0000)

exten => 3333,1,Background(agent-pass)

[colas]
exten=> 8600,1,Answer
exten=> 8600,2,Set(CHANNEL(musicclass)=blues)
exten=> 8600,3,DigitTimeout,5
exten=> 8600,4,ResponseTimeout,10
exten=> 8600,5,Backgroun(welcome)
exten=> 8600,6,Queue(administradores,tT)
exten=> 8600,7,Hangup
```

```
exten=> 8601,1,Answer
exten=> 8601,2,Set(CHANNEL(musicclass)=blues)
exten=> 8601,3,MusicOnHold

;-----
; Configuracion conexion LDAP: Ramiro
;-----

[users]
switch => Realtime/@

[demo]
switch => Realtime/@

[default]
include => users
include => demo
;-----

[from-pstn]
exten => s,1,Answer
exten => s,n,VoiceMail(5000)
```

Fuente: http://informatica/manuais/index.php?title=Arquivo_extensions.conf&oldid=12059

Autores principales: Veiga,

Arquivo iax.conf

```
;
; IAX Archivo de Configuración
;
; IAX dial strings

[general]
bandwidth=low
jitterbuffer=no
forcejitterbuffer=no
tos=lowdelay
autokill=yes
disallow=all
allow=alaw
allow=ulaw
allow=gsm
language=es
canreinvite=yes
```

```
qualify=yes
nat=yes

; Sección de Profesorado

[6000]
type=friend
secret=hellopeople
callerid="IeSC Veiga" <6000>
mailbox=6000@default
context=personal
host=dynamic

[6001]
type=friend
secret=hellopeople
callerid="IeSC Vieites" <6001>
mailbox=6001@default
context=personal
host=dynamic
```

Fuente: http://informatica/manuais/index.php?title=Arquivo_iax.conf&oldid=12057

Autores principales: Veiga,

Arquivo indications.conf

```
; indications.conf
; Configuration file for location specific tone indications
; used by the pbx_indications module.
;
; NOTE:
;   When adding countries to this file, please keep them in
alphabetical
;   order according to the 2-character country codes!
;
; The [general] category is for certain global variables.
; All other categories are interpreted as location specific indications
;
;
[general]
country=es           ; default location
.....
```

Fuente: http://informatica/manuais/index.php?title=Arquivo_indications.conf&oldid=12061

Autores principales: Veiga,

Arquivo meetme.conf

```
;
; Configuration file for MeetMe simple conference rooms for Asterisk of
; course.
;
; This configuration file is read every time you call app meetme()

[general]
;audiobuffers=32      ; The number of 20ms audio buffers to be used
;                      ; when feeding audio frames from non-DAHDI channels
;                      ; into the conference; larger numbers will allow
;                      ; for the conference to 'de-jitter' audio that
arrives
;                      ; at different timing than the conference timing
;                      ; source, but can also allow for latency in hearing
;                      ; the audio from the speaker. Minimum value is 2,
;                      ; maximum value is 32.
;
; Conferences may be scheduled from realtime?
;schedule=yes
;
; Update realtime when members login/out of the conference
;logmembercount=no
;
; How much earlier than the start time should we allow participants to
; join the conference (in seconds)?
;fuzzystart=300
;
; If the participants join too early, how much time should we allow
; to tell them that they have joined too early, rather than telling
; them
; the conference simply doesnt exist (in seconds)?
;earlyalert=3600
;
; How many seconds before the scheduled end of the conference should
; the participants be warned?
;endalert=120
;
[rooms]
;
; Usage is conf => confno[,pin][,adminpin]
```

```
;
; Note that once a participant has called the conference, a change to
the pin
; number done in this file will not take effect until there are no more
users
; in the conference and it goes away. When it is created again, it
will have
; the new pin number.
;

conf => 100
```

Fuente: http://informatica/manuais/index.php?title=Arquivo_meetme.conf&oldid=12067

Autores principales: Veiga,

Arquivo musiconhold.conf

```
;
; Music on Hold -- Sample Configuration
;
[general]
;cachertclasses=yes ; use 1 instance of moh class for all users who are
using it,
; decrease consumable cpu cycles and memory
; disabled by default

; valid mode options:
; files -- read files from a directory in any Asterisk
supported
; media format
; quietmp3 -- default
; mp3 -- loud
; mp3nb -- unbuffered
; quietmp3nb -- quiet unbuffered
; custom -- run a custom application (See examples below)

; =====
; File-based (native) music on hold
; =====
;
; This plays files directly from the specified directory, no external
; processes are required. Files are played in normal sorting order
```

```
; (same as a sorted directory listing), and no volume or other
; sound adjustments are available. If the file is available in
; the same format as the channel's codec, then it will be played
; without transcoding (same as Playback would do in the dialplan).
; Files can be present in as many formats as you wish, and the
; 'best' format will be chosen at playback time.
;
; NOTE:
; If you are not using "autoload" in modules.conf, then you
; must ensure that the format modules for any formats you wish
; to use are loaded _before_ res_musiconhold. If you do not do
; this, res_musiconhold will skip the files it is not able to
; understand when it loads.
;

[default]
mode=files
directory=/var/lib/asterisk/moh

[blues]
mode=files
directory=/var/lib/asterisk/moh/blues

[americanas]
mode=files
directory=/var/lib/asterisk/moh/americanas

[pop]
mode=files
directory=/var/lib/asterisk/moh/pop

;
;[native-random]
;mode=files
;directory=/var/lib/asterisk/moh
;digit=#          ; If this option is set for a class, then when callers
are
;                  ; listening to music on hold, they can press this
digit, and
;                  ; they will switch to listening to this music class.
;sort=random      ; Sort the files in random order

;[native-alphabetical]
;mode=files
;directory=/var/lib/asterisk/moh
;sort=alpha       ; Sort the files in alphabetical order. If this option
is
```

```
;           ; not specified, the sort order is undefined.

; =====
; Other (non-native) playback methods
; =====

;[manual]
;mode=custom
; Note that with mode=custom, a directory is not required, such as when
  reading
; from a stream.
;directory=/var/lib/asterisk/mohmp3
;application=/usr/bin/mpg123 -q -r 8000 -f 8192 -b 2048 --mono -s

;[ulawstream]
;mode=custom
;application=/usr/bin/streamplayer 192.168.100.52 888
;format=ulaw

; mpg123 on Solaris does not always exit properly; madplay may be a
better
; choice
;[solaris]
;mode=custom
;directory=/var/lib/asterisk/mohmp3
;application=/site/sw/bin/madplay -Q -o raw:- --mono -R 8000 -a -12
```

Fuente: http://informatica/manuais/index.php?title=Arquivo_musiconhold.conf&oldid=12071

Autores principales: Veiga,

Arquivo queues.conf

```
[general]
;
; Global settings for call queues
;
; Persistent Members
;   Store each dynamic member in each queue in the astdb so that
;   when asterisk is restarted, each member will be automatically
;   read into their recorded queues. Default is 'yes'.
;
persistentmembers = yes
```



```
;
; Keep Stats
;   Keep queue statistics during a reload. Default is 'no'
;
keepstats = no

;
; AutoFill Behavior
;   The old/current behavior of the queue has a serial type behavior
;   in that the queue will make all waiting callers wait in the queue
;   even if there is more than one available member ready to take
;   calls until the head caller is connected with the member they
;   were trying to get to. The next waiting caller in line then
;   becomes the head caller, and they are then connected with the
;   next available member and all available members and waiting
callers
;   waits while this happens. The new behavior, enabled by setting
;   autofill=yes makes sure that when the waiting callers are
connecting
;   with available members in a parallel fashion until there are
;   no more available members or no more waiting callers. This is
;   probably more along the lines of how a queue should work and
;   in most cases, you will want to enable this behavior. If you
;   do not specify or comment out this option, it will default to no
;   to keep backward compatibility with the old behavior.
;
autofill = yes
;
; Monitor Type
;   By setting monitor-type = MixMonitor, when specifying
monitor-format
;   to enable recording of queue member conversations, app_queue will
;   now use the new MixMonitor application instead of Monitor so
;   the concept of "joining/mixing" the in/out files now goes away
;   when this is enabled. You can set the default type for all queues
;   here, and then also change monitor-type for individual queues
within
;   queue by using the same configuration parameter within a queue
;   configuration block. If you do not specify or comment out this
option,
;   it will default to the old 'Monitor' behavior to keep backward
;   compatibility.
;
monitor-type = MixMonitor
;
```

```
; UpdateCDR behavior.
; This option is implemented to mimic chan_agents behavior of
populating
; CDR dstchannel field of a call with an agent name, which you can
set
; at the login time with AddQueueMember membername parameter.
;
; updatecdr = no

;
; Note that a timeout to fail out of a queue may be passed as part of
; an application call from extensions.conf:
; Queue(queueename,[options],[optionalurl],[announceoverride],[timeout])
; example: Queue(dave,t,,45)

; shared_lastcall will make the lastcall and calls received be the same
in
; members logged in more than one queue.
; This is useful to make the queue respect the wrapuptime of another
queue
; for a shared member
;
shared_lastcall=no
;
;[markq]
;
; A sample call queue
;
; Musicclass sets which music applies for this particular call queue.
; The only class which can override this one is if the MOH class is set
; directly on the channel using Set(CHANNEL(musicclass)=whatever) in
the
; dialplan.
;
;musicclass = default
;
; An announcement may be specified which is played for the member as
; soon as they answer a call, typically to indicate to them which queue
; this call should be answered as, so that agents or members who are
; listening to more than one queue can differentiated how they should
; engage the customer
;
;announce = queue-markq
;
; A strategy may be specified. Valid strategies include:
;
; ringall - ring all available channels until one answers (default)
```

```
; leastrecent - ring interface which was least recently called by this
queue
; fewestcalls - ring the one with fewest completed calls from this
queue
; random - ring random interface
; rrmemory - round robin with memory, remember where we left off last
ring pass
; linear - rings interfaces in the order specified in this
configuration file.
;         If you use dynamic members, the members will be rung in the
order in
;         which they were added
; wrandom - rings random interface, but uses the member's penalty as a
weight
;         when calculating their metric. So a member with penalty 0
will have
;         a metric somewhere between 0 and 1000, and a member with
penalty 1 will
;         have a metric between 0 and 2000, and a member with
penalty 2 will have
;         a metric between 0 and 3000. Please note, if using this
strategy, the member
;         penalty is not the same as when using other queue
strategies. It is ONLY used
;         as a weight for calculating metric.
;
;strategy = ringall
;
; Second settings for service level (default 0)
; Used for service level statistics (calls answered within service
level time
; frame)
;servicelevel = 60
;
; A context may be specified, in which if the user types a SINGLE
; digit extension while they are in the queue, they will be taken out
; of the queue and sent to that extension in this context.
;
;context = qoutcon
;
;-----QUEUE TIMING
OPTIONS-----
; A Queue has two different "timeout" values associated with it. One is
the
; timeout parameter configured in queues.conf. This timeout specifies
the
; amount of time to try ringing a member's phone before considering the
```

```
; member to be unavailable. The other timeout value is the second
argument
; to the Queue() application. This timeout represents the absolute
amount
; of time to allow a caller to stay in the queue before the caller is
; removed from the queue. In certain situations, these two timeout
values
; may clash. For instance, if the timeout in queues.conf is set to 5
seconds,
; the retry value in queues.conf is set to 4, and the second argument
to Queue()
; is 10, then the following may occur:
;
; A caller places a call to a queue.
; The queue selects a member and attempts to ring that member.
; The member's phone is rung for 5 seconds and he does not answer.
; The retry time of 4 seconds occurs.
; The queue selects a second member to call.
;
; How long does that second member's phone ring? Does it ring for 5
seconds
; since the timeout set in app_queue is 5 seconds? Does it ring for 1
second since
; the caller has been in the queue for 9 seconds and is supposed to be
removed after
; being in the queue for 10 seconds? This is configurable with the
timeoutpriority
; option. By setting the timeoutpriority to "conf" then you are saying
that you would
; rather use the time specified in the configuration file even if it
means having the
; caller stay in the queue longer than the time specified in the
application argument.
; For the scenario described above, timeoutpriority=conf would result
in the second
; member's phone ringing for 5 seconds. By specifying "app" as the
value for
; timeoutpriority, you are saying that the timeout specified as the
argument to the
; Queue application is more important. In the scenario above,
timeoutpriority=app
; would result in the second member's phone ringing for 1 second.
;
; There are a few exceptions to the priority rules. For instance, if
timeoutpriority=app
; and the configuration file timeout is set to 0, but the application
argument timeout is
```

```
; non-zero, then the timeoutpriority is ignored and the application
argument is used as
; the timeout. Furthermore, if no application argument timeout is
specified, then the
; timeoutpriority option is ignored and the configuration file timeout
is always used
; when calling queue members.
;
; In timeoutpriority=conf mode however timeout specified in config file
will take higher
; priority than timeout in application arguments, so if config file has
timeout 0, each
; queue member will be called indefinitely and application timeout
will be checked only
; after this call attempt. This is useful for having queue members with
custom timeouts
; specified within Dial application of Local channel, and allows
handling NO ANSWER which
; would otherwise be interrupted by queue destroying child channel on
timeout.
;
; The default value for timeoutpriority is "app" since this was how
previous versions of
; Asterisk behaved.
;
;timeout = 15
;retry = 5
;timeoutpriority = app|conf
;
;-----END QUEUE TIMING
OPTIONS-----
; Weight of queue - when compared to other queues, higher weights get
; first shot at available channels when the same channel is included in

; more than one queue.
;
;weight=0
;
; After a successful call, how long to wait before sending a
potentially
; free member another call (default is 0, or no delay)
;
;wrapuptime=15
;
; Autofill will follow queue strategy but push multiple calls through
; at same time until there are no more waiting callers or no more
; available members. The per-queue setting of autofill allows you
```

```
; to override the default setting on an individual queue level.
;
;autofill=yes
;
; Autopause will pause a queue member if they fail to answer a call
;
;autopause=yes
;
; Maximum number of people waiting in the queue (0 for unlimited)
;
;maxlen = 0
;
; If set to yes, just prior to the caller being bridged with a queue
member
; the following variables will be set
; MEMBERINTERFACE is the interface name (eg. Agent/1234)
; MEMBERNAME is the member name (eg. Joe Soap)
; MEMBERCALLS is the number of calls that interface has taken,
; MEMBERLASTCALL is the last time the member took a call.
; MEMBERPENALTY is the penalty of the member
; MEMBERDYNAMIC indicates if a member is dynamic or not
; MEMBERREALTIME indicates if a member is realtime or not
;
;setinterfacevar=no
;
; If set to yes, just prior to the caller being bridged with a queue
member
; the following variables will be set:
; QEHOLDTIME callers hold time
; QEORIGINALPOS original position of the caller in the queue
;
;setqueueentryvar=no
;
; If set to yes, the following variables will be set
; just prior to the caller being bridged with a queue member
; and just prior to the caller leaving the queue
; QUEUENAME name of the queue
; QUEUEMAX maximum number of calls allowed
; QUEUESTATEGY the strategy of the queue;
; QUEUECALLS number of calls currently in the queue
; QUEUEHOLDTIME current average hold time
; QUEUECOMPLETED number of completed calls for the queue
; QUEUEABANDONED number of abandoned calls
; QUEUESRVLEVEL queue service level
; QUEUESRVLEVELPERF current service level performance
;
;setqueuevar=no
```

```
;
; if set, run this macro when connected to the queue member
; you can override this macro by setting the macro option on
; the queue application
;
; membermacro=somemacro

; How often to announce queue position and/or estimated
; holdtime to caller (0=off)
; Note that this value is ignored if the caller's queue
; position has changed (see min-announce-frequency)
;
;announce-frequency = 60
;
; The absolute minimum time between the start of each
; queue position and/or estimated holdtime announcement
; This is useful for avoiding constant announcements
; when the caller's queue position is changing frequently
; (see announce-frequency)
;
;min-announce-frequency = 15
;
; How often to make any periodic announcement (see periodic-announce)
;
;periodic-announce-frequency=60
;
; Should the periodic announcements be played in a random order?
Default is no.
;
;random-periodic-announce=no
;
; Should we include estimated hold time in position announcements?
; Either yes, no, or only once.
; Hold time will be announced as the estimated time.
;
;announce-holdtime = yes|no|once
;
; Queue position announce?
; Valid values are "yes," "no," "limit," or "more." If set to "no,"
then the caller's position will
; never be announced. If "yes," then the caller's position in the queue
will be announced
; to the caller. If set to "more," then if the number of callers is
more than the number
; specified by the announce-position-limit option, then the caller will
hear that there
; are more than that many callers waiting (i.e. if a caller number 6 is
```

```
in a queue with the
; announce-position-limit set to 5, then that caller will hear that
there are more than 5
; callers waiting). If set to "limit," then only callers within the
limit specified by announce-position-limit
; will have their position announced.
;
; announce-position = yes
;
; If you have specified "limit" or "more" for the announce-position
option, then the following
; value is what is used to determine what announcement to play to
waiting callers. If you have
; set the announce-position option to anything else, then this will
have no bearing on queue operation
;
; announce-position-limit = 5
;
; What's the rounding time for the seconds?
; If this is non-zero, then we announce the seconds as well as the
minutes
; rounded to this value.
; Valid values are 0, 5, 10, 15, 20, and 30.
;
; announce-round-seconds = 10
;
; Use these sound files in making position/holdtime announcements. The
; defaults are as listed below -- change only if you need to.
;
; Keep in mind that you may also prevent a sound from being played if
you
; explicitly set a sound to be an empty string. For example, if you
want to
; prevent the queue from playing queue-thankyou, you may set the sound
using
; the following line:
;
; queue-thankyou=
;
;           ("You are now first in line.")
; queue-youarenext = queue-youarenext
;           ("There are")
; queue-thereare   = queue-thereare
;           ("calls waiting.")
; queue-callswaiting = queue-callswaiting
;           ("The current est. holdtime is")
; queue-holdtime = queue-holdtime
```



```
        ;      ("minutes.")
;queue-minutes = queue-minutes
        ;      ("seconds.")
;queue-seconds = queue-seconds
        ;      ("Thank you for your patience.")
;queue-thankyou = queue-thankyou
        ;      ("Hold time")
;queue-reporthold = queue-reporthold
        ;      ("All reps busy / wait for next")
;periodic-announce = queue-periodic-announce
;
; A set of periodic announcements can be defined by separating
; periodic announcements to reproduce by commas. For example:
;periodic-announce =
queue-periodic-announce,your-call-is-important,please-wait
;
; The announcements will be played in the order in which they are
defined. After
; playing the last announcement, the announcements begin again from the
beginning.
;
; Calls may be recorded using Asterisk's monitor/MixMonitor resource
; This can be enabled from within the Queue application, starting
recording
; when the call is actually picked up; thus, only successful calls are
; recorded, and you are not recording while people are listening to
MOH.
; To enable monitoring, simply specify "monitor-format"; it will be
disabled
; otherwise.
;
; You can specify the monitor filename with by calling
;   Set(MONITOR_FILENAME=foo)
; Otherwise it will use MONITOR_FILENAME=${UNIQUEID}
;
; Pick any one valid extension for monitor format recording. If you
leave
; monitor-format commented out, it will not record calls.
;
; monitor-format = gsm|wav|wav49
;
; Monitor Type
;   By setting monitor-type = MixMonitor, when specifying
monitor-format
;   to enable recording of queue member conversations, app_queue will
;   now use the new MixMonitor application instead of Monitor so
;   the concept of "joining/mixing" the in/out files now goes away
```

```
; when this is enabled. If you do not specify or comment out this
option,
; it will default to the old 'Monitor' behavior to keep backward
; compatibility.
;
; monitor-type = MixMonitor
;
; ----- TYPE MIXMONITOR OPTIONS
-----
;
;
; You can specify the options supplied to MixMonitor by calling
; Set(MONITOR_OPTIONS=av(<x>)V(<x>)W(<x>))
; The 'b' option for MixMonitor (only save audio to the file while
bridged) is
; implied.
;
; You can specify a post recording command to be executed after the end
of
; recording by calling
; Set(MONITOR_EXEC=mv /var/spool/asterisk/monitor/^{MONITOR_FILENAME}
/tmp/^{MONITOR_FILENAME})
;
; The command specified within the contents of MONITOR_EXEC will be
executed when
; the recording is over. Any strings matching ^{X} will be unescaped to
${X} and
; all variables will be evaluated just prior to recording being
started.
;
; The contents of MONITOR_FILENAME will also be unescaped from ^{X} to
${X} and
; all variables will be evaluated just prior to recording being
started.
;
;
; This setting controls whether callers can join a queue with no
members. There
; are three choices:
;
; yes - callers can join a queue with no members or only unavailable
members
; no - callers cannot join a queue with no members
; strict - callers cannot join a queue with no members or only
unavailable
; members
; loose - same as strict, but paused queue members do not count as
```

```
unavailable
;
; joinempty = yes
;
;
; If you wish to remove callers from the queue when new callers cannot
join,
; set this setting to one of the same choices for 'joinempty'
;
; leavewhenempty = yes
;
;
; If this is set to yes, the following manager events will be
generated:
; AgentCalled, AgentDump, AgentConnect, AgentComplete; setting this to
; vars also sends all channel variables with the event.
; (may generate some extra manager events, but probably ones you want)
;
; eventwhencalled = yes|no|vars
;
; If this is set to yes, the following manager events will be
generated:
; QueueMemberStatus
; (may generate a WHOLE LOT of extra manager events)
;
; eventmemberstatus = no
;
; If you wish to report the caller's hold time to the member before
they are
; connected to the caller, set this to yes.
;
; reportholdtime = no
;
; If you want the queue to avoid sending calls to members whose devices
are
; known to be 'in use' (via the channel driver supporting that device
state)
; uncomment this option. (Note: only the SIP channel driver currently
is able
; to report 'in use'.)
;
; ringinuse = no
;
; If you wish to have a delay before the member is connected to the
caller (or
; before the member hears any announcement messages), set this to the
number of
```

```
; seconds to delay.
;
; memberdelay = 0
;
; If timeoutrestart is set to yes, then the timeout for an agent to
answer is
; reset if a BUSY or CONGESTION is received. This can be useful if
agents
; are able to cancel a call with reject or similar.
;
; timeoutrestart = no
;
; If you wish to implement a rule defined in queuerules.conf (see
; configs/queuerules.conf.sample from the asterisk source directory for
; more information about penalty rules) by default, you may specify
this
; by setting defaultrule to the rule's name
;
; defaultrule = myrule
;
; Each member of this call queue is listed on a separate line in
; the form technology/dialstring. "member" means a normal member of a
; queue. An optional penalty may be specified after a comma, such that
; entries with higher penalties are considered last. An optional
member
; name may also be specified after a second comma, which is used in log
; messages as a "friendly name". Multiple interfaces may share a
single
; member name. An optional state interface may be specified after a
third
; comma. This interface will be the one for which app_queue receives
device
; state notifications, even though the first interface specified is the
one
; that is actually called.
;
; It is important to ensure that channel drivers used for members are
loaded
; before app_queue.so itself or they may be marked invalid until
reload. This
; can be accomplished by explicitly listing them in modules.conf before
; app_queue.so. Additionally, if you use Local channels as queue
members, you
; must also preload pbx_config.so and chan_local.so (or pbx_ael.so,
pbx_lua.so,
; or pbx_realtime.so, depending on how your dialplan is configured).
;
```

```

;member => DAHDI/1
;member => DAHDI/2,10
;member => DAHDI/3,10,Bob Johnson
;member => Agent/1001
;member => Agent/1002
;member => Local/1000@default,0,John Smith,SIP/1000

;
; Note that using agent groups is probably not what you want.
Strategies do
; not propagate down to the Agent system so if you want round robin,
least
; recent, etc, you should list all the agents in this file individually
and not
; use agent groups.
;
;member => Agent/@1           ; Any agent in group 1
;member => Agent/:1,1       ; Any agent in group 1, wait for first
                           ; available, but consider with penalty

; Cola de administradores

[administradores]
music=default
strategy=ringall
timeout=15
retry=5
wrapuptime=0
maxlen = 0
announce-frequency = 0
announce-holdtime = yes
member => SIP/7000
member => SIP/5006

```

Fuente: http://informatica/manuais/index.php?title=Arquivo_queues.conf&oldid=12074

Autores principales: Veiga,

Arquivo res ldap.conf

```

;
; Configuration file for res_config_ldap
;

; Sample Asterisk config file for res_config_ldap
; in extconfig.conf; you can use it like this:

```

```
; sipusers = ldap,"dc=myDomain,dc=myDomainExt",sip
; sippeers = ldap,"dc=myDomain,dc=myDomainExt",sip
; extensions = ldap,"dc=myDomain,dc=myDomainExt",extensions
; sip.conf = ldap,"dc=myDomain,dc=myDomainExt",config

[_general]
;
; Specify one of either host and port OR url.  URL is preferred, as you
; can
; use more options.
host=127.0.0.1                ; LDAP host
port=389
url=ldap://localhost:389
protocol=3
basedn="dc=iessanclemente,dc=net"        ; Base DN
user=cn=ramiro,dc=iessanclemente,dc=net  ; Bind DN
pass=xxxxxx    ; Bind password

; Configuration Table
[config]
;
; additionalFilter - This specifies an additional set of criteria to be
; used
; when querying the LDAP server.
;
additionalFilter=(objectClass=AstConfig)
;
; Attributes mapping (asterisk variable name = ldap attribute name)
; When Asterisk requests the variable by the name of the value on the
; left,
; this module will look up the attribute listed on the right.
;
filename = AstConfigFilename
category = AstConfigCategory
variable_name = AstConfigVariableName
variable_value = AstConfigVariableValue
cat_metric = AstConfigCategoryMetric
commented = AstConfigCommented

;
; Extensions Table
;
[extensions]
context = AstExtensionContext
exten = AstExtensionExten
priority = AstExtensionPriority
```

```
app = AstExtensionApplication
appdata = AstExtensionApplicationData
additionalFilter=(objectClass=AstExtension)

;
; Sip Users Table
;
[sip]
name = cn
amaflags = AstAccountAMAFflags
callgroup = AstAccountCallGroup
callerid = AstAccountCallerID
canreinvite = AstAccountCanReinvite
context = AstAccountContext
dtmfmode = AstAccountDTMFMode
fromuser = AstAccountFromUser
fromdomain = AstAccountFromDomain
fullcontact = AstAccountFullContact
fullcontact = gecost
host = AstAccountHost
insecure = AstAccountInsecure
mailbox = AstAccountMailbox
md5secret = realmedPassword
nat = AstAccountNAT
deny = AstAccountDeny
permit = AstAccountPermit
pickupgroup = AstAccountPickupGroup
port = AstAccountPort
qualify = AstAccountQualify
restrictcid = AstAccountRestrictCID
rtptimeout = AstAccountRTPTIMEout
rtpholdtimeout = AstAccountRTPHoldTIMEout
type = AstAccountType
disallow = AstAccountDisallowedCodec
allow = AstAccountAllowedCodec
MusicOnHold = AstAccountMusicOnHold
regseconds = AstAccountExpirationTimestamp
regcontext = AstAccountRegistrationContext
regexten = AstAccountRegistrationExten
CanCallForward = AstAccountCanCallForward
ipaddr = AstAccountIPAddress
defaultuser = AstAccountDefaultUser
regserver = AstAccountRegistrationServer
additionalFilter=(objectClass=AsteriskSIPUser)

;
; IAX Users Table
```

```
;
[iax]
amaflags = AstAccountAMAFlags
callerid = AstAccountCallerID
context = AstAccountContext
fullcontact = AstAccountFullContact
fullcontact = gecost
host = AstAccountHost
mailbox = AstAccountMailbox
md5secret = realmedPassword
deny = AstAccountDeny
permit = AstAccountPermit
port = AstAccountPort
qualify = AstAccountQualify
type = AstAccountType
disallow = AstAccountDisallowedCodec
allow = AstAccountAllowedCodec
regseconds = AstAccountExpirationTimestamp
regcontext = AstAccountRegistrationContext
regexten = AstAccountRegistrationExten
nottransfer = AstAccountNoTransfer
additionalFilter=(objectClass=AstAccountIAX)

;
; A Test Family
;
[testfamily]
MyUSERID = uid
additionalFilter=(objectClass=*)

[accounts]
amaflags = AstAccountAMAFlags
callgroup = AstAccountCallGroup
callerid = AstAccountCallerID
canreinvite = AstAccountCanReinvite
context = AstAccountContext
dtmfmode = AstAccountDTMFMode
fromuser = AstAccountFromUser
fromdomain = AstAccountFromDomain
fullcontact = AstAccountFullContact
fullcontact = gecost
host = AstAccountHost
insecure = AstAccountInsecure
mailbox = AstAccountMailbox
md5secret = realmedPassword
nat = AstAccountNAT
deny = AstAccountDeny
```



```
permit = AstAccountPermit
pickupgroup = AstAccountPickupGroup
port = AstAccountPort
qualify = AstAccountQualify
restrictcid = AstAccountRestrictCID
rtptimeout = AstAccountRTPTIMEOUT
rtpholdtimeout = AstAccountRTPHoldTimeout
type = AstAccountType
disallow = AstAccountDisallowedCodec
allow = AstAccountAllowedCodec
MusicOnHold = AstAccountMusicOnHold
regseconds = AstAccountExpirationTimestamp
regcontext = AstAccountRegistrationContext
regexten = AstAccountRegistrationExten
CanCallForward = AstAccountCanCallForward
additionalFilter=(objectClass=AstAccount)
```

Fuente: http://informatica/manuais/index.php?title=Arquivo_res_ldap.conf&oldid=12077

Autores principales: Veiga,

Arquivo sip.conf

```
;
; SIP Arquivo de Configuración
;
; SIP dial strings

[general]
; Configurar a IP externa do router.
externip=213.96.94.123

; Describir todas as redes locais as que conecta o Asterisk.
localnet=10.0.0.0/255.0.0.0
localnet=192.168.30.0/255.255.255.0

; IP e porto no que escoita o Asterisk
bindaddr=0.0.0.0
port=5060
nat=yes
qualify=yes
realm=voip.iessanclemente.net
fromdomain=voip.iessanclemente.net
canreinvite=no
```

```
language=es

allowoverlap=no
context=default ; Default context for incoming calls

disallow=all
allow=ulaw
allow=alaw
allow=gsm

; Sección de Profesorado

[5000] ; SIP Veiga.
type=friend
secret=hellopeople
callerid="IeSC Veiga" <5000>
mailbox=5000@default
context=personal
host=dynamic

[5001]
type=friend
secret=hellopeople
callerid="IeSC Vieites" <5001>
mailbox=5001@default
context=personal
host=dynamic

[7000] ; Telefono SIP Informatica.
type=friend
secret=welcome
callerid="IeSC Informatica" <7000>
mailbox=7000@default
context=personal
host=dynamic
```

O parámetro **dtmfmode** é importante, xa que dependendo do que poñamos (inband, rfc2833, info, auto) poderemos enviar tonos ó asterisk ou non.

dtmfmode=inband emprégase con codecs alaw, ulaw **dtmfmode=rfc2833** emprégase con gsm

Isto é importante nos teléfonos hardware SIP que configuremos.

A solución consiste en non indicar nada, xa que usará por defecto rfc2833, e poremos no cliente SIP ZOIPER ^[1] na configuración de SIP **USE DTMF SIP INFO**.

<http://www.voip-info.org/wiki/view/Asterisk+sip+dtmfmode>

Enlaces externos

[1] <http://www.zoiper.com/>

Fuente: http://informatica/manuais/index.php?title=Arquivo_sip.conf&oldid=12055

Autores principales: Veiga,

Arquivo system.conf

O ficheiro system.conf quedaría:

```
# Autogenerated by /usr/sbin/dahdi_genconf on Tue May 19 13:07:48 2009
# If you edit this file and execute /usr/sbin/dahdi_genconf again,
# your manual changes will be LOST.
# Dahdi Configuration File
# This file is parsed by the Dahdi Configurator, dahdi_cfg
# Span 1: WCTDM/0 "Wildcard AEX410 Board 1" (MASTER)
fxsks=1
echocanceller=mg2,1
# Global data
loadzone = es
defaultzone = es
```

Fuente: http://informatica/manuais/index.php?title=Arquivo_system.conf&oldid=12097

Autores principales: Veiga,

Arquivo voicemail.conf

```
;!
;! Automatically generated configuration file
;! Filename: voicemail.conf (/etc/asterisk/voicemail.conf)
;! Generator: AppVoicemail
;! Creation Date: Wed May 13 01:47:49 2009
;!
;
; Voicemail Configuration
;
[general]
; Formats for writing Voicemail. Note that when using IMAP storage for
; voicemail, only the first format specified will be used.
;format=g723sf|wav49|wav
format = wav49|gsm|wav
; Who the e-mail notification should appear to come from
serveremail = asterisk
```

```
; Should the email contain the voicemail as an attachment
attach = yes
; How many milliseconds to skip forward/back when rew/ff in message
playback
skipms = 3000
; How many seconds of silence before we end the recording
maxsilence = 10
; Silence threshold (what we consider silence: the lower, the more
sensitive)
silencethreshold = 128
; Max number of failed login attempts
maxlogins = 3

[zonemessages]
eastern = America/New_York|'vm-received' Q 'digits/at' IMp
central = America/Chicago|'vm-received' Q 'digits/at' IMp
central24 = America/Chicago|'vm-received' q 'digits/at' H N 'hours'
military = Zulu|'vm-received' q 'digits/at' H N 'hours' 'phonetic/z_p'
european = Europe/Copenhagen|'vm-received' a d b 'digits/at' HM

[default]
5000 => 1234,Rafa Veiga,rafael@edu.xunta.es
5001 => 1234,Manuel Vieites,vieites@edu.xunta.es
7000 => 1234,informatica,informatica_sc@terra.es
```

Fuente: http://informatica/manuais/index.php?title=Arquivo_voicemail.conf&oldid=12054

Autores principales: Veiga,

Comparación de CODECS de audio

Listado dos codecs que soporta Asterisk e o seu ancho de banda necesario:

- * G.711 ulaw (utilizado en EEUU) (64 Kbps)
- * G.711 alaw (utilizado en Europa) (64 Kbps)
- * G.723.1 - pass-thru sin licencia (6.3/5.3Kbps) usado en H.323
- * G.726 - (16/24/32/40kbps)
- * G.729 - pass-thru sin licencia (8Kbps)
- * GSM (13Kbps)
- * iLBC (13.33/15.2Kbps)
- * LPC10 (non recomendado!)
- * Speex - configurable 4-48kbps

Fuente: http://informatica/manuais/index.php?title=Comparación_de_CODECS_de_audio&oldid=11566

Autores principales: Veiga,

Creación do dialplan

O *dialplan* ou plan de marcado constitúe unha das partes máis importantes de Asterisk. Realmente é nel onde se amosa toda a potencia, versatilidade e flexibilidade que pode chegar a ter unha centralita telefónica software.

O *dialplan* é o centro de Asterisk. Cada dígito que se marca nun terminal recorrerá o *dialplan*, buscando "qué facer". O *dialplan* poderíase asimilar a unha táboa de enrutado, de forma que o usuario marca unha secuencia de díxitos ou caracteres e nela se atopan as accións a realizar para a secuencia de marcado.

Esquema de traballo

O *dialplan* de Asterisk atópase no arquivo `extensions.conf` e está composto por contextos, extensións e prioridades.

As **extensións** son números (ou caracteres alfanuméricos) onde o usuario pode chamar. Estas extensións, conteñen accións asociadas, que son executadas de xeito secuencial, por orden de **prioridade**.

Os **contextos** son agrupacións lóxicas de extensións, e se empregan para dividir o *dialplan* en diversos entes lóxicos. Esta división é necesaria para poder dispor dun plan sostible, escalable e con posibilidades de ofrecer diversos entornos de marcado aillados.

A sintaxis para as extensións dun contexto é a seguinte:

```
exten => numero de extensión, prioridade, aplicación(argumentos)
```

Ó definir un usuario no `sip.conf` ou `iax.conf`, asociámoslle un contexto, polo que ese usuario só poderá marcar as extensións incluídas no seu contexto.

```
[proba]
exten => 1234,1,NoOp(Isto é unha proba)
exten => 1234,n,NoOp(Isto é outra proba)
```

No exemplo de arriba amósase un sencillo contexto cunha extensión e dúas prioridades. Sempre é necesaria a prioridade 1, pero para as seguintes prioridades pódese usar a "n", que fará que se incremente automaticamente nunha unidade, evitando así ter que face-lo manualmente.

Ó marcar unha extensión vanse executando as aplicacións correspondentes de xeito secuencial, de acordo á súa prioridade:

```
exten => 1234,1,Aplicación 1
exten => 1234,2,Aplicación 2
```

O uso de prioridades numéricas pode dificultar a modificación do *dialplan* nun futuro, xa que si se desexa insertar unha acción no medio, sería necesario modificar a prioridade de todas as seguintes.

Para evitar isto podemos empregar a prioridade "n", que simplemente indica "a seguinte prioridade".

```

exten => 1234,1,Aplicación 1
exten => 1234,n,Aplicación 2
...
exten => 1234,n,Aplicación 9

```

Como pode observarse, deste xeito é moito máis sinxelo insertar novas aplicacións en medio, sin necesidade de preocuparnos de cambiar os números de prioridade.

En lugar de ter que expresar todas as combinacións de extensións posibles, podemos empregar patróns e variables, ó obxecto de dispor dun *dialplan* máis claro e sostible.

Os patróns comezan cun guión baixo "_" e empregan o seguintes caracteres especiais:

```

X: Calqueira dígito do 0 ó 9
Z: Calqueira dígito do 1 ó 9
N: Calqueira dígito do 2 ó 9
[: Calqueira dígito que se atope entre os corchetes, por exemplo [123]
  implica o 1, o 2 e o 3.

```

```

. (punto): Calqueira cousa, por exemplo _9.

```

Implica calqueira número que comece por 9, sin ter en conta o 9 en si mesmo.

```

!: Caracter de desambigüación.

```

Indica que o procesamento ten que ser detido tan pronto como se atope un patrón axeitado.

Empregando os patróns é posible simplificar o *dialplan* e na práctica empréganse patróns como os seguintes:

_ [789]XXXXXXXX: Números fixos nacionais (que comece por 7,8 ou 9 e teñen nove díxitos)

_6XXXXXXXX: Números de móbil (comezan por 6 e teñen nove díxitos)

Aplicacións e funcións

Ó marcar unha extensión execútase a aplicación asociada á prioridade correspondente. A diferenza das aplicacións, as funcións soamente poden ser empregadas dentro de ditas aplicacións. Vexamos un exemplo:

```

exten => 2000,1,Dial(SIP/carlos)

```

Ó marcar a extensión 2000 executarase a aplicación *Dial*, á que lle indicamos que chame a Carlos, que é un usuario SIP.

Primeiro indicamos a tecnoloxía (SIP/) e despois o nome do usuario que definimos no arquivo correspondente, neste caso *sip.conf*.

As funcións, en cambio execútanse dentro das aplicacións, por exemplo:

```

exten => 1234,1,NoOp(Estas chamándome dende o : ${CALLERID(num)})

```

Neste caso execútase a aplicación *Noop*, que únicamente imprime a mensaxe que se lle pasa como argumento por pantalla, e no seu interior execútase a función *CALLERID* con argumento *num*, que devolve o número chamante.

Listaxe 1: Exemplo de dialplan básico

```
[a-extensions]
exten => 2000,1,Dial(SIP/carlos,60,Tt)
exten => 2001,1,Dial(SIP/mendez,60,Tt)

[a-servizos]
exten => 1234,1,Playback(tt-monkeys)
exten => 1235,1,Dial(IAX2/guest@pbx.digium.com/s@default)

[dende-usuarios]
include => a-extensions
include => a-servizos
```

Neste exemplo definimos tres contextos: o contexto *a-extensions* inclúe as extensións 2000 e 2001, que fan uso da aplicación *Dial* para chamar a carlos e a mendez respectivamente.

O contexto *a-servizos* contén dous servizos: 1234, que reproduce unha locución de exemplo (o sonido duns monos) e 1235, que fai unha chamada a Digium a través do IAX2.

O contexto *dende-usuarios* inclúe os outros dous contextos, e é o que asignaremos ás extensións no sip.conf, de xeito que sexan capaces de marcar todas as extensións incluídas por este contexto.

Podería facerse todo nun só contexto, pero iso derivaría nun *dialplan* pouco sostible.

Macros

En moitas ocasións atoparémonos con fragmentos do '**dialplan**' que son idénticos, e nos que soamente cambia un pequeno detalle.

Seguindo a filosofía do *dialplan* da **Listaxe 2**, teríamos que copiar o mesmo para a extensión 2001,2002,2003, modificando en cada fragmento o protocolo a usar (SIP ou IAX), nome do usuario e o número de extensión. Isto, ademais de ser bastante traballoso, complicaría o día de mañá calquera modificación.

Listaxe 2: Fragmento dun dialplan

```
[a-extensions]
exten => 2000,1,Dial(SIP/mendez,20)
exten => 2000,n,Goto(2000- $\${DIALSTATUS}$ ,1)

; si está ocupado, reproducése a mensaxe correspondente
exten => 2000-BUSY,1,VoiceMail(2000,b)
exten => 2000-BUSY,n,Hangup

; si non contesta, reproducése a mensaxe de "no dispoñible"
exten => 2000-NOANSWER,1,VoiceMail(2000,u)
exten => 2000-NOANSWER,n,Hangup

; en calqueira outro caso, facemos o mesmo que si non contesta.
exten => 2000-.,1,Goto(2000-NOANSWER)
```

Para solucionar este problema, poderemos definir **macros**, que son o equivalentes a "funcións" ou "subrutinas" das linguaxes de programación. Do mesmo xeito que as funcións, unha macro recibirá uns parámetros que poderemos utilizar dentro da súa definición.

As macros defínense no ficheiro *extensions.conf*, como un contexto máis, tendo en conta unha serie de condicións:

1. O nome do "contexto" empezará por "macro-" para identificar que se trata dunha macro, e non dun contexto normal.
2. As macros sempre empezarán pola extensión "s". Ademais, non se permite un "goto" dende outra parte do *dialplan* a un punto diferente da macro que non sexa ó seu inicio.
3. Dentro da definición da macro, os parámetros recibidos son accesibles mediante `${ARG1}`, `${ARG2}`... E a extensión dende a que se chamou á macro, podémola obter en `${MACRO_EXTEN}`.

Para executar unha macro, no *dialplan* temos dispoñible a aplicación "Macro":

```
exten => 1234,1,Macro(nome_macro, parametro1, parametro2,
parametro3, ...)
```

No exemplo que vemos na **Listaxe 3** usamos unha macro para o código que tiñamos, saltando á caixa de correos de voz se está ocupado. A esta macro chamarémola "extension-VM".

Listaxe 3: Extension VM

```
[macro-extensionVM]
;esta macro recibirá dous parámetros
;o primero indica o dispositivo ó que hai que chamar
;o segundo é o número do buzón de voz ó que desviar a chamada
exten => s,1,NoOp(chamada para ${ARG1})
exten => s,n,Dial(${ARG1},20)
exten => s,n,Goto(s-${DIALSTATUS},1)

; si está ocupado, reproducése a mensaxe correspondente
exten => s-BUSY,1,VoiceMail(${ARG2},b)
exten => s-BUSY,n,Hangup

; si non contesta, reproducése a mensaxe de "non dispoñible"
exten => s-NOANSWER,1,VoiceMail(${ARG2},u)
exten => s-NOANSWER,n,Hangup

; en calqueira outro caso, facemos o mesmo que si non contesta.
exten => s-.,1,Goto(s-NOANSWER,1)
```

E adaptamos o contexto *[a-extensions]* para que use esta macro para chamar ós usuarios:

```
[a-extensions]
exten => 2000,1,Macro(extensionVM,SIP/mendez,2000)
exten => 2001,1,Macro(extensionVM,SIP/carlos,2001)
exten => 2002,1,Macro(extensionVM,SIP/laura,2002)
exten => 2003,1,Macro(extensionVM,IAX2/juan,2003)
exten => 2004,1,Macro(extensionVM,IAX2/maria,2004)
```


Isto facilita moitísimo o mantemento do *dialplan*, e faino ao mesmo tempo máis lexible.

Gardar información na base de datos ASTDB

Asterisk mantén determinada información sobre o seu estado actual nunha base de datos coñecida como **AstDB**. A información que se almacena nesta base de datos inclúe os datos de rexistro dos usuarios, estado das colas, etc. Isto permite que o sistema poida recuperar esta información en caso de ter que reiniciarse.

A información na *AstDB* organízase en familias e identifícanse mediante unha chave que será única dentro da familia. Para cada familia e chave pódese almacenar un valor.

Dende o *CLI* dispomos de distintos comandos cos que podemos acceder á información da *AstDB*, podendo tanto ler como escribir nela:

database show <familia> <chave>: Aмосa o contido da base de datos, podendo filtrar por familia e chave. Vemos un exemplo na *Listaxe 4*.

database del <familia> <chave>: Permite borrar un valor da base de datos.

database deltree <familia> <árbore-de-chaves>: . Borra toda unha árbore de claves da base de datos. Por exemplo, pode borrar toda a árbore "Registry" da familia "IAX".

database put <familia> <chave> <valor>: . Permite introducir información na base de datos (ver *Listaxe 5*).

Listaxe 4: Contido da base de datos AstDB

```
PBX*CLI> database show
/IAX/Registry/caracas : 192.168.15.209:4569:60
/IAX/Registry/juan    : 192.168.15.2:4570:60
/SIP/Registry/carlos :
192.168.15.2:5060:3600:carlos:sip:carlos@192.168.15.2

PBX*CLI> database show iax
/IAX/Registry/caracas : 192.158.15.209:4569:60
/IAX/Registry/juan    : 192.168.15.2:4570:60

PBX*CLI> database show iax registry/juan
/IAX/Registry/juan    : 192.168.15.2:4570:60

PBX*CLI> database showkey iax/registry/juan
/IAX/Registry/juan    : 192.168.15.2:4570:60
```

Listaxe 5: Introducindo información

```
PBX*CLI> database put familia chave valor
Updated database successfully

PBX*CLI> database show
/IAX/Registry/caracas : 192.168.15.209:4569:60
/IAX/Registry/caracas : 192.168.15.2:4570:60
/IAX/Registry/caracas :
192.168.15.2:5060:3600:carlos:sip:carlos@192.168.15.2
/familia/chave       : valor
```

Ler a información que Asterisk garda na base de datos pode resultar interesante para determinadas aplicacións, pero o verdadeiramente interesante desta base de datos é que nós podemos tamén gardar e ler información propia, para utiliza-la dende o *dialplan*. Para isto dispomos dunha serie de funcións:

Gardar un valor na base de datos:

```
exten => s,1,Set( DB(familia/chave) = valor )
```

Ler un valor da base de datos:

```
exten => s,1,Set( var = ${DB(familia/chave)})
```

Comprobar se existe un valor para unha familia e chave:

```
exten => s,1,GotoIf ( ${DB_EXISTS(familia/chave)} = 0 )?s, chamar )
```

No caso de existir un valor para a familia/chave, *DB_EXISTS* garda este valor na variable *DB_RESULT*.

Podemos probar a chamar ao número *78 e despois executar database *show* no *CLI*. Veremos que se gardou un rexistro na base de datos, da forma: *DND/num_extension*.

Agora actualizaremos a nosa macro *ex-tensionVM* para que teña en conta se o usuario ten activado o "non-molestar", como podemos ver na *Listaxe 6*.

Listaxe 6: Actualizando extension VM

```
[macro-extensionVM]
; esta macro recibirá dous parámetros
; o primeiro indica o dispositivo ó que hai que chamar
; o segundo é o número do buzón de voz ó que desviar a chamada
exten => s,1,NoOp(chamada para ${ARG1})

; si NON ten activado o no-molestar, saltamos a "chamar"
exten => s,n,Gotolf( ${DB_EXISTS(DND/${MACRO_EXTEN})} = 0 )?s, chamar )
exten => s,n(DND),Playback(vm-extension)
exten => s,n,Playback(vm-isunavail)
exten => s,n,Hangup
exten => s,n(chamar),Dial( ${ARG1},20)
exten => s,n,Goto(s-${DIALSTATUS},1)

; si está ocupado, reproducése a mensaxe correspondente
exten => s-BUSY,1,VoiceMail( ${ARG2},b)
exten => s-BUSY,n, Hangup

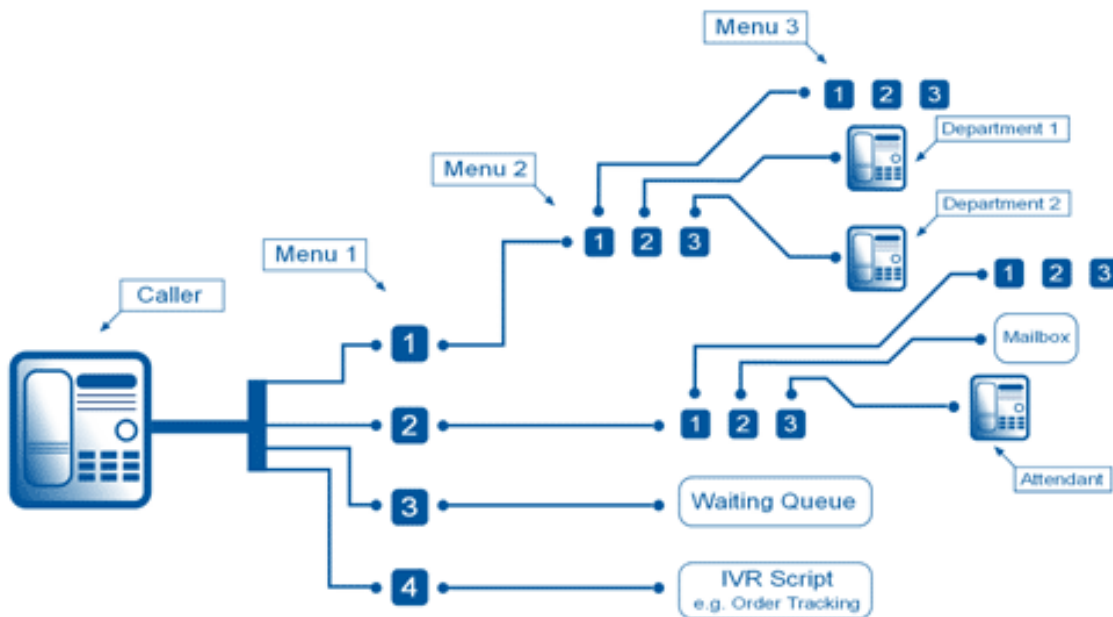
; si non contesta, reproducése a mensaxe de "non dispoñible"
exten => s-NOANSWER,1,VoiceMail( ${ARG2},u)
exten => s-NOANSWER,n,Hangup
exten => s-NOMOLESTAR,1,Playback(vm-extension)
exten => s-NOMOLESTAR,n,Playback(vm-isunavail)
exten => s-NOMOLESTAR,n,Hangup

; en calqueira outro caso, facemos o mesmo que si non contesta.
```

```
exten => s-.,1,Goto(s-NOANSWER,1)
```

Menú de voz interactivo ou IVR

Un IVR é un menú onde o usuario pode interactuar mediante pulsacións DTMF (tonos multifrecuencia) co sistema telefónico. As empresas adoitan usar estes menús, reproducindo unha mensaxe de benvida cando reciben unha chamada, e ofrecéndonos despois as típicas opcións de "pulse 1 para falar co departamento de vendas; 2 para falar con contabilidade...".



Ésta sería a forma máis sinxela de IVR coñecido tamén como **Operadora Automática**.

Asterisk ofrécenos todas as ferramentas necesarias para crear dende o IVR máis simple ata os sistemas máis complexos.

O primeiro que necesitaremos para crear un IVR son os sons que se van a reproducir. Estes deben atoparse no directorio de sonidos do Asterisk, normalmente `/var/lib/asterisk/sounds/`, e ter un formato recoñecido por éste.

Xunto con Asterisk se distribúe un xogo de sons estándar en inglés. Existen xogos de voces de gran calidade en castelán, pero normalmente necesitaremos algúns sons personalizados ("grazas por chamar ó Instituto San Clemente", "pulse 1 para falar co departamento de informática...", etc.).

Aínda que é posible usar a aplicación Record de Asterisk para gravar as mensaxes, para obter a mellor calidade aconséllase gravalos cunha aplicación especializada (Audacity ^[1], por exemplo). Neste caso debemos gravar os arquivos a 8KHz e 16 bits.

Para gravar mensaxes dende o *dialplan* emprégase a aplicación *Record*, coa sintaxe:

```
Record(nome.formato|silencio|maxDuracion|opcións)
```

Si dentro do nome do arquivo engadimos os caracteres %d, éstos serán substituídos por un número secuencial, para evitar que sobreescibamos unha gravación existente. Neste caso, a variable `${RECORDED_FILE}` conterá o nome final do arquivo. A aplicación Record gravará a mensaxe ata que o usuario pulse a tecla #, ou ben ata que chegue á duración máxima indicada como parámetro ou se detecten tantos segundos de silencio como se

indicaron.

```
[servizos]
exten => *77,1,Record(sonido-%d.alaw)
exten => *77,n,Playback(beep)
exten => *77,n,Playback(${RECORDED_FILE})
exten => *77,n,Hangup
```

Para programar os nosos IVR dispoñemos de algunhas aplicacións interesantes para o *dialplan*:

1. **Playback(sonido)**: Reproduce un sonido
2. **WaitExten(tempo)**: Espera a que o usuario teclee unha opción (ou un número de extensión).
3. **Background(sonido)**: Reproduce un son, pero o usuario pode interrompir a reprodución, tecleando un número de opción. Sería equivalente a empregar as aplicacións Playback y WaitExten de xeito simultáneo.
4. **GotoIfTime(hora|dias_semana|dias_mes|ano?si_certo:si_falso)**: Realiza un salto a outro punto do *dialplan* dependendo da data e a hora. Resulta moi útil para actuar de xeito diferente si estamos en horario de oficina ou non.

Para que non entre nun bucle infinito dispoñemos de dous tipos de retardo: tempo inter-dixito e tempo de resposta:

```
Set(TIMEOUT(digit)=3)
Set(TIMEOUT(response)=9)
```

Unha vez que o usuario comeza a teclear unha opción, o tempo máximo permitido entre díxitos será *TIMEOUT(digit)*. Si pasa este tempo dende o último díxito tecleado, o sistema considera que o usuario rematou de teclear o número da opción e saltará a esa extensión (ou opción). Si non existe, saltará á extensión "i".

Por outra parte, si o usuario non comeza a teclear unha opción no tempo indicado por *TIMEOUT(response)*, o IVR saltará á extensión "t". (Ver Listaxe 7).

Listaxe 7: Programando o IVR

```
; IVR ó que chegarán as chamadas entrantes
; A mensaxe de Benvida dice "Grazas por chamar ó IES San Clemente
; Para falar có departamento de Informática pulse 1,
; para falar có departamento de Idiomas pulse 2.
; Se coñece a extensión da persoa
; coa que desexa falar, márquea agora
```

```
[entrada]
exten => s,1,Set(TIMEOUT(digit)=3)
exten => s,n,Set(TIMEOUT(response)=9)

; Comprobamos si estamos no horario do instituto
exten => s,n,GotoIfTime(09:00-22:30|mon-fri|*|*?dentro_horario)
exten => s,n,Playback(estamos-pechados-deixe-mensaxe)
exten => s,n,VoiceMail(999,s)
exten => s,n,Hangup
```

```
exten => s,n(dentro_horario),Background(benvida)
exten => s,n,WaitExten(10)
exten => s,n, Goto (operadora)

; Si elixe a opción 1, mandamos ó usuario á cola de Informatica
exten => 1,1,Queue(informatica,t,60)
exten => 1,n, Hangup

; Si elixe a opción 2, mandamos ó usuario á cola de Idiomas
exten => 2,1,Queue(idiomas,t,60)
exten => 2,n, Hangup

; Si elixe unha opción incorrecta, volvemos a darlle as instruccións
exten => i,1,Goto(s,1)

; Si non selecciona ningunha opción, o mandamos á cola de operadora
exten => t,1,Queue(operadora)

; Ó incluír o contexto a-extensions, permitimos que o usuario
; teclee un número de extensión directamente
include => a-extensions
```

Sala de Conferencias

As salas de conferencia permiten que varios usuarios manteñan unha conversación entre eles, como si estiveran reunidos nunha mesma sala.

As posibilidades que ofrece Asterisk para a creación de salas de conferencias son enormes, podendo crear salas nas que unha persoa pode falar mentras todos os demais escoitan, ou ben que todos poidan intervir ó mesmo tempo.

Tamén existe a posibilidade de enmudecer ou expulsar a un usuario dunha sala durante a conferencia, bloquear para que non poidan entrar máis usuarios, gravar a conferencia, etc.

Podemos definir unha serie de salas de xeito estático, no arquivo *meetme.conf*. Este arquivo contén unha sección *[general]* cun único parámetro *audiobuffers* para indicar a cantidade de buffers que empregará a aplicación para amortiguar o jitter (retardo).

Despois da sección *[general]* hay unha sección *[rooms]* coa definición das distintas salas de conferencias. Ver na **listaxe 8** un exemplo do arquivo *meetme.conf*.

Listaxe 8: Arquivo *meetme.conf*

```
[general]
audiobuffers=5

[rooms]
; Sala de conferencias nº 100, sin PIN
conf => 100

; Sala de conferencias nº 101 con PIN 1234
conf => 101,1234
```

```
; Sala de conferencias nº 102, con PIN 1234 e PIN de administrador 5544
conf => 102,1234,5544
```

Aínda tamén é posible crear salas dinamicamente, por parte dos mesmos usuarios.

A aplicación que empregaremos no *dialplan* para acceder ás salas de conferencias é *Meetme*:

```
exten => xxxx,n,Meetme(numero_de_sala[|opcions[|PIN]])
```

As opcións que podemos especificar son as seguintes: **a.** Entrar en modo Administrador. O administrador pode bloquear a conferencia para que non entren máis participantes. (ver opción "s")

A. Entrar en modo "marcado". Cando este usuario salga da conferencia, ésta finaliza.

b. Executa o script AGI indicado en {MEETME_AGI_BACKGROUND}. Por defecto: conf-background.agi só funciona se todas as canles da conferencia son ZAP.

c. Reprodúcese un aviso indicando cantos usuarios hai na conferencia.

d. Crea a conferencia dinamicamente.

D. Crea a conferencia dinamicamente, e asígnalle un PIN.

Amplíemos o noso contexto [*servizos*], cunha extensión para crear e unirse a unha conferencia:

```
exten => 800,1,Meetme(,DMIsr)
exten => 800,n,hangup
```

Cando un usuario chame á extensión 800, o sistema pediralle que introduza un número de conferencia, terminando con #. O usuario pode asignar o número que desexe (con calquera cantidade de dígitos).

Seguidamente terá que asignar un PIN a esta conferencia, e dicir o seu propio nome. Desta forma créase unha nova conferencia dinamicamente, co número asignado polo usuario (parámetro D).

O usuario que creou a conferencia quedarase conectado, escoitando unha música en espera ata que se conecte un segundo usuario (parámetro M).

Para que os demais usuarios se poidan unir á conferencia soamente teñen que marcar a extensión 800, teclear o número de conferencia que creou o primeiro usuario, e o mesmo PIN.

Cada usuario que se une á conferencia deberá dicir o seu nome. Asterisk usará este nome para reproducir un aviso aos demais usuarios, indicando o nome de quen se une á conferencia, e de quen sae da mesma (parámetro I).

Durante a conferencia, cada usuario poderá marcar a tecla * para enmudecer a súa terminal ou adaptar os niveis de volume do audio que envía/recibe (parámetro s).

E por último, a conferencia gravarase nun arquivo co nome *meetme-conf-rec-<nº conferencia>-<ID unico>.wav* no directorio */var/lib/asterisk/sounds/* (parámetro r).

Tamén existe unha aplicación para administrar salas de conferencias desde o mesmo *dialplan*. Esta aplicación é *MeetmeAdmin* e a súa sintaxis é:

```
exten => xxx.n,MeetmeAdmin (nº_conferencia|comando[|usuario])
```

Esta aplicación, xunto coa opción "X" de *Meetme*, que permite que o usuario poida saír da conferencia tecleando un número de extensión dun só dígito, seranos moi útil para dar ao administrador da conferencia o poder de expulsar ou silenciar a aqueles usuarios molestos na conferencia.

Enlaces externos

[1] <http://audacity.sourceforge.net/?lang=es>

Fuente: http://informatica/manuais/index.php?title=Creación_do_dialplan&oldid=11373

Autores principais: Veiga,

Asterisk GUI -Graphic User Interface

Para poder administrar en modo gráfico via web nos descargaremos a derradeira versión de Asterisk GUI.

```
cd /usr/src
svn co http://svn.digium.com/svn/asterisk-gui/branches/2.0
asterisk-gui

cd asterisk-gui
./configure
make
make install
```

Una vez feito isto teremos que edita-los seguintes arquivos e activar os parámetros indicados:

en **/etc/asterisk/httpd.conf**:

```
# Neste arquivo /etc/asterisk/httpd.conf
#Activaremos o seguinte:

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

en **/etc/asterisk/manager.conf**:

```
#No arquivo /etc/asterisk/manager.conf
#Activaremos o seguinte

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

```
# E crearemos un novo usuario para administrar o Asterisk.
```

```
[admin]
```

```
secret = contrasinal
```

```
read=system,call,log,verbose,command,agent,user,config,originate,read,write
```

```
write=system,call,log,verbose,command,agent,user,config,originate,read,write
```

Unha vez terminada a configuración de arquivos faremos:

```
/usr/src/asterisk-gui/make checkconfig
```

```
# Se todo está ok, teremos que '''reinicia-lo Asterisk''' para que esté  
dispoñible o servizo web.
```

```
# dende CentOS poremos facer dende a liña de comandos:
```

```
service asterisk restart
```

```
# ou dende o CLI de Asterisk tamén poderemos facer un reload:
```

```
CLI> reload
```

A partir deste momento teremos dispoñible a administración web en:

```
http://ip-asterisk:8088/asterisk/static/config/index.html
```

Fuente: http://informatica/manuais/index.php?title=Asterisk_GUI_-_Graphic_User_Interface&oldid=11510

Autores principais: Veiga,

Instalación de Festival -sintetizador de voz

Para instalar o sintetizador de voz para poder usalo no Asterisk faremos dende CentOS:

```
yum install festival
```

Para arrancar o sintetizador en modo debug:

```
festival --server
```

Cando nos aseguremos de que Festival acepta as solicitudes sen problemas poderemos arrancalo con:

```
festival_server 2>&1 >/dev/null &
```


Chamando a Festival dende o dialplan

Agora que xa temos configurado Festival e o servidor está funcionando, llamaremos directamente a Festival dende o dialplan:

```
exten => 123,1,Answer()
exten => 123,2,Festival(Asterisk e Festival estan traballando xuntos)
```

Sempre temos que chamar a aplicación Answer() antes de chamar a Festival(), para asegurarnos de que o canal está establecido.

Cando Asterisk conecta a Festival, deberíamos ver unha saída como a seguinte:

```
festival --server
server Sun May 1 18:38:51 2005 : Festival server started on port 1314
client(1) Sun May 1 18:39:20 2005 : accepted from voip.local
client(1) Sun May 1 18:39:21 2005 : disconnected
```

Si vemos unha saída como a seguinte, isto quere dicir que temos que engadir ó arquivo festival.scm o noso host:

```
festival --server
server Sun May 1 18:30:52 2005 : Festival server started on port 1314
client(1) Sun May 1 18:32:32 2005 : rejected from voip.local not in
access list
```

Outro xeito de empregar Festival con Asterisk

Algunha xente da comunidade Asterisk reportou casos de éxito enviando texto á utilidade text2wave de Festival. Por exemplo:

```
exten => 124,1,Answer()
exten => 124,2,System(echo "This is a test of Festival" |
/usr/bin/text2wave
-scale 1.5 -F 8000 -o /tmp/festival.wav)
exten => 124,3,Playback(/tmp/festival)
exten => 124,4,System(rm /tmp/festival.wav)
exten => 124,5,Hangup()
```

Este método tamén permite chamar a outro tipo de sintetizadores como por exemplo o popular Cepstral <http://www.cepstral.com>, que é un derivado comercial non demasiado caro con voces de boa calidade. Para o seguinte exemplo asumimos que cepstral está instalado en /usr/local/cepstral/:

```
exten => 125,1,Answer()
exten => 125,2,System(/usr/local/cepstral/bin/swift -o /tmp/swift.wav
"This is a test of Cepstral")
exten => 125,3,Playback(/tmp/swift)
exten => 125,4,System(rm /tmp/swift.wav)
exten => 125,5,Hangup()
```

Fuente: http://informatica/manuais/index.php?title=Instalación_de_Festival_-_sintetizador_de_voz&oldid=11496

Autores principales: Veiga,

Encamiñamento de chamadas a través do provedor máis económico

Introdución

A existencia de múltiples provedores de VoIP, con diferentes tarifas, ubicacións, etc., fan necesaria a existencia de ferramentas para conseguir, en tempo real, realizar unha chamada ao menor custo posible. Asterisk::LCR proporciona esta funcionalidade, aínda que con algunhas limitacións.

Asterisk::LCR (Least Cost Routing)

LCR son un conxunto de ferramentas Perl open-source pensadas para xestionar de xeito eficiente múltiples provedores de VoIP en Asterisk, de forma que sempre se seleccione o máis económico.

Configuración

Unha vez instalado o módulo Asterisk::LCR hai que escribir o ficheiro de configuración `/etc/asterisk-lcr.cfg` que será similar ao seguinte:

```
[storage]
package = Asterisk::LCR::Storage::DiskBlob.pm

[comparer]
package = Asterisk::LCR::Comparer::XERAND
currency = eur

[dialer]
package = Asterisk::LCR::Dialer::MinCost
locale = es
limit = 3

[import:voipjet]
package = Asterisk::LCR::Importer::VoIPJet
dial = us IAX2/login@voipjet/REPLACEME

[import:nufone]
package = Asterisk::LCR::Importer::NuFone
dial = us IAX2/login@NuFone/REPLACEME
```

O significado das distintas seccións é o seguinte:

- **[storage]**. Nesta sección defínense o mecanismo de almacenamento de datos. É posible usar algún xestor de base de datos como MySQL.

- **[comparer]**. Aquí defínese a estratexia de comparación. Pódese usar unha que, por exemplo, compares as tarifas sen ter en conta certos detalles, como o establecemento de chamada ou a facturación por minuto. A través do atributo `package` pódese cambiar a estratexia. De momento, só hai dous módulos (packages) de comparación escritos, XERAN e DUMMY. Para escribir módulos propios pódese consultar a páxina de manual de Asterisk::LCR::Comparer ^[1].
- **[dialer]**. Nesta sección inclúese a estratexia de marcación. Actualmente hai dúas, 'MinCost' que tenta minimizar o custo para unha ruta dada probando co operador máis económico ata o máis caro, cun límite de 3 provedores; 'MinTime' que minimiza o PDD (Post-Dialing-Delay ou Retardo-Post-Marcación) probando os tres provedores máis baratos ao mesmo tempo. Máis información sobre esta sección nesta páxina ^[2].
- **[import:XXX]**. Por último, defínense que tarifas de que provedores se quere importar / descargar. Hai unha sección para cada provedor e debe seguir a nomenclatura `[import:<nome>]`. O nome non se pode repetir en cada sección `import`. No exemplo hai dous provedores. No primeiros deles o parámetro 'dial' é unha plantilla, 'us' indica que VoIPjet utiliza o estilo de marcación US e `IAX2/login@voipjet/REPLACEME` hai que reempazalo cun login real. Tamén `REPLACEME` ten que reempazarse cos datos reais cando se xenere o dialplan. Suponse tamén que a definición do peer `voipjet` se atopa no ficheiro `iax.conf` baixo a sección `[voipjet]`. Os provedores soportados son:
 - Asterisk::LCR::Importer::NuFone
 - Asterisk::LCR::Importer::PlainVoIP
 - Asterisk::LCR::Importer::RichMedium
 - Asterisk::LCR::Importer::VoIPJet

Utilización

Crearase un directorio temporal onde copiar a ferramenta LCR:

```
mkdir /tmp/lcr
```

Unha vez que se teña escrito o ficheiro de configuración hai que seguir tres pasos:

- **Paso 1:** importar as tarifas dos provedores. O seguinte comando importa todos os provedores definidos na seccións `[provider:nome]` e gárdaos en disco nun formato normalizado:

```
cd /tmp/lcr
asterisk-lcr-import
```

- **Paso 2:** xerar a árbore LCR, que creará unha árbore de "`<prefix> => [lista de tarifas ordenadas]`" a partir das tarifas importadas no paso anterior:

```
cd /tmp/lcr
asterisk-lcr-build
```

- **Paso 3:** xerar un dialplan optimizado que se pode cortar e pegar (ou mellor incluír mediante `include`) no dialplan do Asterisk.

```
cd /tmp/lcr
asterisk-lcr-dialplan >/etc/asterisk/lcr-dialplan.conf
```

Localización da ferramenta

En relación coas opcións dependentes de cada idioma/país, Asterisk::LCR pode xerar dialplan adaptados a cada un. Os ficheiros para as opcións de localización están no cartafol `/lib/Asterisk/LCR/Locale/`.

Actualmente, hai dúas táboas de tradución implementadas: `us.txt` (para a marcación US-style) e `fr.txt` (para Francia + chamadas extranxeiras).

Información para desenvolvedores e descarga

Para descargar o código fonte e obter información para desenvolvedores temos as seguintes páxinas:

- Código fonte do Asterisk::LCR ^[3].
- Información para desenvolvedores ^[4].

Enlaces externos

- [1] <http://help-site.com/Programming/Languages/Perl/CPAN/Bundles/Asterisk-LCR/Comparer.pm/>
- [2] <http://help-site.com/Programming/Languages/Perl/CPAN/Bundles/Asterisk-LCR/Dialer.pm/>
- [3] <http://cpan.uwinnipeg.ca/cpan/authors/id/J/JH/JHIVER/Asterisk-LCR-0.08.tar.gz>
- [4] <http://cpan.uwinnipeg.ca/dist/Asterisk-LCR>

Fuente: http://informatica/manuais/index.php?title=Encamiñamento_de_chamadas_a_través_do_provedor_máis_económico&oldid=11915

Autores principais: Arribi,

Conexión de Asterisk con openLDAP

Conexión Asterisk con openldap.

Co propósito de manter a información dos usuarios/as nun directorio, evitando deste xeito a duplicación de datos conectaremos Asterisk con un servidor openldap, donde se almacenará basicamente a información que doutro xeito estaría nos ficheiros de configuración Asterisk `sip.conf` e `extensions.conf`.

Configuración ldap.

Instalaremos un servidor openldap no mesmo servidor onde corre Asterisk.

```
yum install openldap openldap-clients openldap-servers
```

Configuramos o servidor ldap cuns datos de exemplo para asterisk.

slapd.conf

Editamos o ficheiro `/etc/openldap/slapd.conf`

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include      /etc/openldap/schema/core.schema
include      /etc/openldap/schema/cosine.schema
include      /etc/openldap/schema/inetorgperson.schema
include      /etc/openldap/schema/nis.schema

# Allow LDAPv2 client connections. This is NOT the default.
allow bind_v2

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral    ldap://root.openldap.org

pidfile      /var/run/openldap/slapd.pid
argsfile     /var/run/openldap/slapd.args

# Load dynamic backend modules:
# modulepath /usr/lib/openldap

# modules available in openldap-servers-overlays RPM package:
# moduleload accesslog.la
# moduleload auditlog.la
# moduleload denyop.la
# moduleload dyngroup.la
# moduleload dynlist.la
# moduleload lastmod.la
# moduleload pcache.la
# moduleload ppolicy.la
# moduleload refint.la
# moduleload retcode.la
# moduleload rwm.la
# moduleload smb5pwd.la
# moduleload syncprov.la
# moduleload translucent.la
# moduleload unique.la
# moduleload valsort.la

# modules available in openldap-servers-sql RPM package:
# moduleload back_sql.la

# The next three lines allow use of TLS for encrypting connections
using a
```

```
# dummy test certificate which you can generate by changing to
# /etc/pki/tls/certs, running "make slapd.pem", and fixing permissions
on
# slapd.pem so that the ldap user or group can read it. Your client
software
# may balk at self-signed certificates, however.
# TLSCACertificateFile /etc/pki/tls/certs/ca-bundle.crt
# TLSCertificateFile /etc/pki/tls/certs/slapd.pem
# TLSCertificateKeyFile /etc/pki/tls/certs/slapd.pem

# Sample security restrictions
#     Require integrity protection (prevent hijacking)
#     Require 112-bit (3DES or better) encryption for updates
#     Require 63-bit encryption for simple bind
# security ssf=1 update_ssf=112 simple_bind=64

# Sample access control policy:
#     Root DSE: allow anyone to read it
#     Subschema (sub)entry DSE: allow anyone to read it
#     Other DSEs:
#         Allow self write access
#         Allow authenticated users read access
#         Allow anonymous users to authenticate
#     Directives needed to implement policy:
# access to dn.base="" by * read
# access to dn.base="cn=Subschema" by * read
# access to *
#     by self write
#     by users read
#     by anonymous auth
#
# if no access controls are present, the default policy
# allows anyone and everyone to read anything but restricts
# updates to rootdn. (e.g., "access to * by * read")
#
# rootdn can always read and write EVERYTHING!

#####
# ldbm and/or bdb database definitions
#####

database            bdb
suffix              "dc=my-domain,dc=com"
rootdn              "cn=Manager,dc=my-domain,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoided. See slapd.conf(5) and slapd.passwd(8) for details.
# Use of strong authentication encouraged.
```

```

# rootpw          secret
# rootpw          {crypt}ijFYncSNctBYg

# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory        /var/lib/ldap

# Indices to maintain for this database
index objectClass          eq,pres
index ou,cn,mail,surname,givenname  eq,pres,sub
index uidNumber,gidNumber,loginShell eq,pres
index uid,memberUid        eq,pres,sub
index nisMapName,nisMapEntry      eq,pres,sub

# Replicas of this database
#repllogfile /var/lib/ldap/openldap-master-replog
#replica host=ldap-1.example.com:389 starttls=critical
#   bindmethod=sasl saslmech=GSSAPI
#   authcId=host/ldap-master.example.com@EXAMPLE.COM

```

Modificaremos as seguintes liñas:

Engadimos a seguinte liña a lista de includes, co esquema LDAP necesario para Asterisk.

```
include        /etc/openldap/schema/asterisk.schema
```

Previamente debemos copiar o ficheiro asterisk.schema o directorio schema de openldap

```

[root@hercules scripts]# cp
usr/src/asterisk/asterisk-1.6.1.0/contrib/scripts/asterisk.schema
/etc/openldap/schema/

```

Modificamos a liñas

```

suffix        "dc=iessancllemente,dc=net"
rootdn        "cn=Manager,dc=iessancllemente,dc=net"
...
rootpw        {SSHA}p1aUQtijSsqYnsMjaJgyPrV58RYngkc
...
directory     /var/lib/ldap/iessancllemente-ldap

```

O password do *rootdn* obtémolo usando a ferramenta

```

[root@hercules openldap]# slappasswd
New password:
Re-enter new password:
{SSHA}p1aUQtijSsqYnsMjaJgyPrV58RYngkc
[root@hercules openldap]#

```

Tamén temos que crear o directorio `/var/lib/ldap/iessanclemente-net` que conterá a base de datos LDAP. Faremos este directorio pertencente o usuario `ldap:ldap` cos permisos 700 tal e como se indica no ficheiro `slapd.conf` Para que surtan efecto estes cambios rearrancamos o servidor ldap.

```
[root@hercules ldap]# /etc/init.d/ldap restart
```

Neste punto só nos queda cargar algúns datos de exemplo na base de datos LDAP. Faremos isto mediante ficheiros ldif. A continuación detallamos os datos de exemplo que cargaremos, así como os ficheiros necesarios e os comandos ldap para facer a carga.

sippeers.ldif

```
dn: ou=sippeers,dc=iessanclemente,dc=net
ou: sippeers
objectClass: top
objectClass: organizationalUnit

dn: cn=Antonio Perez,ou=sippeers,dc=iessanclemente,dc=net
objectClass: top
objectClass: inetOrgPerson
objectClass: AsteriskSIPUser
cn: Antonio Perez
sn: Perez
AstAccountCallerID: 2001
AstAccountHost: dynamic
AstAccountRealmedPassword: {SSHA}9aenZD/V5WDUIQdDkd5llKej0QHN09sq
AstAccountContext: default

dn: cn=María Arias,ou=sippeers,dc=iessanclemente,dc=net
objectClass: top
objectClass: inetOrgPerson
objectClass: AsteriskSIPUser
cn: María Arias
sn: Arias
AstAccountCallerID: 2002
AstAccountHost: dynamic
AstAccountRealmedPassword: {SSHA}sCtnyggFBp8oxWE61KZAE2XTcIiJ63G
AstAccountContext: default
```

Ficheiro ldif para crear dous usuarios SIP (Antonio Perez e María Arias) coas extensións 2001 e 2002 respectivamente.

extensions.ldif

```
dn: ou=extensions,dc=iessanclemente,dc=net
ou: extensions
objectClass: top
objectClass: organizationalUnit
```



```
dn: cn=2001,ou=extensions,dc=iessanclemente,dc=net
cn: 2001
objectClass: top
objectClass: device
objectClass: AsteriskExtension
AstContext: users
AstExtension: 2001
AstPriority: 1
AstApplication: Dial
AstApplicationData: SIP/antonio perez
```

```
dn: cn=2002,ou=extensions,dc=iessanclemente,dc=net
cn: 2002
objectClass: top
objectClass: device
objectClass: AsteriskExtension
AstContext: users
AstExtension: 2002
AstPriority: 1
AstApplication: Dial
AstApplicationData: SIP/maria arias
```

Ficheiro ldif para crear 2 extensións 2001 e 2002 que utilizaremos para chamar aos dous usuarios anteriormente dados de alta.

eco-demo.ldif

```
dn: cn=600-1,ou=extensions,dc=iessanclemente,dc=net
cn: 600-1
objectClass: top
objectClass: device
objectClass: AsteriskExtension
AstContext: demo
AstExtension: 600
AstPriority: 1
AstApplication: Playback
AstApplicationData: demo-echotest
```

```
dn: cn=600-2,ou=extensions,dc=iessanclemente,dc=net
cn: 600-2
objectClass: top
objectClass: device
objectClass: AsteriskExtension
AstContext: demo
AstExtension: 600
AstPriority: 2
AstApplication: Echo
```

```
dn: cn=600-3,ou=extensions,dc=iessanclemente,dc=net
cn: 600-3
objectClass: top
objectClass: device
objectClass: AsteriskExtension
AstContext: demo
AstExtension: 600
AstPriority: 3
AstApplication: Playback
AstApplicationData: demo-echodone
```

Ficheiro ldif co plan dunha proba de echo para a extensión 600.

Carga dos datos de exemplo

A continuación cargamos en LDAP os anteriores ficheiros de configuración utilizando o comando ldapadd. Exemplo de carga das extensións.

```
[root@hercules ldap]# ldapadd -x -D
"cn=Manager,dc=iessanclemente,dc=net" -W -f extensions.ldif
Enter LDAP Password:
```

Configuración Asterisk

Editamos o ficheiro `/etc/asterisk/res_ldap.conf` e dentro da opción `[_general]` establecemos os seguintes parámetros

```
[_general]
;
; Specify one of either host and port OR url.  URL is preferred, as you
; can
; use more options.
host=127.0.0.1 ; LDAP host
port=389
url=ldap://localhost:389
protocol=3
basedn="dc=iessanclemente, dc=net" ; Base DN
user="cn=Manager,dc=iessanclemente,dc=net" ; Bind DN
pass=abc123.
```

Nos arpartados `host` e `url` establecemos o enderezo onde está instalado o servidor openldap. Neste caso está no mesmo servidor.

Editamos o ficheiro `/etc/asterisk/extconfig.conf` e engadimos as seguintes liñas ao final

```
sipusers => ldap,"dc=iessanclemente,dc=net",sip
sippeers => ldap,"dc=iessanclemente,dc=net",sip
voicemail => ldap,"dc=iessanclemente,dc=net",voicemail
voicemail_data => ldap,"dc=iessanclemente,dc=net",voicemail
extensions => ldap,"dc=iessanclemente,dc=net",extensions
queues => ldap,"dc=iessanclemente,dc=net",queue
```

```
queue_members => ldap,"dc=iessanclamente,dc=net",queue_member
musiconhold => mysql,asterisk
queue_log => mysql,asterisk
meetme => ldap,"dc=iessanclamente,dc=net",meetme
```

Editamos o ficheiro `/etc/asterisk/extensions.conf` para incluir as seguintes liñas

```
[users]
switch => Realtime/@

[demo]
switch => Realtime/@

[default]
include => users
include => demo
```

A versión 1.6 de Asterisk xa ven co módulo LDAP realtime interface, de todas maneiras comprobamos que este módulo está cargado no noso asterisk

```
hercules*CLI> module show like ldap
Module                Description
                    Use Count
res_config_ldap.so    LDAP realtime interface
                    0
1 modules loaded
hercules*CLI>
```

Fuente: http://informatica/manuais/index.php?title=Conexión_de_Asterisk_con_openLDAP&oldid=11696

Autores principais: Ramiro,

Sincronización de tempo

Para manter o reloxo do noso servidor sincronizado poderemos empregar un script cron do seguinte estilo:

```
#!/bin/bash
ntpdate es.pool.ntp.org # servidor de hora español
/sbin/hwclock --systohc # sincronizamos o reloxo hardware
```

Fuente: http://informatica/manuais/index.php?title=Sincronización_de_tempo&oldid=11527

Autores principais: Veiga,

Información do estado de Asterisk en Tempo Real e vía web

FPO - The Flash Operator Panel

FOP presenta un xeito visual e vía web de comprobar o estado do sistema Asterisk.

FOP é comúnmente empregado para ver en tempo real aos usuarios do sistema e conectar chamadas entre eles. Tamén pode empregarse para avisar de cando un usuario está activo ou non.

A web de FOP é www.asternic.org ^[1].

Enlaces externos

[1] <http://www.asternic.org>

Fuente: http://informatica/manuais/index.php?title=Información_do_estado_de_Asterisk_en_Tempo_Real_e_vía_web&oldid=11941

Autores principais: Veiga, Vieites,

Elementos hardware dos Sistema de VOIP

Neste apartado veremos todo o *hardware* preciso para implementar un sistema con Asterisk.

Tamaño dun sistema con Asterisk

O tamaño dun sistema con Asterisk non depende do número de usuarios, senón polo número de chamadas simultáneas que se poidan dar.

Propósito	Número de chamadas	Mínimo recomendado
MiniSistema	Non mais de 5	400MHz x86, 256 MB RAM
Pequena Oficina	Entre 5 e 10	1 GHz x86, 512 MB RAM
Mediana Oficina	Menos de 25	3 GHz x86, 1 GB RAM
Gran Oficina	Mais de 25	CPU Dual

Se a nosa intención é conectar o Asterisk a un sistema de comunicacións tradicional precisaremos do *hardware* adecuado. O *hardware* preciso dependerá do sistema ao que vaíamos a conectalo:

Conexión á PSTN (*Public Switched Telephone Network*)

Neste intre, o xeito mais efectivo e popular de conectar Asterisk á **PSTN** é empregando tarxetas de expansión desenroladas polo Zapata Telephony Project ^[1].

A continuación vemos os tipos de conexión existentes e os dispositivos empregados:

Rede Telefónica Básica (RTB)

A instalación básica de **telefonía analóxica (RTB)** correspóndese co esquema de bloques amosados na figura da dereita.

A instalación de abonado está directamente unida á central local mediante un circuíto de 2 fíos, xeralmente de cobre. A terminación de central consta dos elementos de conmutación e a terminación de liña a forman os equipos de transmisión e recepción.

O **punto de terminación de rede (PTR)** que a compañía instala na vivenda do abonado, proporciona proteccions contra sobretensions que puideran danar os equipos e establece os límites legais entre as responsabilidades da compañía e as do abonado. Incorpora unha pestaña deslizante que, ao levantala deixa cortada a liña telefónica. Debaxo dela dispón dun conector para un teléfono coa fin de determinar se a avaría está antes do PTR, ou na instalación de usuario.

A **roseta ou base de acceso ao terminal (BAT)** consta dun conector femia 6P2C, 6P4C o 6P6C, popularmente nomeado **RJ-11**, ao que se conectará o latiguillo do terminal telefónico. Poden ser empotrables ou de superficie.

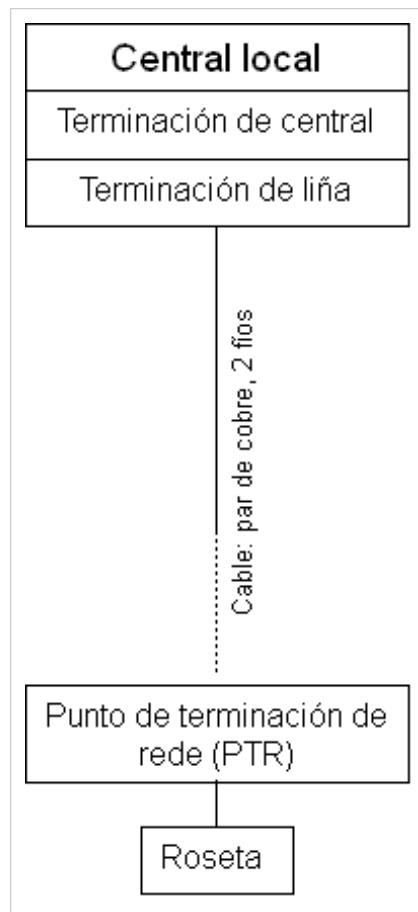
O sistema complétase có **terminal telefónico** que se conecta á roseta: un teléfono analóxico, un fax, etc.

- **Comunicacións de datos a través da liña RTB**

O sistema descrito anteriormente é analóxico, mentres que os datos xerados nun ordenador (DTE: equipo terminal de datos) son dixitais. Para facer posible a comunicación de datos a través da rede RTB, é necesario insertar un equipo de comunicacións de datos (DCE) que teña en conta a dualidade analóxico-dixital do proceso, o modem.

Un **modem (modulador-demodulador)** transforma, no extremo emisor (modulador), o sinal dixital xerado polo ordenador (dato) nun sinal analóxico que poida ser transmitido pola liña telefónica analóxica e, no extremo receptor (demodulador), o sinal analóxico recibido polo equipo nun sinal dixital que poida ser recoñecida polo ordenador.

Conectando directamente o *modem* á liña telefónica acadáronse velocidades de transmisión ata 56 kpps. O usuario doméstico demanda velocidades máis altas comparables ás que as liñas dixitais e o cable ofrecen en empresas o centros de



ensinanza. A solución é a **ADSL (*Asynchronous Digital Subscriber Lines*)**; a tecnoloxía básica pode alcanzar velocidades de 8 Mbps, máis rápida que as redes dixitais RDSI, e máis barata que a instalación con acceso por cable. As melloras que ofrecen os novos desenvolvementos ADSL2+ permiten acadar velocidades de 24 Mbps. En ADSL emprégase unha modulación dos sinais de datos con frecuencias superiores ás utilizadas nas conversacións telefónicas convencionais (300-3.400 Hz), función que realiza o router ADSL.

A instalación da ADSL require o emprego dun filtro (*splitter*) para evitar distorsións nos sinais transmitidos; o filtro separa o sinal de voz convencional dos sinais moduladas da conexión de datos ADSL. Dende a roseta, mediante un adaptador 6P4C dobre, saen 2 cables, un directamente ao modem ADSL, e outro ao terminal telefónico no que se intercala o filtro.

Na liña ADSL establécense 3 canles de comunicación, que son: o de servizo telefónico normal, o de envío de datos e o de recepción de datos. A velocidade de recepción (dende a rede ata o abonado) é maior que a velocidade de envío (dende o abonado ata a rede); dícese que a liña é asimétrica.

A ADSL ofrece a posibilidade de empregar voz IP para chamadas a longa distancia que eran moi custosas fai uns anos, de tal xeito que os operadores telefónicos ofrecen tarifas planas para o seu uso.

Non se recomenda instalar ADSL cando a lonxitude de liña entre abonado e central é superior a 3 km.

- **Tarxetas de expansión analóxicas**

Empregando o hardware axeitado que se proporciona como tarxetas de ampliación para insertar na ranura dun PC é posible montar un sistema similar a una central telefónica privada (PBX), é dicir, unha rede de comunicacións interna en unha empresa con un determinado número de extensións, en función do número de terminais que se queren instalar.

Senón precisamos moitas canles, ou temos poucos cartos, a nosa PSTN pode consistir en un ou varios circuitos analóxicos. Cada un deles precisará dun porto FXO (Foreing eXchange Office).

O ordenador necesita ter instalado o software Asterisk e para conectar á rede RTB precísanse tarxetas de comunicación analóxicas; son varios os fabricantes que as comercializan entre eles Digium ^[2] que desenvolveu Asterisk. As tarxetas teñen varios conectores hembra 6P6C, nos que segundo a configuración do sistema pódense conectar a liña telefónica ou terminais analóxicos.

Outras compañías que tamén fabrican tarxetas compatibles con Asterisk para canles analóxicos serían:

- Junghanns ^[3]
- Beronet ^[4]
- Rhino ^[5]
- Sangoma ^[6]
- Voicetronix ^[7]
- Pika Technologies ^[8]

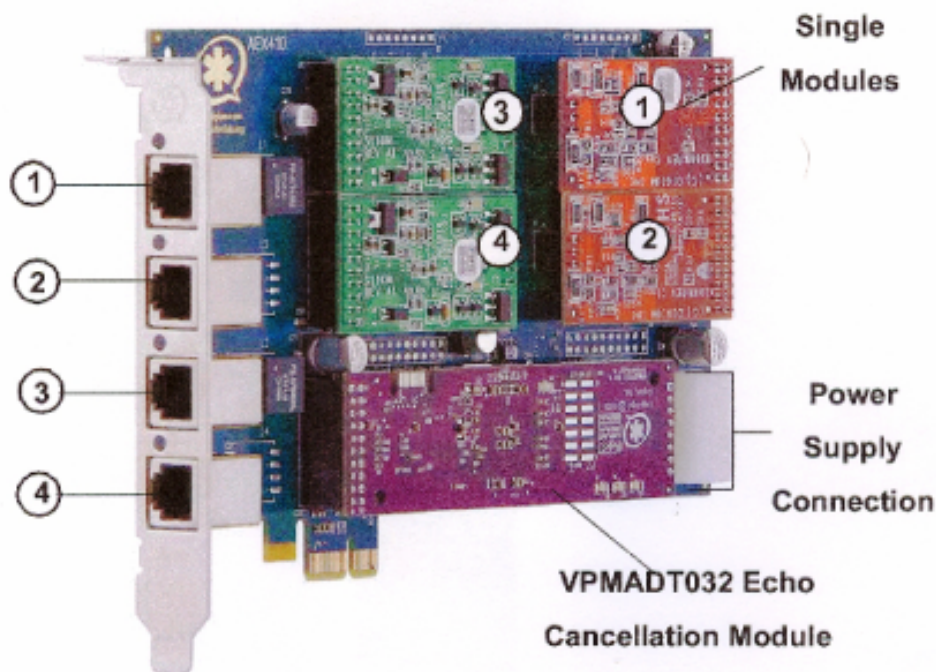
A configuración *hardware* realízase engadindo á tarxeta base módulos de comunicacións: **Os módulos FXO** comunican a tarxeta coa liña telefónica RTB. Empregaranse tantos módulos deste tipo como liñas RTB se contraten: a maior nº de liñas maior nº de conversacións telefónicas poden manter simultaneamente co exterior as extensións da rede de comunicación.

Os módulos FXS comunican a tarxeta cós terminais telefónicos. Precisaranse tantos módulos FXS como extensións telefónicas inclúa a rede de comunicación.

É evidente que aumentar o nº de liñas telefónicas contratadas ou o nº de terminais analóxicos da rede de comunicación implica incrementar o nº de módulos que habería que insertar na tarxeta, e polo tanto o coste do sistema.

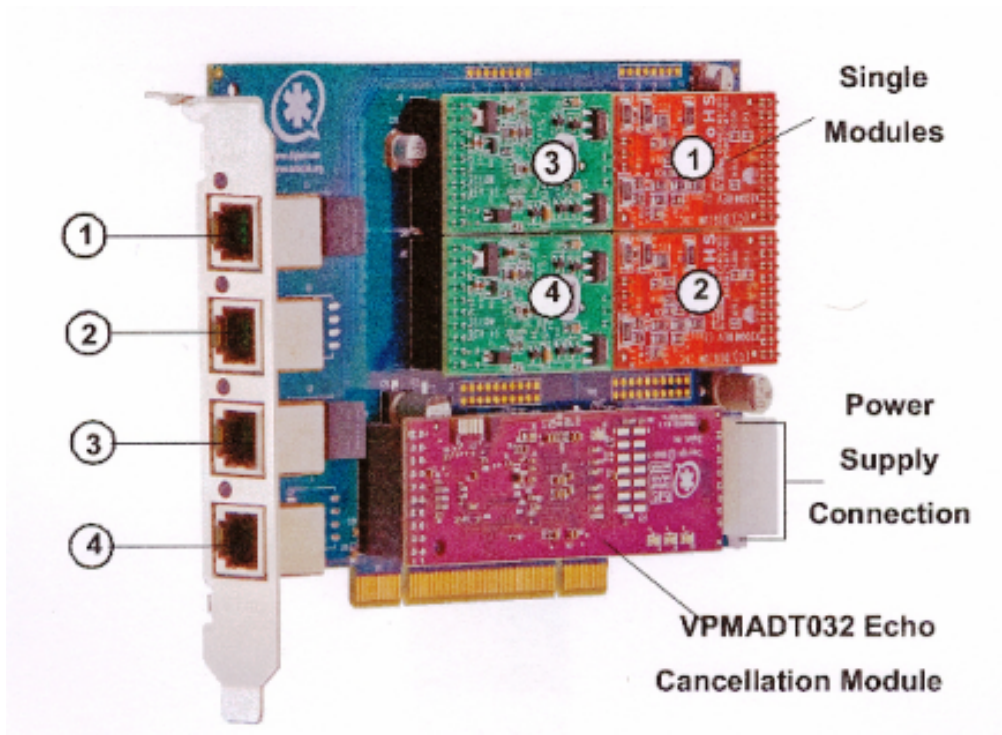
Tamén se deduce do exposto que o nº de conectores RJ11 da tarxeta debe ser maior ou igual que a suma de liñas RTB de entrada máis o nº de extensións da rede de comunicación que se planifique.

A tarxeta analóxica que seleccionamos para as nosas probas é algún dos modelos de **Digium**: TDM410 (slot PCI) ou AEX410 (slot PCI Express x1).



As tarxetas XXX410 incorporan 4 zócalos nos que se montan 4 módulos FXS ou FXO e que se corresponden cós 4 conectores RJ11 da tarxeta. Os módulos se identifican pola súa cor, os módulos FXO son vermellos e os FXS son verdes.

A tarxeta tamén incorpora un zócalo onde se pode montar un cancelador de eco *hardware* como amosan as imaxes. Digium proporciona coas súas tarxetas un cancelador de eco software sen coste adicional mais non ofrece o mesmo rendemento que o módulo *hardware*.



O usuario configura o nº de módulos segundo as súas necesidades; posto que o noso obxectivo é conectar Asterisk á unha soa rede RTB, a lo menos precisamos un módulo FXO, por exemplo, Digium dispón do módulo FXO-X100M.

Os módulos FXS non son imprescindibles para as probas xa que se o ordenador se conecta a unha rede local, por exemplo, a través do modem-router ADSL, nesta rede de datos podemos conectar os terminais de voz sobre IP.

- Configuración AEX410

A Rede Dixital de Servizos Integrados (RDSI)

O CCITT (Comité Consultivo Internacional para Telegrafía e Telefonía), define a RDSI como: “Rede que procede por evolución da Rede Dixital Integrada e que facilita conexións dixitais extremo a extremo para proporcionar unha ampla gama de servizos, tanto de voz como doutros tipos, e á que os usuarios acceden a través dun conxunto definido de interfaces normalizados”.

Polo tanto é unha rede que procede da rede telefónica básica.

Está baseada en conexións por conmutación de circuitos a 64 kbps, se ben no futuro incorporará elementos de conmutación de paquetes, para así converterse paulatinamente na rede única que integre os servizos actuais.

Mentres dure este proceso de evolución, a RDSI coexistirá coas redes convencionais en telefonía e datos, pero incorporando elementos de interconexión con ditas redes.

A RDSI considerada ata agora, denomínase de Banda Estreita (RDSI-BE) posto que traballa con conexións conmutadas a 64 kbps, podendo chegar no futuro próximo ata os 2 Mbps, coa mesma tecnoloxía.

Estando a dar os primeiros pasos, para que a RDSI, incorporando elementos conmutadores a velocidades superiores, propicie a aparición doutra nova xeración de servizos, por exemplo, distribución de programas de TV (Imagenio), videotelefonía de alta calidade ou transmisión de datos a moi alta velocidade; temos así a RDSI de Banda Ancha (RDSI-BA).

• Acceso do abonado á RDSI

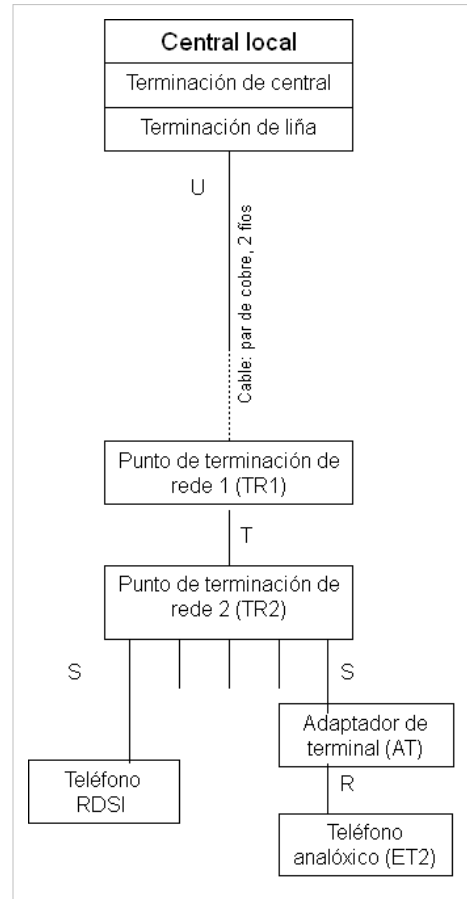
Na figura da dereita, amósase a configuración de referencia do acceso de usuario á RDSI, definida polo CCITT, na que se sinala o conxunto de **puntos de referencia** e de **agrupacións funcionais** que se consideraron dentro do proceso de normalización destes accesos.

Os puntos de referencia son as separacións entre funcionalidades distintas identificadas nas agrupacións funcionais, e poden representar interfaces reais ou virtuais.

-Punto de referencia S: Representa o interface de usuario-rede e constitúe o punto de conexión física dos terminais de abonado á RDSI. É un interface a catro fíos, dous para transmisión e dous para recepción.

-Punto de referencia T: Representa a separación entre as instalacións do usuario e os equipos de transmisión de liña. Posúe as mesmas características mecánicas e eléctricas que o interface S.

-Punto de referencia U: Define a liña de transmisión entre as dependenzas do abonado e a central telefónica, e correspóndese fisicamente có bucle de abonado analóxico a dous fíos actualmente existente.



-Punto de referencia V: Representa a separación entre os elementos de transmisión e os de conmutación dentro da central local RDSI.

-Punto de referencia R: É o punto de conexión de calquera terminal que soporte un interface normalizado, por exemplo, os terminais con interface analóxico a dous fíos. Desta maneira e mediante o emprego do adaptador correspondente, pódense conectar os terminais da RTB ao entorno da RDSI.

As agrupacións funcionais que aparecen na configuración de referencia, representan distintos niveis de funcións entrelazadas mediante o interface que representa cada punto de referencia, e pode corresponder a un equipo físico ou simplemente a unha parte ou función do mesmo.

-ET1: É o equipo terminal que soporta directamente o interface S, é dicir, está deseñado para conectarse directamente á RDSI. Como exemplo poden citarse os teléfonos dixitais RDSI, Fax Grupo IV con interface S, etc.

-ET2: Representa calquera terminal que non se conecta directamente ao interface de usuario-rede, da RDSI, como os terminais modo paquete, terminais analóxicos a dous fíos, etc.

-AT: Este equipo conecta por un lado un interface R determinado e por outro o interface S do acceso do usuario, e permite, por tanto, a conexión dos ET2 comentados anteriormente, á RDSI. Como exemplo pódense citar o adaptador analóxico AT a/b, para terminais analóxicos a dous fíos e o adaptador AT X.25 para terminais modo paquete.

-TR1: Establece a separación física entre instalacións de usuario e a rede exterior, e realiza as funcións de transmisión entre estas e a central.

-TR2: Realiza as funcións de conmutación e control no interior das instalacións de abonado. A TR2 podería ser unha centralita, unha rede de área local, etc., dependendo do tamaño das instalacións. No caso máis sinxelo, podería chegar a desaparecer, coincidendo entón fisicamente os interfaces S e T.

• Canles de acceso

Para a transferencia de información e sinalización definíronse na RDSI os seguintes tipos de canles ou vías de transferencia de información:

- **Canle B:** canle de 64 kbps que transporta información xerada polo terminal do usuario.

- **Canle D:** canle de 16 kbps ou 64 kbps, dependendo da estrutura do acceso do abonado, que se utiliza para transportar a sinalización no interface usuario-rede. Tamén pode usarse para transmitir información do usuario a baixa velocidade.

- **Canle H:** Permite a transferencia de información de usuario a velocidades superiores a 64 kbps. Existen tres modos de canle H:

- **Canle H0 de 384 kbps,** equivale a 6 canles de 64 kbps.

- **Canle H11 de 1.536 kbps,** equivale a 24 canles de 64 kbps, utilizable en países con xerarquía dixital a 1.544 kbps. (USA e Xapón).

- **Canle H12 de 1.920 kbps,** equivale a 30 canles de 64 kbps, utilizable en países con xerarquía dixital a 2 Mbps.

• Estrutura do acceso de abonado

As canles de información e sinalización poderían combinarse para formar diversos tipos de acceso de usuario. Polo momento normalizáronse dous estruturas de acceso diferentes, atendendo ao número e tipo de canles de información e sinalización que conteñan, e se denominan acceso básico e acceso primario.

- **Acceso Básico:** Está constituído por dúas canles B a 64 kbps para a transmisión de información, e unha canle D a 16 kbps para a sinalización de usuario.

No lado das instalacións de usuario (Interface S/T), ofrece unha velocidade de transmisión total de 192 kbps (B+B+D+ Control+ Sincronismo+ Mantemento), e está montado nunha configuración a catro fíos, dous para transmisión e dous para recepción.

- **Acceso Primario:** Está constituído por 30 canles B e unha canle D a 64 kbps, cunha velocidade total de 2 Mbps.

No lado das instalacións de usuario aparece "o interface I ", que pode ser totalmente equivalente ao S ao presentar una estrutura 30B+D, dependendo do seu tamaño, e que se utiliza para unir a unidade funcional TR1, interface de línea, coa TR2, normalmente unha centralita dixital, cuas extensións poden ser liñas de interface S.

No lado de rede, está soportado por un sistema de transmisión MIC a 2 Mbps.

Dado o coste deste servizo soamente é contratado polas grandes empresas.

- **Tarxetas de expansión dixitais**

Se precisamos mais de 10 circuitos, ou conectividade dixital, debemos de ter unha tarxeta T1 ou E1.

Os fabricantes **Digium**, **Zaptel**, **Sangoma** e **Rhino** teñen no seu catálogo tarxetas dixitais.

- mISDN ^[9]

Channel banks

Un *channel bank* é un dispositivo que permite demultiplexar un circuito dixital en varios circuitos analóxicos (ou vice versa). Mais especificamente, un *channel bank* permitirá conectar liñas e teléfonos analóxicos a un sistema ao través dunha liña T1.

Os *channel bank* son dispositivos realmente caros.



Outros tipos de interfaces PSTN

Outro xeito de conectarse á PSTN é pola RDSI. Este tipo de conexión son dúas canles a 144kbps pero, o realmente importante para nós é que **Digium** fabrica as tarxetas B410P [10], TE412P [11] TE410P [12], TE407P [13] e TE405P [14] para conectar o sistema Asterisk á RDSI.

Conexión a unha Rede Telefónica Baseada en Paquetes

Se non se precisa conectar á PSTN, Asterisk só precisa de *hardware* unha tarxeta de rede (NIC).

Sen embargo, se queremos proporcionar *music on hold* ou conferencias e non se ten ningunha fonte de tempo precisaríamos o **módulo de Linux *ztdummy***, que é unha especie de "metrónomo" que permite ao sistema misturar múltiples correntes de audio dun xeito correcto e sincronizado.

Cancelación do ECO

Un dos problemas que vai surxir cando se conecten interfaces analóxicos sobre un sistema de VoIP é o **eco**.

O **eco** é simplemente que, o que un interlocutor di lle chega reflexado un tempo mais tarde. Existe cancelación de eco por *hardware* e por *software* pero, sen dúbida, é mellor a primeira opción.

A **cancelación de eco por *hardware*** fai mais caro o noso sistema, pero vale a pena pola calidade do son que se vai a obter. A maioría das tarxetas PSTN de Digium veñen con cancelación de eco por *hardware*.

Tipos de Teléfonos

Como é lóxico nun sistema de VoIP teñen que existir os teléfonos como elementos finais cós que os usuarios se comunican. Iso si, existen moitos "tipos" de teléfonos que se poden integrar có Asterisk.

Teléfonos Físicos analóxicos

Os Teléfonos analóxicos existiron dende a invención do teléfono [15] (1.871). Ata fai uns 20 anos todos os teléfonos eran analóxicos.

Cando unha persoa fala, as cordas vocais, a lingua, os dentes, e os labos crean unha variedade complexa de sons. O obxectivo do teléfono é de capturar estes sons e convertilos nun formato conveniente para a transmisión sobre cables. Nun teléfono analóxico, a sinal transmitida é análoga ás ondas sonoras producidas pola oratoria da persoa.

Os Teléfonos Analóxicos son a única clase de teléfono que está comúnmente dispoñible en calquera tenda de electrónica de venda ao público. Isto cambiará radicalmente nos próximos anos...



Teléfonos Físicos IP

O teléfono IP (ou Terminal IP) é o principal dispositivo utilizado e especificamente deseñado para o seu uso en VoIP, e que permite realizar unha comunicación utilizando unha rede IP xa sexa mediante rede de área local (LAN) ou a través de Internet. O teléfono IP convirte e comprime a sinal da voz en paquetes de datos que serán enviados na rede IP, en lugar de utilizar unha conexión de rede telefónica.

- Linksys SPA942

Softphones

Un Softphone é un programa que ofrece a funcionalidade dun teléfono nun dispositivo que non o é, como pode ser un PC ou unha PDA.

Pouco a pouco nos imos alonxando do teléfono tradicional e tendendo cara unha cultura de comunicacións mais multimedia, así, por exemplo, poderíase comprar un terminal de comunicacións e instalarlle un *softphone* para proporcionarlle todas as funcións que desexemos.

Para os nosos obxectivos, podemos definir un *softphone* como calquera dispositivo que se executa sobre un ordenador personal, ten a aparencia e se comporta como un teléfono, e proporciona como a súa función primaria a capacidade de facer e recibir chamadas.

- Zoiper IAX e SIP Softphone
- X-Lite SIP Softphone

Adaptadores de teléfonos

Un "adaptador de teléfono" (normalmente referenciado como **ATA - Analog Terminal Adaptor** -) pode ser descrito como un dispositivo que fai unha conversión de protocolos de comunicación. Mais comunmente, estes dispositivos empréganse para converter un sinal dixital nun analóxico (é ao revés), para conectar un dispositivo IP, por exemplo, nun teléfono analóxico ou un fax.

- Linksys SPA3102

Terminais de comunicación

Este concepto define "calquera tipo de equipo" que se empregue para facer chamadas: PDA, PC, Teléfono Analóxico, Teléfono IP,...

Enlaces externos

- [1] <http://www.zapatatelephony.org/>
- [2] <http://www.digium.com/en/>
- [3] <http://www.junghanns.net/en/home.html>
- [4] <http://www.beronet.com/>
- [5] <http://www.channelbanks.com/>
- [6] <http://www.sangoma.com/>
- [7] <http://www.voicetronix.com/>
- [8] <http://www.pikatechnologies.com/english/view.asp?x=1>
- [9] http://www.misdn.org/index.php/Main_Page
- [10] <http://www.digium.com/en/products/digital/b410p.php>
- [11] <http://www.digium.com/en/products/digital/te412p.php>

- [12] <http://www.digium.com/en/products/digital/te410p.php>
- [13] <http://www.digium.com/en/products/digital/te407p.php>
- [14] <http://www.digium.com/en/products/digital/te405p.php>
- [15] <http://es.wikipedia.org/wiki/Tel%C3%A9fono>

Fuente: http://informatica/manuais/index.php?title=Elementos_hardware_dos_Sistema_de_VOIP&oldid=11961

Autores principales: Vieites,

Caso de Estudio: VoIP nun Centro Educativo

Escenario

O IES San Clemente é un centro educativo situado na rúa San Clemente s/n en Santiago de Compostela.

O claustro está composto de 90 e pico profesores, temos cerca de 25 departamentos conectados todos á centralita dixital de telefónica. En cada departamento hai conexións de rede RJ45 ó centro de cálculo san clemente onde se atopan os servidores centrais, firewall, routers, balanceador, ADSLs, etc.

Hai 4 liñas dispoñibles de entrada ó centro, e unha saída que desvía as chamadas a móbiles a través dela.

Deséxase dar servizo de voip e ofrecer servizos de centralita automática, que permita contactar con departamentos e profesores directamente sin ter que falar con ninguén, desvío intelixente de chamadas, servizos avanzados de información, buzóns de voz, etc.

Discusión

Asterisk é a solución apropiada para o IES San Clemente. Asterisk proporciona todas as características que ten actualmente a centralita dixital do IES. Aínda que non se vai a suprimir o uso desa centralita mediante Asterisk vanse a integrar as comunicacións voip e as liñas actuais.

Asterisk é unha ferramenta moi potente e relativamente barata. Simplemente temos que configurar un equipo que faga o servizo de centralita e empregar tarxetas adicionais de conexión á centralita como por exemplo a tarxeta Digium T100P que dispón de sinalización ISDN e se integra moi ben con Asterisk.

Nos equipos dos departamentos empregaremos clientes voip software ou teléfonos voip como Linksys.

Propostas

Arquivos de configuración:

1. Arquivo chan_dahdi.conf
2. → Arquivo dahdi-channels.conf
3. → Arquivo extensions.conf
4. → Arquivo iax.conf
5. → Arquivo indications.conf
6. → Arquivo meetme.conf
7. → Arquivo musiconhold.conf
8. → Arquivo queues.conf
9. Arquivo res_ldap.conf
10. → Arquivo sip.conf
11. → Arquivo system.conf
12. → Arquivo voicemail.conf

Fuente: http://informatica/manuais/index.php?title=Caso_de_Estudo:_VoIP_nun_Centro_Educativo&oldid=12117

Autores principais: Veiga,

Enlaces sobre programación e utilidades sobre Asterisk

Clientes Web IAX/SIP para Asterisk

A necesidade principal é dispor dun softwaree que se execute dende un navegador Web e que implemente os protocolos AIX e/ou SIP. As vantaxes son obvias:

- Independencia do sistema operativo.
- Non se necesita instalar nada.
- Facilidade de uso ao ser unha interface Web.
- Accesible dende calquera computador con acceso á rede.

Non é doado atopar alternativas libres. Existen pola contra algúns programas gratuítos ou de proba, e bastantes de pago. Algunhas alternativas son as seguintes:

- Libres:
 - IAXWebPhone ^[1].
- Gratuítos:
 - Flaphone ^[2].
 - DODDLE Web-Driven SIP Phone. Non é libre pero é gratuíto. Funciona en Windows e Linux e practicamente con calquera navegador.
- De pago:
 - Thinkingvoice ^[3].

Programación AGI PHP para Asterisk

http://www.voip.unam.mx/mediawiki/index.php?title=Ejemplo_de_AGI_PHP_para_Asterisk&printable=yes

Enlaces externos

- [1] <http://www.silicontechnix.com/iaxcontrol/webTelefone.zip>
- [2] <http://www.flaphone.com/>
- [3] <http://thinkingvoice.com/>

Fuente: http://informatica/manuais/index.php?title=Enlaces_sobre_programación_e_utilidades_sobre_Asterisk&oldid=11910

Autores principales: Arribi, Veiga, Vieites,

Membros do Grupo de Trabajo VoIP

Membros do Grupo de Trabajo:

- Arribi Vilela, Jesús
- Carrión Álvarez, Carlos
- García Seco, José
- Pérez Gómez, Catalina (*Coordinadora*)
- Seoane López, Cristina
- Vázquez López, Ramiro
- Veiga Cid, Rafael
- Vieites Rodríguez, Manuel

Relator:

- José Antonio Zumalave Rivas da empresa Level Telecom ^[1]

Enlaces externos

- [1] <http://www.leveltelecom.es/>

Fuente: http://informatica/manuais/index.php?title=Membros_do_Grupo_de_Trabalho_VoIP&oldid=12120

Autores principales: Vieites,
