

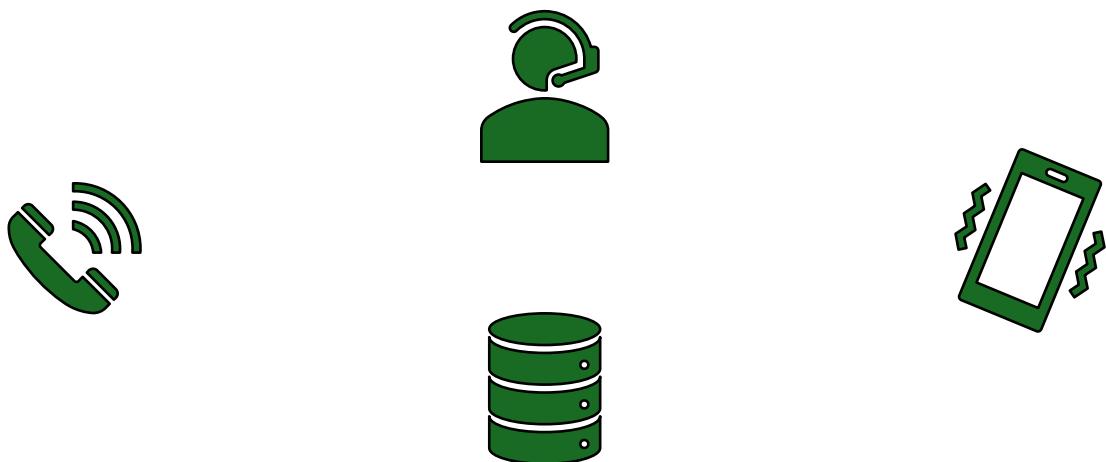
Marchel



Practica 2

Configuración de Sistemas de Telefonía IP

Asterisk



Índice de contenido

Introducción:	- 2 -
Desarrollo de Guía:	- 3 -
Instalación de Asterisk:	- 3 -
Creación de Usuarios:	- 11 -
Zoiper:.....	- 14 -
Buzón de Voz:.....	- 17 -
Configuración del Idioma:.....	- 18 -
Música en Espera:	- 21 -
Transferencias de Llamadas:	- 22 -
Conferencias:	- 23 -
Texto a Voz – TTS:.....	- 25 -
Festival:.....	- 26 -
Menú IVR:	- 29 -
Llamadas en Cola:	- 32 -
Conexión entre PBXs:	- 36 -
Instalación de MariaDB:	- 37 -
Tarificación:	- 45 -
www.marchel.com:.....	- 47 -
Conclusión:	- 51 -
Bibliografía:	- 51 -

Índice de imágenes

Imagen 1: Instalación Asterisk completada	- 4 -
Imagen 2: Configuración de Asterisk	- 4 -
Imagen 3: Menú de configuraciones de Asterisk (1)	- 5 -
Imagen 4: Menú de configuraciones de Asterisk (2)	- 5 -
Imagen 5: Menú de configuraciones de Asterisk (3)	- 6 -
Imagen 6: Menú de configuraciones de Asterisk (4)	- 6 -
Imagen 7: Menú de configuraciones de Asterisk (5)	- 7 -
Imagen 8: Menú de configuraciones de Asterisk (6)	- 7 -
Imagen 9: Menú de configuraciones de Asterisk (7)	- 8 -
Imagen 10: Instalación de configuraciones de Asterisk mediante el Makefile	- 8 -
Imagen 11: Instalación de la configuración de Asterisk completada	- 9 -
Imagen 12: Interfaz de Asterisk en modo verbose.....	- 10 -
Imagen13: Protocolo de transporte udp dentro de pjsip.conf	- 11 -
Imagen 14: Comprobación de los endpoints en Asterisk.....	- 12 -
Imagen 15: Comprobación de los aors en Asterisk	- 13 -
Imagen 16: Formulario de inicio de sesión de Zoiper	- 15 -
Imagen 17: Comprobación y configuración de la IP en Zoiper.....	- 15 -

Imagen 18: Conexión establecida de un usuario final en ZoiPer.....	- 16 -
Imagen 19: Interfaz de llamadas de ZoiPer	- 16 -
Imagen 20: Menú de configuración de usuarios de ZoiPer	- 16 -
Imagen 21: Ejemplo visual de cómo deberían quedar las carpetas de audios del idioma español (es).....	- 20 -
Imagen 22: Diagrama de flujo del funcionamiento de las redirecciones	- 23 -
Imagen 23: Menús de control de usuario del archivo confbridge.conf	- 24 -
Imagen 24: Comprobación de que festival está activo	- 28 -
Imagen 25: Comprobación de que festival se ha configurado correctamente en Asterisk	- 28 -
Imagen 26: Diagrama de flujo sobre el funcionamiento del Menú IVR ¡Error! Marcador no definido.	
Imagen 27: Diagrama de flujo de funcionamiento de la cola de llamadas del soporte técnico ...	- 32 -
Imagen 28: Comprobación de que MariaDB se encuentra activa.....	- 37 -
Imagen 29 Comprobación de creación del usuario asterisk dentro de MariaDB.....	- 39 -
Imagen 30: Comprobación de permisos del usuario asterisk	- 39 -
Imagen 31: Comprobación de la creación de la tabla cdr.....	- 41 -
Imagen 32: Comprobación de que MariaDB se ha conectado correctamente a Asterisk	- 44 -
Imagen 33: Comprobación de que el conector ODBC está bien conectado a Asterisk	- 45 -
Imagen 34: Ejemplo de cómo deben salir los datos que se almacenan en la base de datos tras las llamadas	- 45 -
Imagen 35: Interfaz CRON	- 47 -
Imagen 36: Web Inicio de Sesión.....	- 47 -
Imagen 37: Web Panel Principal.....	- 48 -
Imagen 38: Web Estadísticas	- 49 -
Imagen 39: Web Tarifas.....	- 49 -
Imagen 40: Web Historial.....	- 49 -
Imagen 41: Web Tendencias	- 50 -

Índice de tablas

Tabla 1: Opciones para los canales de voz	- 18 -
Tabla 2: Opciones de las colas de llamadas.....	- 35 -
Tabla 3: Opciones de los CDRs que pueden ser recogidos de Asterisk en MariaDB	- 40 -

Introducción:

Bienvenidos a Marchel, somos una central telefónica creada por estudiantes de la Universidad Politécnica de Alcalá de Henares, basada en Asterisk. Este proyecto se ha realizado como objetivo de una práctica de la asignatura ‘laboratorio de redes, señales y sistemas’. Esta memoria se ha creado a modo de guía explicativa de cada uno de los pasos de creación y configuración de la centralita. Se tratarán temas como la instalación de Asterisk, la configuración de los usuarios, el idioma, la instalación de TTSs como festival, la instalación de MariaDB como base de datos utilizada dentro de la central, algunas funciones básicas como la redirección de llamadas, el buzón de voz, conferencias con amigos (usuarios creados), etc. Finalmente, la mayor parte de las funcionalidades se reúnen en el Menú IVR como la atención al cliente mediante el soporte técnico utilizando colas de llamadas, información sobre la tarificación, entre otros.

Desarrollo de Guía:

Instalación de Asterisk:

Asterisk es un software de código abierto que permite crear sistemas de telefonía IP (VoIP), como centrales telefónicas (PBX), sistemas de respuesta automática (IVR), colas de llamadas, conferencias y más. Fue desarrollado por Digium y es usado en empresas, call centers y operadores de telecomunicaciones. Es decir, Asterisk convierte un servidor común en una central telefónica completa. Por lo tanto, se va a usar para llevar a cabo esta práctica en la que se desarrolla la central telefónica de Marchel.

Para ello procederemos con los siguientes pasos a seguir para la instalación, configuración y explicación de todos los módulos y conceptos necesarios para poder tener tu propia central.

Se comenzará actualizando el sistema

- sudo apt update && sudo apt upgrade

Se deben instalar dependencias necesarias para Asterisk

```
sudo apt install build-essential wget git curl libxml2-dev libncurses5-dev uuid-dev  
libjansson-dev libsdlite3-dev  
sudo apt install libssl-dev libsrtp2-dev
```

Se descarga e instala Asterisk en la carpeta correspondiente, además en esta página puedes encontrar la versión de Asterisk más adecuada para ti, se recomienda para la utilización de pjsip una versión superior a la v20

<https://downloads.asterisk.org/pub/telephony/asterisk/>

```
sudo apt install build-essential wget git curl libxml2-dev libncurses5-dev uuid-dev  
libjansson-dev libsdlite3-dev  
sudo apt install libssl-dev libsrtp2-dev
```

- cd /usr/src/
- sudo wget <https://downloads.asterisk.org/pub/telephony/asterisk/asterisk-22.2.0.tar.gz>
- sudo tar -zvxf asterisk-22.2.0.tar.gz

Se debe situar en la carpeta de Asterisk e instalará las siguientes dependencias adicionales (a partir de aquí hay procesos que tardan bastante, debes ser paciente y esperar a que se vea como en las fotos proporcionadas, además de asegurar con estas que no ha habido problemas)

- cd asterisk-22.2.0/
- sudo contrib/scripts/install_prereq install

```
#####
## install completed successfully
#####
marcos@asterisk:/usr/src/asterisk-22.2.0$ █
```

Imagen 1: Instalación Asterisk completada

A continuación, se procederá con la configuración de Asterisk

- sudo ./configure

```
configure: Menuselect build configuration successfully completed

    .$$$$$$$$$$$$$$$$$=..
    .7$7..          .7$$7:.
    .$$:.           ,$$7
    .7.      7$$$$$     .$$77
    ..$$.      $$$$
    .7$   .?.   $$$$.  .?.
    $.$.   .$$7.  $$$$7 .7$$$. .
    .777.  .$$$$$77$$$$77$$$$$7.   $$$,
    $$$~   .$$$$$77$$$$$7.   .$$$. .
    .$$7   .7$$$$$7:       ?$$$. .
    $$$.   ?7$$$$$7$$.    .$$7
    $$$   .7$$$$$7$$.    :$$. .
    $$$   $$$$.7$$.    .$$$. .
    $$$   $$.7$$.    .$$$. .
    7$$7   7$$.    7$$$. .
    $$$$$.           $$$
    $$$$7.           $$. (TM)
    $$$$$$$.        .7$$$$$$. $$
    $$$$$$7$$.7$$.    $$$$$$$. $$$$$. .
    $$$$$$7$$.7$$.    $$$$$$$. $$$$$. .

configure: Package configured for:
configure: OS type : linux-gnu
configure: Host CPU : x86_64
configure: build-cpu:vendor:os: x86_64 : pc : linux-gnu :
configure: host-cpu:vendor:os: x86_64 : pc : linux-gnu :
marcos@asterisk:/usr/src/asterisk-22.2.0$ █
```

Imagen 2: Configuración de Asterisk

Ahora pondremos que dependencias o aplicaciones queremos que se instalen en Asterisk. Se recomienda que como se van a emplear más adelante algunas cosas como la conversión de texto a voz (TTS /festival) o la base de datos (MariaDB) se revise que estas están activadas para no tener que configurar las de nuevo más adelante. Así pues, al lanzar el siguiente comando fíjate que los módulos en los que ponga **festival**, **mysql** u **odbc** estén activados.

- sudo make menuselect



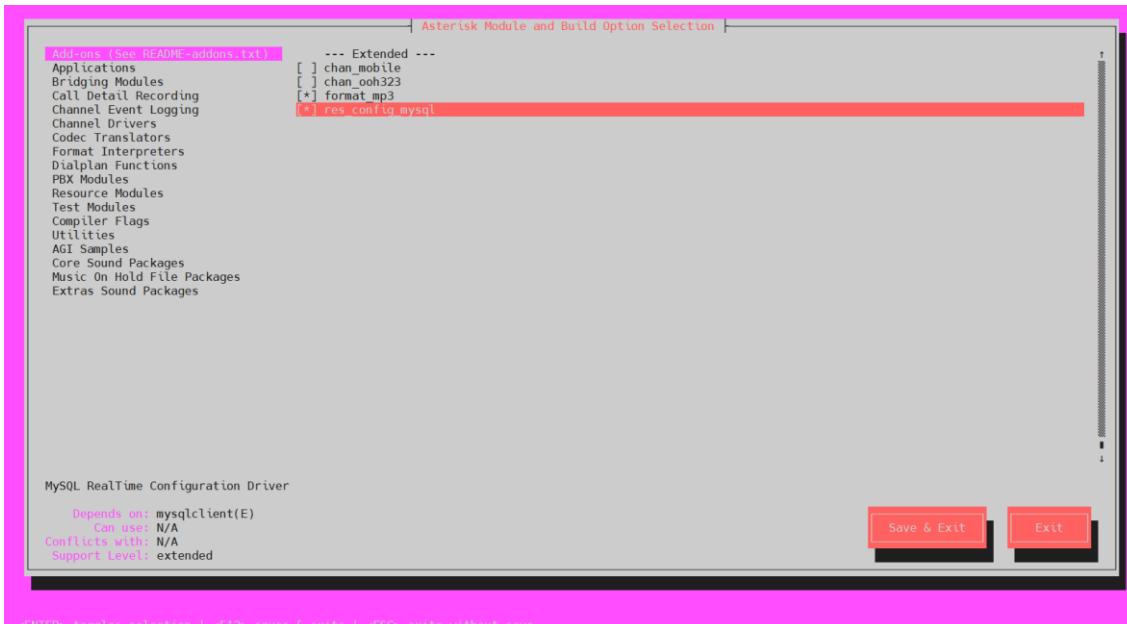


Imagen 3: Menú de configuraciones de Asterisk (1)

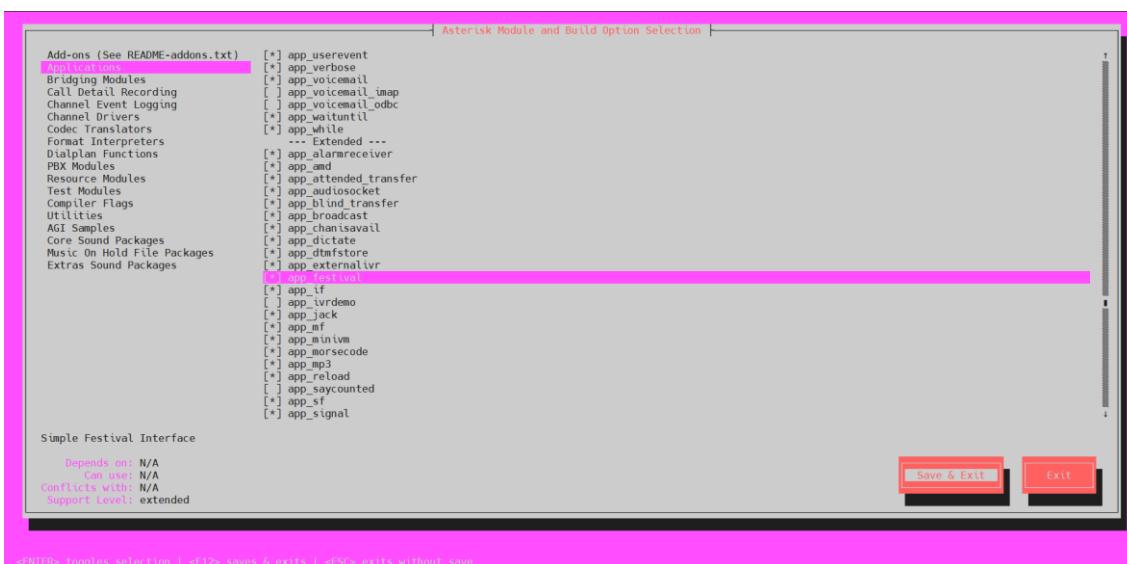


Imagen 4: Menú de configuraciones de Asterisk (2)

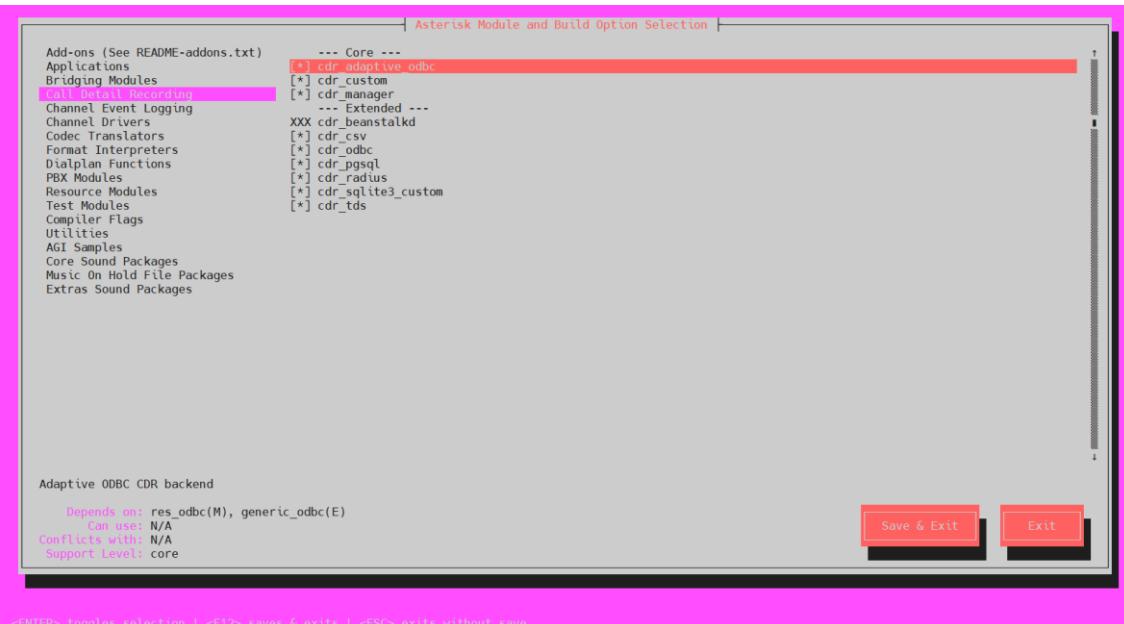


Imagen 5: Menú de configuraciones de Asterisk (3)

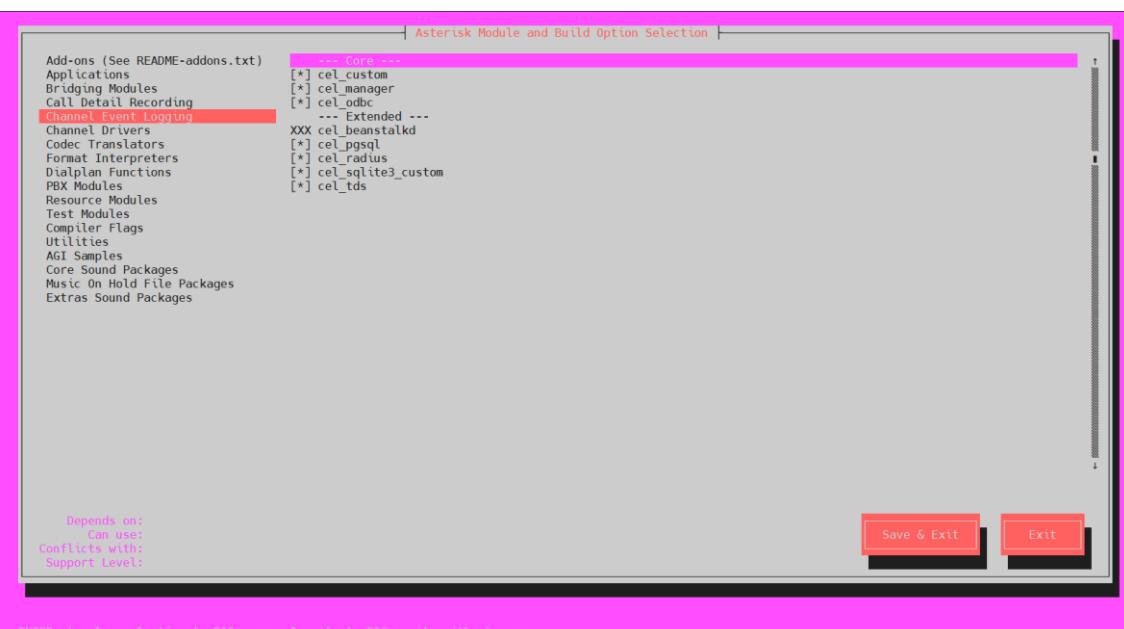
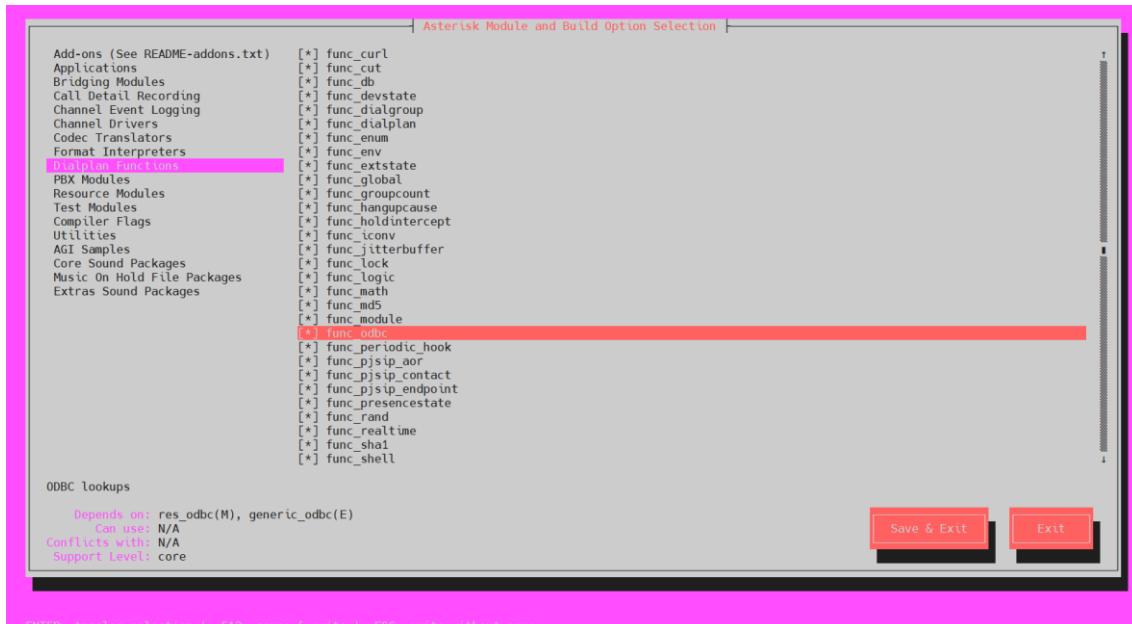
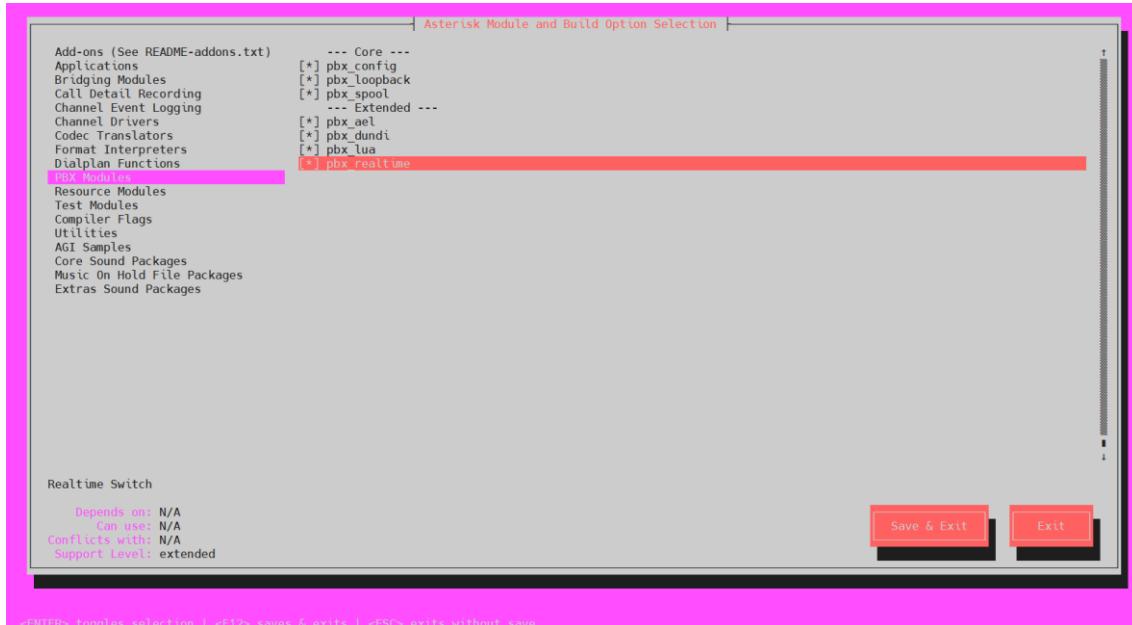


Imagen 6: Menú de configuraciones de Asterisk (4)



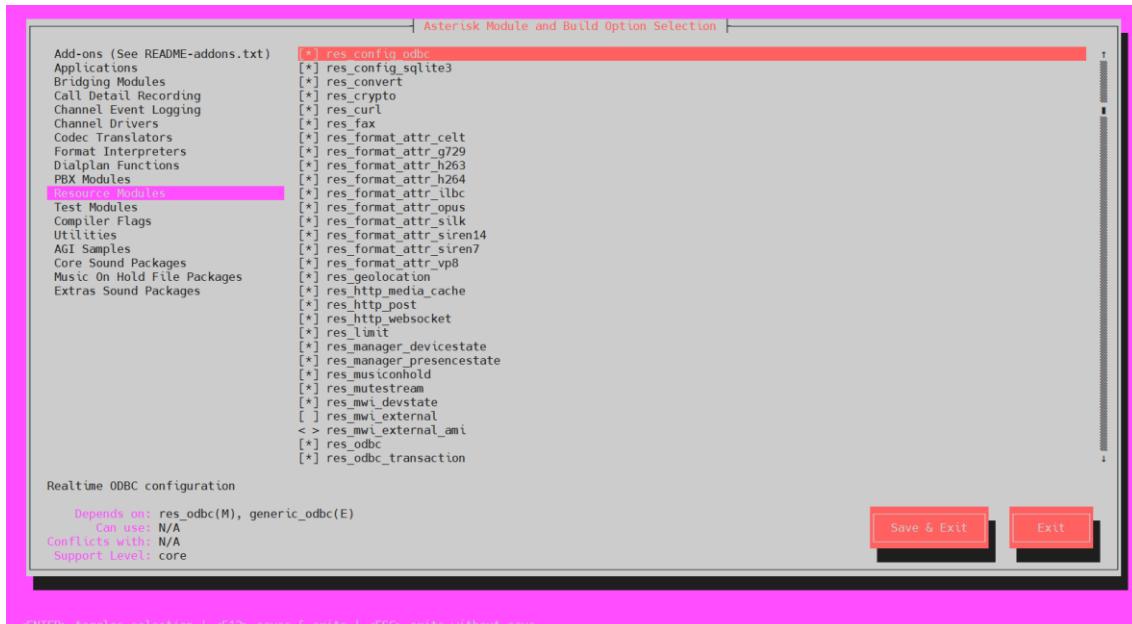
<ENTER> toggles selection | <F12> saves & exits | <ESC> exits without save

Imagen 7: Menú de configuraciones de Asterisk (5)



<ENTER> toggles selection | <F12> saves & exits | <ESC> exits without save

Imagen 8: Menú de configuraciones de Asterisk (6)



<ENTER> toggles selection | <F12> saves & exits | <ESC> exits without save

Imagen 9: Menú de configuraciones de Asterisk (7)

Pulsa **Save & Exit** y termina la instalación haciendo uso del Makefile

- sudo make -j\$(nproc)

Este comando ejecuta el Makefile en función de la cantidad de núcleos de los que dispongas

```
*****
***      ***  

***  --> IMPORTANT INFORMATION ABOUT format_mp3 <---  ***  

***      ***  

*** format_mp3 has been selected to be installed, but the ***  

*** MP3 decoder library has not yet been downloaded into ***  

*** the source tree. To do so, please run the following ***  

*** command: ***  

***      ***  

***      $ contrib/scripts/get_mp3_source.sh ***  

***      ***  

*****  

[CC] res_config_mysql.c -> res_config_mysql.o  

[LD] res config mysql.o -> res config mysql.so  

Building Documentation For: channels pbx apps codecs formats cdr cel bridges funcs tests main res addons  

+----- Asterisk Build Complete -----+  

+ Asterisk has successfully been built, and +  

+ can be installed by running: +  

+          +  

+          make install +  

+-----+  

marcos@asterisk:/usr/src/asterisk-22.2.0$
```

Imagen 10: Instalación de configuraciones de Asterisk mediante el Makefile

- sudo make install

```

+---- Asterisk Installation Complete -----+
+
+ YOU MUST READ THE SECURITY DOCUMENT +
+
+ Asterisk has successfully been installed. +
+ If you would like to install the sample +
+ configuration files (overwriting any +
+ existing config files), run:
+
+ For generic reference documentation:
+   make samples
+
+ For a sample basic PBX:
+   make basic-pbx
+
+
+----- or -----+
+
+ You can go ahead and install the asterisk +
+ program documentation now or later run:
+
+   make progdocs
+
+ **Note** This requires that you have +
+ doxygen installed on your local system
+-----+
marcos@asterisk:/usr/src/asterisk-22.2.0$ █

```

Imagen 11: Instalación de la configuración de Asterisk completada

- sudo make samples
- sudo make config
- sudo ldconfig

Crea un usuario y grupo Asterisk

- sudo adduser --system --group --no-create-home --disabled-login asterisk

Descomenta las líneas **AST_USER="asterisk"** y **AST_GROUP="asterisk"** del siguiente archivo

- sudo vim /etc/default/asterisk

Asegura los permisos del usuario y sobre los archivos de las carpetas a utilizar

- sudo chown -R asterisk:asterisk /etc/asterisk
- sudo chown -R asterisk:asterisk /var/{lib,log,spool}/asterisk
- sudo chown -R asterisk:asterisk /usr/lib/asterisk
- sudo chmod -R 750 /etc/asterisk
- sudo chmod -R 750 /var/{lib,log,spool}/asterisk

Después de esto, recarga y reinicia el servicio



- sudo systemctl daemon-reload
- sudo systemctl restart asterisk

Inicia Asterisk, habilítalo e inícialo

- sudo systemctl start asterisk
- sudo systemctl enable asterisk
- sudo asterisk -rvvvvvv

```
marcos@asterisk:/usr/src/asterisk-22.2.0$ sudo asterisk -rvvvvvv
Asterisk 22.2.0, Copyright (C) 1999 - 2025, Sangoma Technologies Corporation and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 22.2.0 currently running on asterisk (pid = 44519)
asterisk*CLI>
```

Imagen 12: Interfaz de Asterisk en modo verbose

Si te sale esto, ya debería funcionar solo se debe configurar los teléfonos y sus extensiones.

Recuerda que puedes hacer copia de seguridad de los ficheros por lo que pudiese pasar. Se recomienda además de esto tener alguna copia fuera de la máquina virtual, por experiencias presenciadas durante la práctica.

- cd /etc/asterisk
- sudo cp fichero.conf fichero.conf.bck

Para proceder a configurar los dispositivos finales y realizar llamadas, te invito a seguir esta guía. En la que se abordarán bastantes temas relacionados con Asterisk que te podrían interesar.

Con el siguiente comando se pueden consultar todos los archivos que se pueden modificar para realizar las distintas configuraciones de Asterisk. Cada uno de estos archivos contiene ejemplos comentados que pueden ser de ayuda.

- sudo vim /etc/asterisk/

Para un archivo específico solo tienes que agregar su nombre. Ejemplo:

- sudo vim /etc/asterisk/pjsip.conf

Siempre que se **modifique** alguno de estos archivos es conveniente **recargar** las configuraciones desde la terminal de Asterisk (CLI):

- reload -> de forma **global** (recarga todo)

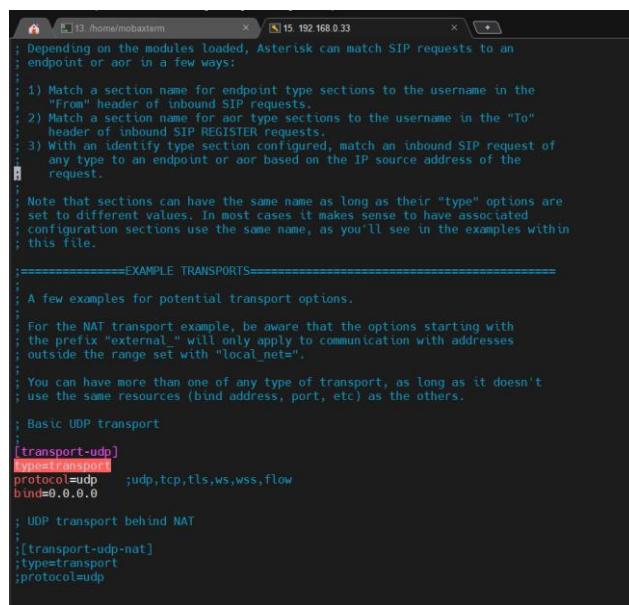
- diaplan reload -> de forma **específica** (recarga únicamente extension.conf)

Creación de Usuarios:

Una vez que se haya instalado Asterisk correctamente lo primero es crear un primer usuario, para ello, es necesario modificar el archivo **pjsip.conf**. en este archivo es donde se configuraran todos los usuarios y números endpoints que sean necesarios. Para abrir el archivo **pjsip.conf**:

- sudo vim /etc/asterisk/pjsip.conf

Una vez ejecutado el comando aparecerá lo siguiente:



```
Depending on the modules loaded, Asterisk can match SIP requests to an endpoint or aor in a few ways:
1) Match a section name for endpoint type sections to the username in the "From" header of inbound SIP requests.
2) Match a section name for aor type sections to the username in the "To" header of inbound SIP REGISTER requests.
3) With an identify type section configured, match an inbound SIP request of any type to an endpoint or aor based on the IP source address of the request.

Note that sections can have the same name as long as their "type" options are set to different values. In most cases it makes sense to have associated configuration sections use the same name, as you'll see in the examples within this file.

=====EXAMPLE TRANSPORTS=====
A few examples for potential transport options.

For the NAT transport example, be aware that the options starting with the prefix "external_" will only apply to communication with addresses outside the range set with "local_net=".

You can have more than one of any type of transport, as long as it doesn't use the same resources (bind address, port, etc) as the others.

: Basic UDP transport
[transport-udp]
;type=transport
;protocol=udp ;udp,tcp,tls,ws,wss,flow
;bind=0.0.0.0

: UDP transport behind NAT
[transport-udp-nat]
;type=transport
;protocol=udp
```

Imagen 13: Protocolo de transporte udp dentro de pjsip.conf

Todos los comentarios que aparecen en azul (comenzados por un ;) son ejemplos que ofrece el propio archivo (recomendamos leerlos). En primer lugar, se des comentarán o se agregarán las tres líneas que se pueden observar en la imagen. Con estas 3 líneas le estamos diciendo que el protocolo que utilizarán las llamadas IP será **UDP** ya que, al ser una llamada, no interesa la retransmisión de paquetes sino la velocidad con la que se transmiten.

- [transport-udp]
type=transport
protocol=udp
bind=0.0.0.0

Una vez configurado el protocolo de transporte, se procede a la creación de usuarios en este mismo archivo.

- [2001-softphone]
 type=endpoint
 context=office-phone -> contexto que usaremos en el **extensions.conf**
 disallow=all
 allow=ulaw
 auth=2001-auth
 aors=2001-softphone
 language=es

- [2001-auth] -> debe coincidir con el auth =.. del usuario
 type=auth
 auth_type=userpass
 username=2001-softphone -> usuario que usaremos en **zoiper**
 password=Hola123

- [2001-softphone] -> debe coincidir con el aor =.. del usuario
 type=aor
 max_contacts=1

Con estas líneas se ha configurado un usuario llamado 2001-softphone de tipo endpoints, es decir, este será uno de nuestros usuarios finales. En Marchel se crearán como usuarios finales los **2XXX**, que se irán creando en base al código anterior. Es importante destacar que estos usuarios se crearan con el contexto **office-phone**. En caso de querer comprobar que la configuración de usuarios finales se ha realizado adecuadamente, introduzca el siguiente comando en la terminal de Asterisk:

- *CLI> pjsip show endpoints

```

Endpoint: 2001-softphone                               Not in use   0 of inf
  InAuth: 2001-auth/2001-softphone
  Aor: 2001-softphone                                1
  Contact: 2001-softphone/sip:2001-softphone@192.168. 5758366fb3 NonQual      nan

Endpoint: 2002-softphone                               Not in use   0 of inf
  InAuth: 2002-auth/2002-softphone
  Aor: 2002-softphone                                1
  Contact: 2002-softphone/sip:2002-softphone@192.168. d893419951 NonQual      nan

Endpoint: 2003-softphone                               Not in use   0 of inf
  InAuth: 2003-auth/2003-softphone
  Aor: 2003-softphone                                1
  Contact: 2003-softphone/sip:2003-softphone@192.168. 18d2d2449d NonQual      nan

```

Imagen 14: Comprobación de los endpoints en Asterisk

Por otra parte, también es importante comprobar la configuración **aor** (Address of Record) ya que sin esta Asterisk no sabría a dónde enviar las llamadas que se realicen para este usuario, ya que **aor** define las direcciones registradas de un **endpoint**.

- *CLI> pjsip show aors

Aor: 2001-softphone	1	
Contact: 2001-softphone/sip:2001-softphone@192.168.0. 5758366fb3 NonQual		nan
Aor: 2002-softphone	1	
Contact: 2002-softphone/sip:2002-softphone@192.168.0. d893419951 NonQual		nan
Aor: 2003-softphone	1	
Contact: 2003-softphone/sip:2003-softphone@192.168.0. 18d2d2449d NonQual		nan

Imagen 15: Comprobación de los aors en Asterisk

Una vez que se tengan mínimo dos usuarios **configurados correctamente** es momento de pasar al archivo **extensions.conf**. Este archivo junto al **pjsip.conf** serán los más importantes de la práctica, ya que extensions.conf es donde se procesa la configuración de todas las llamadas dentro de la PBX. Se podría decir que este es el "cerebro" de la centralita, puesto que dice qué hacer cuando alguien marca un número.

- [office-phone]


```
exten => 2001,1,Answer() ; Responde la llamada
exten => 2001,n,Dial(PJSIP/2001-softphone,20,m) ; Llama a la extensión
exten => 2001,n,Hangup(); Cuelga la llamada
```

```
exten => 2002,1,Answer()
exten => 2002,n,Dial(PJSIP/2002-softphone,20,m)
exten => 2002,n,Hangup()
```

```
exten => 2003,1,Answer()
exten => 2003,n,Dial(PJSIP/2003-softphone,20,m)
exten => 2003,n,Hangup()
```

A la hora de realizar la configuración es muy importante tener en cuenta el contexto que se ha establecido en el **pjsip.conf**, ya que la configuración de estas llamadas se va a procesar dentro de este contexto. En el caso de Marchel, todos los usuarios de la central del tipo 2XXX se han configurado con el contexto **office-phone**.

Otro aspecto importante a tener en cuenta es el **orden de prioridades**, pues define en que orden se ejecutarán los procesos. Una prioridad no puede duplicarse dentro de una misma línea de acciones. En el código de ejemplo, se ha utilizado la prioridad 1 para responder la llamada **Answer()**. Esta aplicación no es estrictamente necesaria para que se realice correctamente la llamada, ya que solamente cumple la función de aceptar la conexión con un tercero, pero ayuda a la lógica de comprensión y a futuras modificaciones. Por ejemplo, si se quieren realizar redirecciones o añadir lógicas TTS antes de establecer la llamada en sí. En el resto de las aplicaciones, se ha utilizado la prioridad n que se define como la prioridad siguiente, también se podría haber continuado con la lógica de prioridad (1, 2, 3... etc.)

En el código proporcionado la aplicación que se ejecuta como segunda prioridad es **Dial()**, que se encarga de conectar la llamada con otro destino siguiendo la siguiente sintaxis **Dial(tech/resource, tiempo, opciones)**. En este caso el recurso es **PJSIP**, espera 20 segundos a ser contestado y con la opción **m** activa la llamada en espera. La última

aplicación según el orden de prioridades es **Hangup()** que se encarga de finalizar la llamada.

Como conceptos extra a tener en cuenta y que ayudan a simplificar el código de este archivo se tiene, en primer lugar, el uso de **same**, que una vez ya está definida la extensión se puede emplear para no tener que poner el número de extensión constantemente. En segundo lugar, se pueden emplear también variables que se definen si son locales dentro de un contexto con **Set()** y si son globales en la sección **[globals]**, para emplearlas se sigue la nomenclatura **\${VARIABLE}**. También se pueden utilizar variables que vienen predefinidas por Asterisk como veremos más adelante. Estos conceptos se pueden usar de la siguiente forma:

- [globals]
TIEMPO_ESPERA=20
- [office-phone]
exten => 2001,1,Answer()
same => n,Dial(PJSIP/2002, \${TIEMPO_ESPERA},m)
same => n,Hangup()

Con estas configuraciones básicas en extensions.conf y en pjsip.con estas preparado para registrar tus usuarios en Zoiper y realizar las primeras llamadas.

Zoiper:

Una vez instalado Asterisk en tu máquina virtual y configurados los distintos usuarios, será necesario descargar una aplicación que permita la realización de llamadas telefónica mediante IP. Existen varias apps gratuitas como Pangolin, Linphone, Jitsi, **Zoiper**, entre otras. Estas aplicaciones hacen uso de la tecnología VoIP ya que se conectan a servidores SIP a través de Internet o redes IP por lo cual son ideales para usar Asterisk.

Dada su simplicidad y compatibilidad con esta práctica se utilizará **Zoiper**. Esta app está disponible de forma gratuita tanto en la AppStore para Android o IOS como para el ordenador. <https://www.zoiper.com/en/voip-softphone/download/current>

Zoiper va a ser la aplicación que nos sirva de interfaz para poder comunicar a los usuarios de la red. Una vez que se esté en Zoiper se continuará con la opción gratuita y nos aparecerá como primera pantalla, algo muy similar a la imagen 16, aquí se debe introducir:

- El **nombre del usuario** que se ha configurado anteriormente y se quiere utilizar en el dispositivo. Ejemplo: 2001-softphone.
- La **contraseña** que se ha configurado para dicho usuario. Ejemplo: Hola123

En cada dispositivo Zoiper solo se puede tener un usuario activo. Es decir, para utilizar dos usuarios se necesita mínimo el ordenador y un móvil.

En la imagen 17 aparece la siguiente pantalla en la cual se debe introducir la **IP** del ordenador se está corriendo Asterisk.

1. Introducir en la máquina virtual el comando ***ip a***
2. Introducir la **IP** en Zoiper (si estas en modo bridge cada vez que cambies de subred cambiara también la IP de la máquina virtual)



Imagen 16:
Formulario de inicio
de sesión de ZoiPer

```
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of jue 18 mar 2025 10:55:10 UTC
System load: 0.07 Processes: 145
Usage of /: 79.5% of 11.21GB Users logged in: 0
Memory usage: 16% IPv4 address for enp0s3: 192.168.0.33
Swap usage: 0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado
Se puede aplicar 1 actualización de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable
Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

chelseafh@lrss:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host noprefixroute
                valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0c:e9:c5 brd ff:ff:ff:ff:ff:ff
        inet 192.168.0.33/24 metric 100 brd 192.168.0.255 scope global dynamic enp0s3
            valid_lft 8621sec preferred_lft 8621sec
            inet6 fe80::aa0:27ff:fe0c:e9c5/64 scope link
                valid_lft forever preferred_lft forever
chelseafh@lrss:~$ chelseafh@lrss:~$ chelseafh@lrss:~$ _
```

Imagen 17: Comprobación y configuración de la IP en ZoiPer

¡¡¡Si aparece esta imagen significa que todo ha ido **BIEN!!!**

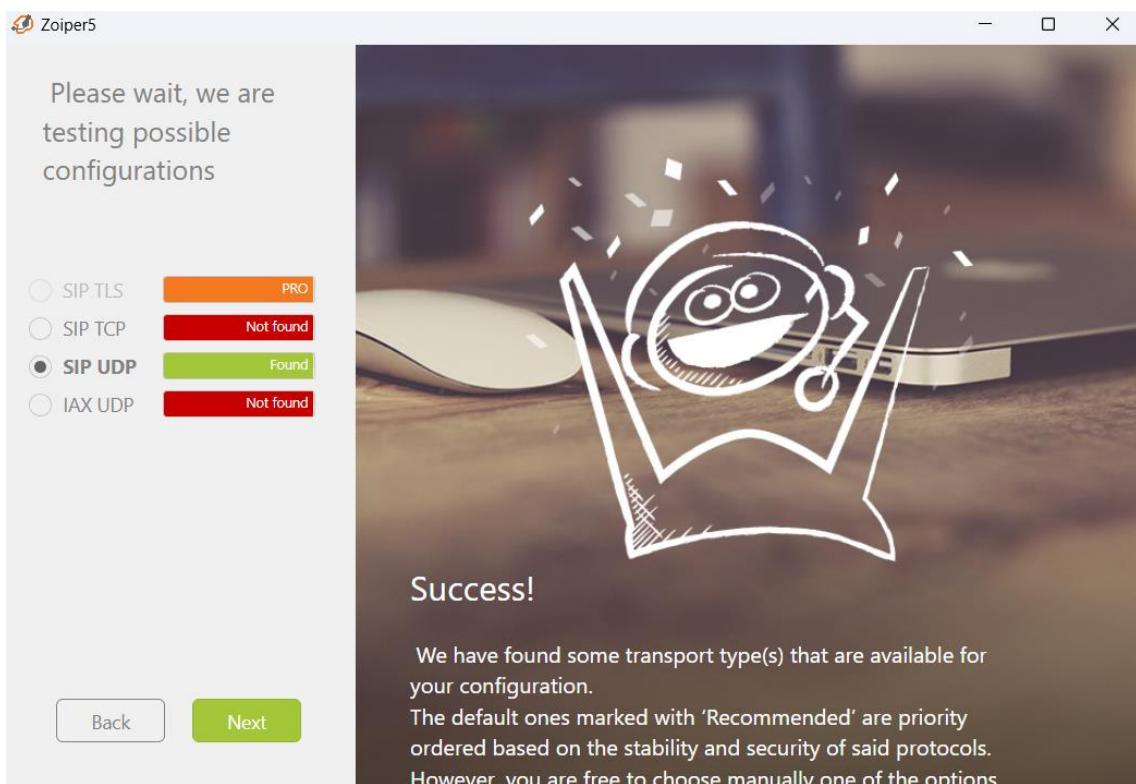


Imagen 18: Conexión establecida de un usuario final en ZoiPer

Una vez que configurado el usuario aparecerá una imagen similar a la de abajo a la izquierda y estarás preparado para realizar tus **primeras llamadas**. En caso de que necesites hacer alguna configuración porque hayas cambiado de IP o quieras cambiar de usuario lo puedes hacer desde **ajustes -> accounts** (imagen abajo a la derecha)

The screenshot shows the Zoiper5 software interface. On the left, there is a sidebar with "Contacts" and "Recent" sections. In the center, there is a "Accounts" section for "SIP". One account is listed: "2002-softphone" (status: 2 errors). The account details are shown in a table:

Domain	92.168.190.180
Username	2002-softphone
Password	*****

Below the account details, there are sections for "Optional SIP credentials" with checkboxes for "Use auth. username" and "Use outbound proxy". There is also a "Outbound proxy" input field.

Imagen 19: Interfaz de llamadas de ZoiPer

Imagen 20: Menú de configuración de usuarios de ZoiPer

Buzón de Voz:

Para configurar el buzón de voz el archivo que se debe modificar principalmente es el **voicemail.conf**. Se sigue la siguiente estructura **Extensión => PIN, Nombre de usuario, Correo electrónico, [opciones adicionales]**.

- [default]
2001 => 1234,2001-softphone
2002 => 1234,2002-softphone
2003 => 1234,2003-softphone

En este caso, con estas líneas se configurarán 3 extensiones 2XXX dentro del contexto default, con una misma contraseña (por comodidad) y no se ha configurado el correo electrónico ya que no es obligatorio. En cuanto a las opciones adicionales se tiene algunas como:

- attach=yes para adjuntar el mensaje en el correo
- maxmessage para definir la duración máxima del mensaje
- serveremail, fromstring, y emailbody para configurar el envío de notificaciones por correo

A continuación, se va a configurar en el **extensions.conf** tanto la opción de buzón de voz en los usuarios como el propio número del buzón de voz, esta parte es **importante** para el funcionamiento del buzón.

En primer lugar, se debe añadir una línea a la configuración de cada llamada

- exten => 2001,1,Answer() ; Responde la llamada
exten => 2001,n,Dial(PJSIP/2001-softphone,20,m) ; Llama a la extensión
exten => 2001,n,VoiceMail(2001,u) -> Nueva Línea
exten => 2001,n,Hangup(); Cuelga la llamada

Esta línea se debe agregar en todos los usuarios en los que se quiera configurar el buzón de voz, en el caso de Marchel se ha configurado el buzón de voz en todos los usuarios 200X. Recuerde que al hacer las configuraciones del **Dial()** se agregó como tiempo de espera a ser contestado 20 seg, por tanto, el buzón de voz saltara a los 20 seg en caso de que el usuario llamado no conteste. Además, se ha agregado la opción **u** del buzón de voz que permite al usuario grabar un mensaje personalizado. Existen otras opciones como:

Opción	Descripción
b	Reproduce el saludo antes de permitir que el llamante deje un mensaje.
s	Silencioso: No reproduce el saludo del buzón de voz, pero permite dejar un mensaje.
u	Saludo de usuario: Si el usuario tiene un saludo personalizado, lo usa en lugar del mensaje predeterminado.
p	Pausa antes de grabar: Introduce una pausa antes de empezar a grabar el mensaje.
g	Reproduce el nombre grabado (si está disponible) antes del saludo.

d([c])	Modo Directorio: En lugar de grabar un mensaje, envía el llamante al directorio de extensiones de la centralita. Si se usa d(c), también permite buscar por el apellido.
f	Forzar inicio de grabación sin esperar a un tono de voz.
k	Mantiene la llamada abierta después de grabar el mensaje en lugar de colgar.
t	Permite transferir la llamada a otra extensión en lugar de dejar un mensaje.
T	Ofrece la opción de transferencia , pero requiere confirmación del usuario antes de ejecutar la acción.
a(x)	Anuncia el mensaje como urgente en la bandeja de entrada.
n	No reproduce el mensaje "Deje su mensaje después del tono".

Tabla 1: Opciones para los canales de voz

Por último, la extensión 2000 será el número en el que se almacenarán los mensajes recibidos, al llamar pedirá tu número y contraseña configurada anteriormente (1234) para poder escuchar los mensajes. Por otra parte, en caso de haber elegido la opción **u**, dentro del menú de configuraciones podrás **personalizar** tu mensaje de voz.

- exten => 2000,1,VoiceMailMain(\${CALLERID(num)})
exten => 2000,n,Hangup()

Configuración del Idioma:

Una vez que realices las primeras llamadas, al saltar el buzón de voz, podrás notar que el idioma no está en español o es un español no muy fluido.

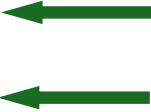
El primer paso será crear un directorio en la carpeta sounds. Esta la puedes llamar de cualquier forma, pero por comprensión, se recomienda llamarla en función al idioma a configurar, en este caso **es**. Para ello utilizaremos el siguiente comando:

- sudo mkdir /var/lib/asterisk/sounds/es

Una vez creada la carpeta, el siguiente paso a seguir sería descargar el idioma, <https://www.sinologic.net/proyectos/vocesbak/> en este enlace encontraras algunas opciones para descargar. En esta práctica se va a utilizar la extensión de GSM y se descargaran los dos paquetes correspondientes para que estén todos los archivos posibles y no haya errores.

Voces de Asterisk

licenciadeuso.txt
voipnovatos-extra-sounds-es-alaw-1.4.tar.gz
voipnovatos-extra-sounds-es-g729-1.4.tar.gz
asterisk-voces-es-v1_2-moh-voipnovatos.tar.gz
voipnovatos-extra-sounds-es-ulaw-1.4.tar.gz
voipnovatos-extra-sounds-es-gsm-1.4.tar.gz
voipnovatos-core-sounds-es-g729-1.4.tar.gz
voipnovatos-core-sounds-es-alaw-1.4.tar.gz
voipnovatos-core-sounds-es-gsm-1.4.tar.gz
voipnovatos-core-sounds-es-ulaw-1.4.tar.gz



Mirror de la web <http://www.voipnovatos.es/voces/> para todos los que lo necesiten y no esté disponible en la web original.
Las voces de las versiones de Asterisk 1.2 las puede encontrar [aqui](#).

Comandos útiles:

- `sudo chmod -R 777 /var/lib/asterisk/sounds`
`cd /var/lib/asterisk/sounds/es` -> acceder a la carpeta

Este comando se emplea para conceder todos los permisos a cualquier usuario sobre un directorio y todos subdirectorios dentro de él. Se emplea el comando con la siguiente sintaxis **sudo chmod -R permisos ruta/directorio**

- `sudo wget https://www.sinologic.net/proyectos/vocesbak/voipnovatos-core-sounds-es-gsm-1.4.tar.gz`

Este comando se usa para descargar archivos comprimidos, por lo tanto, la ubicación es algo importante a tener en cuenta. Si no se especifica la ubicación, se descargará automáticamente en el directorio de trabajo en el que te encuentres y esto podría suponer un problema a la hora de descomprimir los archivos. Se usa siguiendo el siguiente formato **sudo wget enlace/a/descargar ubicación/de/descarga**

- `sudo tar -xvf voipnovatos-core-sounds-es-gsm-1.4.tar.gz`

Este comando sirve para descomprimir archivos **.tar** en caso de ser **.zip** o similar se emplearía un comando distinto como puede ser **unzip**. Este comando sigue la siguiente estructura **sudo tar -xvf ubicación/archivo (-C ruta/destino)**. Esta última parte sólo es necesaria si se quiere descomprimir el archivo en un directorio en específico.

De forma análoga se emplearán los siguientes comandos para el otro archivo comprimido:

- `sudo wget https://www.sinologic.net/proyectos/vocesbak/voipnovatos-extra-sounds-es-gsm-1.4.tar.gz`
- `sudo tar -xvf voipnovatos-extra-sounds-es-gsm-1.4.tar.gz`

Una vez se han descargado y descomprimido los archivos se puede observar que estos se descargan dentro de una subcarpeta llamada **es** dentro de las carpetas dictate, digits, es,

followme, letters, phonetic y silence. Para que Asterisk encuentre los archivos y funcione la configuración del idioma de manera correcta. Se debe extraer el contenido de dichas subcarpetas a las carpetas origen (dictate, digits, es, followme, letters, phonetic y silence). Para reorganizar los archivos se empleará un comando con la siguiente estructura **sudo mv ubicación/archivo ruta/destino**. Cabe destacar que es importante hacerlo para cada carpeta.

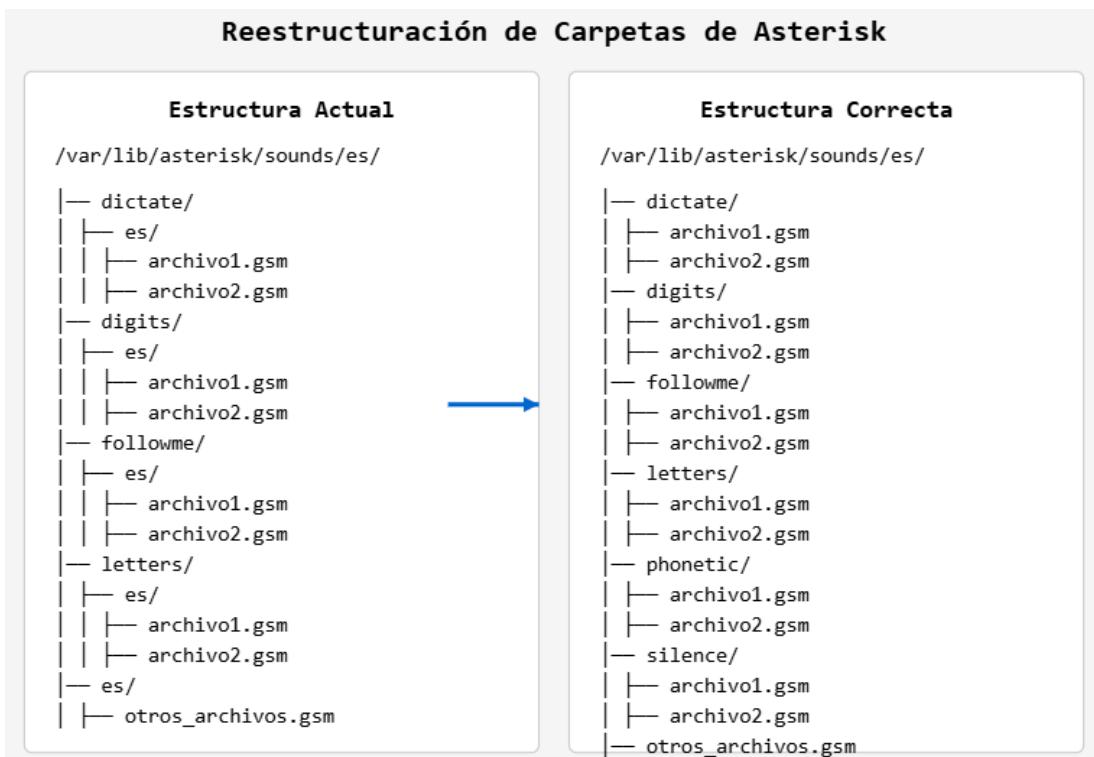


Imagen 21: Ejemplo visual de cómo deberían quedar las carpetas de audios del idioma español (es)

Algún ejemplo sería:

- sudo mv ./dictate/es/* ./dictate/
- sudo mv ./digits/es/* ./digits/
- sudo mv ./es .
- sudo mv ./followme/es/* ./followme/

Por último, se empleará otra vez el comando para dar permisos a los audios para asegurarnos de que funcione todo correctamente

- sudo chmod -R 777 /var/lib/asterisk/sounds

Una vez que este todo descargado y en la carpeta correcta, es momento de añadir dos líneas en el pjsip.conf

- [general]
language=es -> nombre de la carpeta

En caso de que descargues varios idiomas y tengas varias carpetas, así como: *es*, *en*, *fr*, *es1*, *es2*, etc. El parámetro *language* se debe igualar al nombre de la carpeta donde se tiene el idioma que se quiere utilizar.

Al colocar *language* dentro del contexto general, todos los usuarios configurados usaran el idioma español, en caso de querer utilizar distintos idiomas en distintos usuarios será necesario añadir la línea en cada usuario.

Ejemplo:

- [2001-softphone]
type=endpoint
context=office-phone
.....
language=en -> nueva linea

Música en Espera:

Para añadir música en espera a las llamadas, se deberá, primeramente, crear un directorio en la carpeta ***moh*** o en la carpeta ***sounds***. En este caso se ha utilizado la carpeta ***sounds***

- sudo mkdir /var/lib/asterisk/sounds/custom

En esta carpeta es donde se almacenarán los sonidos que deseas que suenen como música de espera. Para ello será necesario descargar el archivo de audio que se quiera, a poder ser en extensión ***.wav*** y pasar este archivo al servidor mediante una conexión ***ssh***. Para asegurar que el formato y la frecuencia son correctos, se empleara el siguiente comando:

- sox dirección-archivo/nombre-archivo.wav -c 1 -r 8000 -t wav
/var/lib/asterisk/sounds/custom/nombre-deseado.wav

Una vez hecho esto se deberá modificar el archivo ***musiconhold.conf***. Por tanto, nos dirigiremos a este con un editor de texto y modificaremos las siguientes líneas dentro del contexto ***default***:

- [default]
mode=files
directory=/var/lib/asterisk/sounds/custom/

En ***directory*** debe aparecer la ruta exacta a la carpeta donde se hallan los archivos de sonido que se quieren utilizar, ***mode = files*** significa que la música en espera se reproducirá a partir de archivos de audio almacenados en el servidor. En Marchel se ha utilizado el contexto ***default***, pero en caso de querer canciones de espera personalizadas para cada ocasión se puedes crear dentro de otro contexto

Ejemplo:

- [jazz]

mode=files

directory=/var/lib/asterisk/sounds/custom/jazz

Otra forma de utilizar diferentes canciones, pero de forma indiferente es añadir más de una canción a la carpeta y añadir la línea **random** para que estas suenen de forma aleatoria cada vez que llames.

- random=yes

Por último, se debe añadir una línea al diaplan para que funciones correctamente la música de espera, esta debe tener la mínima prioridad para que se ejecuten el resto de las funciones sin problemas:

- exten => 2002,1,Answer() ; Responde la llamada

exten => 2002,n,Dial(PJSIP/2002-softphone,20,m) ; Llama a la extensión

exten => 2002,n,VoiceMail(2002,b) ; Si no se contesta, va al buzón de voz

exten => 2002,n,Hangup() ; Cuelga la llamada

exten => 2002,n,MusicOnHold() -> linea nueva

En caso de haber añadido otro contexto la línea cambiaria a exten => 2002, n, MusicOnHold(jazz)

Transferencias de Llamadas:

En este apartado se van a realizar configuraciones en el archivo **features.conf**, en Asterisk se usa para definir funcionalidades avanzadas de llamadas, como **transferencias, estacionamiento y grabación en vivo**. La mayoría de estas configuraciones se realizan dentro del contexto **[featuremap]**.

Una de las funciones que se pueden configurar es activar la Grabación Automática de Llamadas. Esta opción permite que los usuarios inicien una grabación de la llamada de forma manual. El usuario debe pulsar *3 y la grabación comenzara. El audio producto de la grabación se archivará en /var/spool/asterisk/monitor/. Es importante configurar la opción W o w en el Dial() o en Queues() en el archivo extensions.conf.

- [featuremap] -> en el features.conf

automixmon => *3
- exten => 2001,1,Dial(SIP/2001,30,w) -> en el extension.conf

Otra función interesante es las Transferencias de llamadas. Que se pueden hacer en dos modos en modo ciego y en modo asistido doble. Ambas funciones se activan cuando el usuario presiona el comando predefinido en features.conf. La diferencia se haya en su funcionabilidad, en el modo ciego una vez que el usuario 1 marca el comando y el número hacia donde desea realizar la transferencia (usuario 3), el usuario 1 es expulsado de la llamada y quedan conectados el usuario 2 y el 3. En el modo asistido el usuario 1 y 2 se quedan conectados y son capaces de hablar entre ellos, pero el 1 y el 3 también se

conectan una vez que el usuario 1 cuelga la transferencia se hace efectiva entre el usuario 2 y 3.

- [featuremap] -> en el features.conf
 blindxfer => ## ; Transferencia ciega con doble #
 atxfer => ** ; Transferencia asistida con doble *
 disconnect => *0 ; Cuelga la llamada al presionar *0
- exten => 2001,1,Dial(SIP/2001,30,Tt) -> en el extension.conf

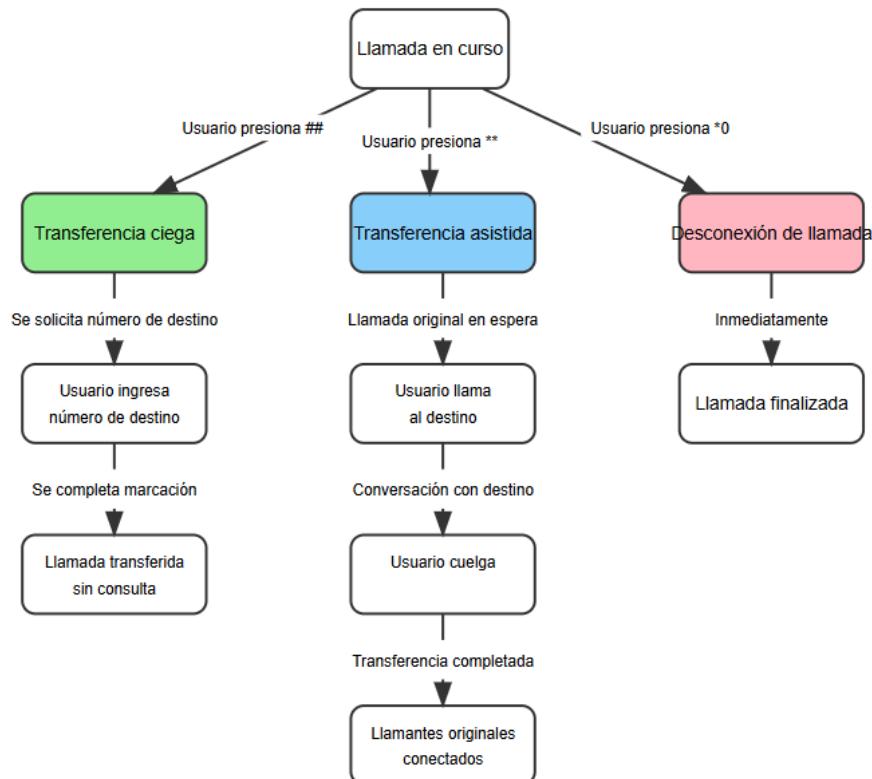


Imagen 22: Diagrama de flujo del funcionamiento de las redirecciones

En esta práctica se van a utilizar ambas funciones para el soporte técnico para ayudar a la lógica a de funcionamiento y aportar intercomunicación entre los agentes del soporte técnico para ofrecer así un mejor servicio. La aplicación de estas funcionalidades en el soporte técnico se detallará más adelante en los apartados de Colas y Menú IVR.

Conferencias:

Se quiere crear una sala de conferencias del estilo de las salas de la plataforma Discord donde se puedan unir usuarios a una llamada o dejar esta cuando quieran. Para ello, se tendrá que modificar el archivo **confbridge.conf**. En este archivo vendrán ya algunas líneas de código configuradas por defecto. En primer lugar, podemos observar al final del

archivo los menús de control de usuario según su nivel de control sobre la conferencia, los cuales se pueden modificar si se desea, pero no sería necesario. Estos son útiles sobre todo en contexto de administrador, ya que puedes bloquear la llamada o expulsar a gente de la misma pulsando las distintas extensiones, estas funciones se observan de forma más clara en la siguiente imagen:

```
[sample_user_menu]
type=menu
*=playback_and_continue(conf-usermenu)
*1=toggle_mute
1=toggle_mute
*4=decrease_listening_volume
4=decrease_listening_volume
*6=increase_listening_volume
6=increase_listening_volume
*7=decrease_talking_volume
7=decrease_talking_volume
*8=leave_conference
8=leave_conference
*9=increase_talking_volume
9=increase_talking_volume

[sample_admin_menu]
type=menu
*=playback_and_continue(conf-adminmenu)
*1=toggle_mute
1=toggle_mute
*2=admin_toggle_conference_lock ; only applied to admin users
2=admin_toggle_conference_lock ; only applied to admin users
*3=admin_kick_last ; only applied to admin users
3=admin_kick_last ; only applied to admin users
*4=decrease_listening_volume
4=decrease_listening_volume
*6=increase_listening_volume
6=increase_listening_volume
*7=decrease_talking_volume
7=decrease_talking_volume
*8=no_op
8=no_op
*9=increase_talking_volume
9=increase_talking_volume
```

Imagen 23: Menús de control de usuario del archivo confbridge.conf

En segundo lugar, se deben modificar las siguientes líneas de código o añadirlas en caso de que no estén. Se deben tener los contextos user, admin y bridge. Tanto para user como para admin, ambos son **type=user** y se distinguen en usuarios y administradores con **admin=no/yes**. También es importante añadir el **pin=contraseña**, donde *contraseña* será lo que debe ingresar cada persona al entrar a la conferencia, con esto puedes discernir a los usuarios habituales de los administradores, ya que estos disponen de mayores permisos sobre la conferencia.

Además, como se quiere que sea similar a una sala de Discord, para se pueda iniciar la conferencia y que esta no finalice cuando el moderador, es decir el número marcado, no

esté conectado (o se desconecte) es necesario añadir **wait_marked=no** y **end_marked=no**. Asimismo, se establece a los administradores como moderadores con **marked=yes**. El resto de las funcionalidades son opcionales y se pueden añadir si se desean, de no ser así funcionaría de forma correcta de todos modos.

Para bridge solo se deberá definir que es **type=bridge**. El resto es opcional, pero es una buena práctica como mínimo añadir el idioma y un límite de participantes en la conferencia para que esta no se sature.

- [default_user]
type=user
admin=no
music_on_hold_when_empty=yes
quiet=no
announce_user_count=yes
announce_join_leave=yes
wait_marked=no
end_marked=no
dsp_drop_silence=no
pin=1234

- [default_admin]
type=user
marked=yes
admin=yes
pin=9876

- [default_bridge]
type=bridge
language=es1
max_members=10

Por último, para que funcione todo de manera correcta se debe añadir en el archivo **extensions.conf** una extensión que sirve de puente entre los usuarios. Esto define una sala de conferencia con el id 1 y los permisos de los usuarios en función de su contexto siguiendo la siguiente estructura: **ConfBridge (IDconferencia, contexto_bridge, contexto_usuario/admin)**.

- exten => 9000,1,Answer()
 same => n,ConfBridge(1,default_bridge,default_user)
 same => n,Hangup()

 exten => 9001,1,Answer()
 same => n,ConfBridge(1234,default_bridge,default_admin)
 same => n,Hangup()

Texto a Voz – TTS:

Para conseguir este servicio de TTS (Text To Speech) se pueden utilizar distintos sistemas diferentes como Festival, eSpeak, Coqui TTS entre otras. También se pueden utilizar APIs

de pago como podrían ser los servicios de Google Cloud o Amazon. En este caso se va a usar Festival ya que es el más sencillo y compatible con Asterisk. Así que con esto en claro, se va a proceder a la instalación y configuración del mismo.

Festival:

Instala Festival

```
sudo apt-get install festival
```

Detén Asterisk

```
sudo /etc/init.d/asterisk stop
```

Modifica **/usr/share/festival/festival.scm** y añade el siguiente código antes de la última línea de código (**provide 'festival'**). En este archivo se establecen las configuraciones y parámetros de funcionamiento del sistema de síntesis de voz **Festival**, definiendo opciones como el idioma, la voz predeterminada y otros ajustes relacionados con la conversión de texto a voz. También puede cargar módulos adicionales y definir funciones en el lenguaje Scheme para personalizar el comportamiento del sistema.

```
sudo vim /usr/share/festival/festival.scm

(define (tts_textasterisk string mode)
(let ((wholeutt (utt.synth (eval (list 'Utterance 'Text string))))))
(utt.wave.resample wholeutt 8000)
(utt.wave.rescale wholeutt 5)
(utt.send.wave.client wholeutt)))
```

A continuación, se crea un servicio en **/etc/systemd/system/festival.service**. Este archivo se define el servicio **Festival** para **systemd**, permitiendo que se ejecute en segundo plano como un servidor de síntesis de voz. Se especifica cómo y cuándo debe iniciarse, qué usuario lo ejecutará y cómo manejar sus registros.

```
Description=Servicio para Festival TTS
Wants=network.target
After=syslog.target network-online.target

[Service]
Type=simple
ExecStart=/usr/bin/festival --server
Restart=on-failure
RestartSec=10
KillMode=process

[Install]
WantedBy=multi-user.target
```

Se recargan los servicios y se habilita el servicio

```
sudo systemctl daemon-reload  
sudo systemctl enable festival.service
```

Descarga y descomprime el paquete de voces en español de la Junta de Andalucía y copia las voces a la carpeta correspondiente

```
wget https://github.com/franjvasquezg/festival-spanish-voices/archive/master.zip  
unzip master.zip  
cd festival-spanish-voices-master/  
sudo mkdir /usr/share/festival/voices/Spanish  
sudo cp -r JuntaDeAndalucia_es_* /usr/share/festival/voices/spanish/
```

Con el siguiente comando puedes comprobar que se han movido correctamente las dos carpetas correspondientes a los audios. JuntaDeAndalucia_es_pa_diphone se corresponde con la voz masculina y JuntaDeAndalucia_es_sf_diphone con la femenina

```
ls -l /usr/share/festival/voices/spanish/
```

Modifica **/etc/festival.scm** y añade el siguiente contenido. En este archivo se pueden definir configuraciones personalizadas para **Festival**, como la selección de la voz predeterminada, ajustes de síntesis de voz y parámetros específicos para su funcionamiento. Se coge la voz masculina por defecto.

```
(set! voice_default 'voice_JuntaDeAndalucia_es_pa_diphone)
```

El último archivo que deberemos modificar es **/etc/asterisk/festival.conf**, en el cual solo se des comentarán las siguientes líneas de código. En este archivo se configura la integración entre **Asterisk** y **Festival**, permitiendo que el sistema de telefonía IP utilice el motor de síntesis de voz para generar mensajes hablados en llamadas. Se definen parámetros como el método de conexión y las opciones de síntesis.

```
[general]  
host=localhost  
port=1314  
festivalcommand=(tts_textasterisk "%s" 'file)(quit)\n
```

Reiniciamos el servicio de festival y volvemos a iniciar Asterisk

```
sudo systemctl restart festival.service  
sudo /etc/init.d/asterisk start
```

Reiniciamos Ubuntu y comprobamos que festival está activo

```
sudo reboot  
sudo systemctl status festival.service
```

```
chelseafh@lrss:~$ sudo systemctl status festival.service
● festival.service
    Loaded: loaded (/etc/systemd/system/festival.service; enabled; preset: enabled)
      Active: active (running) since Tue 2025-04-01 16:33:22 UTC; 1min 42s ago
        Main PID: 3782 (festival)
          Tasks: 1 (limit: 2271)
        Memory: 316.2M (peak: 316.5M)
          CPU: 447ms
        CGroup: /system.slice/festival.service
                  └─3782 /usr/bin/festival --server

abr 01 16:33:22 lrss systemd[1]: Started festival.service.
abr 01 16:33:22 lrss festival[3782]: server      Tue Apr  1 16:33:22 2025 : Festival serv>
Lines 1-12/12 (END)
```

Imagen 24: Comprobación de que festival está activo

Accede a Asterisk y comprueba que se ha configurado festival correctamente

```
sudo asterisk -rvvvv
core show application festival
```

```
lrss*CLI> core show application festival
  -= Info about Application 'Festival' =-
  [Synopsis]
  Say text to the user.

  [Since]
  1.6.2.0

  [Description]
  Connect to Festival, send the argument, get back the waveform, play it to the
  user, allowing any given interrupt keys to immediately terminate and return the
  value, or 'any' to allow any number back (useful in dialplan).

  [Syntax]
  Festival(text[,intkeys])

  [Arguments]
  Not available

  [See Also]
  Not available
```

Imagen 25: Comprobación de que festival se ha configurado correctamente en Asterisk

Por último, solo queda configurar el servicio en **extensions.conf** en el que únicamente se tendrá que añadir la siguiente línea de código en el número que quieras que diga cierta frase o dato.

```
exten => 1010 n,Festival(Esta es una frase generada con Festival)
```

Actualiza el diaplan y prueba la nueva configuración

```
sudo asterisk -rvvvv
dialplan reload
```



Menú IVR:

En Marchel nos importa dar un buen servicio a nuestros clientes (usuarios 2XXX) por tanto se ha creado un menú IVR que funciona como una operadora de atención al cliente para atender las distintas necesidades de nuestros usuarios.

El primer paso es crear un usuario en el archivo **pjsip.conf** que funcione como operadora del menú dentro del contexto **[operadora-menu]**, en este caso se ha creado el usuario 1010 de forma análoga a los usuarios 2XXX que se han explicado al inicio de la memoria ()

- [1010]
type=endpoint
context=operadora-menu
disallow=all
allow=ulaw
auth=1010-auth
aors=1010

- [1010-auth]
type=auth
auth_type=userpass
username=1010
password=Hola123

- [1010]
type=aor
max_contacts=1

El segundo paso es configurar el direccionamiento de esta extensión en el **extensions.conf**. Cabe destacar que antes de comenzar la lógica del menú IVR es necesario crear un direccionamiento condicionado dentro del mismo contexto en el que se hayan configurado los usuarios o clientes en este caso, que pueden o se desea que llamen al menú IVR. Es decir, en nuestro caso se tienen los usuarios 2XXX dentro del contexto **office-phone**. Se quiere que estos usuarios sean los que contacten con la operadora, por tanto, en dicho contexto se añade esta línea:

- [office-phone]
exten => 1010,1, Goto(operadora-menu,s,1)

La aplicación Goto() se utiliza para realizar saltos dentro del diaplan ya sea dentro del mismo contexto o de un contexto a otro. sigue la sintaxis **Goto([contexto], extensión, prioridad)**. En caso de querer saltar a otro contexto este se debe especificar en caso contrario se puede dejar en blanco. Por otra parte, es obligatorio especificar la extensión y el numero de prioridad dentro de dicha extensión a la que se quiere saltar. Esta aplicación será muy útil para la lógica del menú.

Dentro del propio archivo extension.conf se ha creado la siguiente lógica para la

Sistema IVR de Atención al Cliente

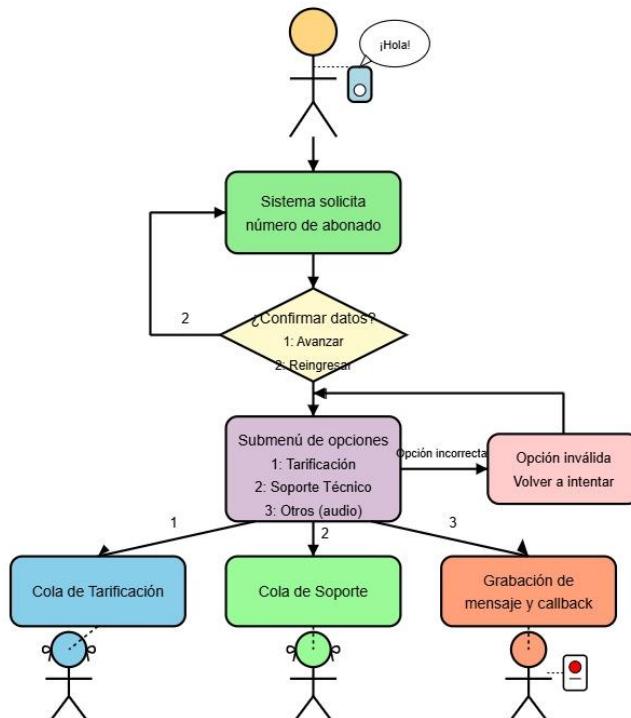


Imagen 26: Diagrama de flujo sobre el funcionamiento del Menú IVR

Operadora Menu:

```

• [operadora-menu]
exten => s,1,Answer()
exten => s,2,Playback(custom/BienvenidoMenu)
exten => s,3,Read(NUMBER,beep,9)
exten => s,4,Playback(custom/NumCorrectos)
exten => s,5,SayDigits(${NUMBER})
exten => s,6,WaitExten()

```

```

exten => 1,1,Goto(operadora-submenu,s,1)
exten => 2,1,Goto(s,2)

```

```

exten => i,1,Verbose(1, Extension Invalida)
exten => i,n,Playback(custom/OpcionVal)
exten => i,n,Goto(s,6)

```

```
exten => t,1,Verbose(1,Colgado,por Timeout)
exten => t,n,Congestion(3)
exten => t,n,Hangup()
```

Una vez que el usuario entra el Menú este debe identificarse con su número de abonado (2XXX), dicho numero será recogido en la variable NUMBER mediante la aplicación Read(NUMBER,beep,9), que sigue como sintaxis **Read (variable, audio, número_maximo)**. En este caso se ha seleccionado como audio el típico beep antes de introducir dígitos.

Cuando el usuario ya ha introducido su número, se utiliza la aplicación **SayDigits()** para repetir al cliente el número que se ha registrado y así comprobar que esta todo correcto. En caso de que el usuario marque la extensión 1 continuara al submenú para elegir el servicio. Pero en caso de seleccionar la opción 2 volverá a la prioridad 2 para repetir el número de abonado. Estos saltos se realizan mediante la aplicación **Goto()**.

Por otra parte, en caso se introducir una opción no valida se utiliza la extensión **exten => i,1** que maneja **entradas no válidas** en un IVR. Además, se utiliza la aplicación **Verbose()** muestra un mensaje en la consola de Asterisk. En caso de no seleccionar ninguna opción la operadora dispone de un timeout que gestiona mediante la extensión **exten => t,1**

A continuación, una vez que el usuario se ha identificado correctamente se pasa al submenú que recoge la lógica de los servicios que ofrece el menú:

- [operadora-submenu]
exten => s,1,BackGround(custom/SubmenuAccs)
exten => s,2,WaitExten()

```
exten => 1,1,Answer()
exten => 1,n,festival(Un agente de tarificación se pondrá en contacto con usted.
Recuerde que puede consultar sus datos de tarificación en nuestra página web oficial
www.marchel.com)
exten => 1,n,Goto(soporte-tecnico,1100,1)
```

```
exten => 2,1,Playback(custom/AgenteContacto)
exten => 2,n,Goto(soporte-tecnico,1000,1)
exten => 2,n,NoOp("Número ingresado: ${NUM}")
exten => 2,n,Hangup()
```

```
exten => 3,1,Playback(custom/DescribaBrev)
exten => 3,n,Record(/tmp/numero.wav,3,10,q)
exten => 3,n,festival(Su mensaje ha sido guardado en nuestro sistema un agente lo
analizara y se pondrá en contacto con usted en breve)
exten => 3,n,Hangup()
```

```
exten => t,1,Verbose(1,Colgado,por Timeout)
exten => t,n,Congestión(3)
exten => t,n,Hangup()
```

```

exten => i,1,Verbose(1, Extension Invalida)
exten => i,n,Playback(custom/OpcionVal)
exten => i,n,Goto(operadora-submenu,s,2)

```

Una vez que el usuario entra en el submenú escucha un audio con información sobre los servicios que ofrece la operadora en cada extensión. En este caso no se utilizará la aplicación **Playback()** sino que se empleará la aplicación **BackGround()** ya que esta última permite al usuario interrumpir el audio para marcar la extensión que desea sin la necesidad de escuchar todas las opciones.

La **primera opción** del menú permite al usuario entrar en una **cola de agentes** que se encargará de obtener información sobre tarificación, como el paquete contratado por el cliente, los precios de tarificación, el consumo que del usuario o cualquier otra duda que pueda surgirle al cliente. En la **segunda opción** el usuario entraría también en una cola, pero en esta ocasión se trata de agentes del **soporte técnico** que pueden solucionar dudas como problemas con el servicio telefónico, o con algún dispositivo vinculado a la empresa. El funcionamiento de estas colas se detallará en el siguiente apartado.

Si el usuario no tiene claro que servicio se ajusta más a su problema puede seleccionar la **tercera opción**. Esta opción recoge en un audio con el problema que plantea el cliente con la idea de que este sea analizado por los agentes y según la solución que le puedan proporcionar los propios agentes se pondrán en contacto con el cliente.

Por último, el submenú igual que el menú gestiona tanto las extensiones no validas como como el salto del timeout ante la falta de actividad por parte del usuario.

Llamadas en Cola:

En Marchel las llamadas en cola se crearán en el contexto de soporte técnico. De forma tal que existen 3 agentes que atienden a las llamadas de los clientes.

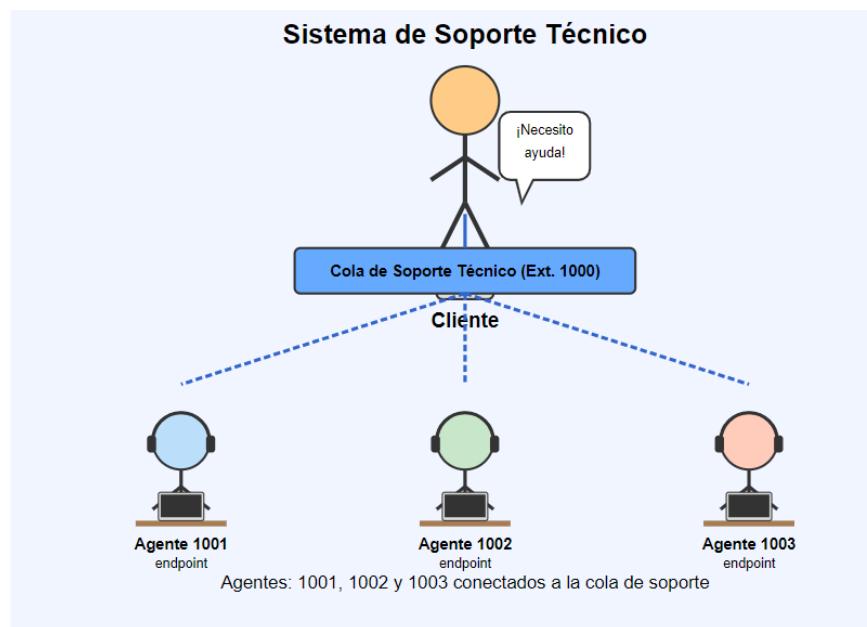


Imagen 27: Diagrama de flujo de funcionamiento de la cola de llamadas del soporte técnico

Para la creación de los agentes de forma análoga a como se crearon los primeros usuarios se crearán ahora los agentes 100X en el archivo **pjsip.conf**

- [1001-support]
type=endpoint
context=soporte-tecnico -> contexto del extensions.conf
disallow=all
allow=ulaw
auth=1001-auth
aors=1001-support

- [1001-auth]
type=auth
auth_type=userpass
username=1001-support
password=Hola123

- [1001-support]
type=aor
max_contacts=1

Una vez creado mínimo 3 agentes en pjsip.conf en momento de configurar el funcionamiento de la cola en el archivo **queues.conf**:

- [soporte-tecnico] -> nombre de la cola
musicclass=default -> configurado anteriormente
strategy= leastrecent
timeout=15
retry=5
 maxlen=5
wrapuptime=10

- member=>PJSIP/1001-support
- member=>PJSIP/1002-support
- member=>PJSIP/1003-support

En el código anterior se ha creado la configuración dentro del **contexto** soporte técnico. Se han establecidos parámetros como la música que escuchara el cliente mientras espera a ser atendido, **musicclass** se debe igualar al contexto dentro del cual se ha haya configurado la canción o canciones que se quieren escuchar.

Además, se ha configurado que, en 15 segundos si la llamada no es contestada por el agente, esta se redirigirá a otro agente y así sucesivamente hasta que alguno conteste (**timeout = 15**).

Con el parámetro **retry** se define el tiempo que espera el sistema antes de volver a llamar a un agente. Es decir, se agotan los 15 segundos y no se ha contestado, espera 5 segundo y llama a otro agente. Este tiempo de espera entre llamadas evita congestiones en la red.

Con el parámetro **maxlen** se establecen el número máximo de llamadas en cola, en Marchel al tener 3 agentes activos se ha establecido como número máximo 5 para que los agentes no se saturen y puedan atender de forma correcta a los clientes.

Por último, el parámetro más importante a configurar es la estrategia (**strategy**) en sí que seguirán las colas a la hora de realizar las llamadas. Las distintas posibles estrategias a seguir son las siguientes:

- **Ringall** → Sonará en todos al mismo tiempo hasta que uno conteste.
- **Linear** → Llama a los miembros en orden.
- **Leastrecent** → Llama al agente que no ha atendido una llamada en más tiempo.
- **Random** → Llama a los agentes de forma aleatoria.

Por último, una vez configurados los agentes en el `pjsip.conf` y la estructura de las colas en el `queues.conf` es momento de configurar el funcionamiento en el **extensions.conf**:

<pre>• [soporte-tecnico] exten => 1000,1,Answer() exten => 1000,n,Playback(custom/EsperaUnMom) exten => 1000,n,Queue(soporte-tecnico,TrW,,,45) exten => 1000,n,Playback(custom/NoContactar) exten => 1000,n,Gotolif("\${QUEUESTATUS}" = "FULL"?overflow,1) exten => 1000,n,Congestion(3) exten => 1000,n,Hangup()</pre>
<pre>exten => overflow,1,festival(Todos nuestros agentes se encuentran ocupados inténtelo más tarde) exten => overflow,n,Hangup()</pre>

La aplicación `Queue()` que pone en marcha la lógica de las colas sigue la siguiente sintaxis: **Queue(nombre_de_cola, opciones, URL, anunciador, tiempo_espera)**.

El **nombre de colas** es el contexto que se ha utilizado en el archivo `queues.conf`. En este caso se ha decidido llamar a todos los contextos relaciones soporte técnico para seguir una lógica que ayude a la comprensión. Pero dentro del contexto soporte-tecnico del extensions se podrían tener distintos sistemas de colas cada uno con su contexto en el `queues.conf`. Por ejemplo: soporte-tecnico-luz, soporte-tecnico-gas y soporte-tecnico-energiarenovable, estos serían los distintos nombres de cola a la hora de llamar a cada `Queue()`.

Las posibles **opciones** se encuentran recogidas en la siguiente tabla. Cabe destacar que se pueden utilizar varias a la vez, por ejemplo, `rtn`.

Opción	Descripción
<code>t</code>	Permite al llamante transferir la llamada si está permitido en el dialplan.
<code>T</code>	Permite a los agentes transferir la llamada.

r	No reproduce el tono de espera (ringing) al llamante.
N	No pone la llamada en la cola si no hay agentes disponibles. Si no hay agentes, sigue con la siguiente prioridad del dialplan.
h	Permite que el llamante cuelgue presionando #.
H	Permite que el agente cuelgue la llamada presionando #.
C	Registra la llamada en el CDR (Call Detail Record) solo cuando el agente la conteste.
d	Permite que el llamante marque * para salir de la cola y ejecutar la extensión s en el contexto exit.
c	Muestra la posición del llamante en la cola y el tiempo estimado de espera.
g	Si el agente cuelga, Asterisk sigue ejecutando el dialplan en lugar de terminar la llamada.
q	Silencia los anuncios de posición en la cola para el llamante.
k	Permite que el agente "parquee" la llamada presionando ##.

Tabla 2: Opciones de las colas de llamadas

El parámetro **URL** permite ejecutar una URL externa cuando la llamada entra en la cola, en esta práctica no se utiliza. Sería interesante utilizarlo en caso de tener una página web que ayude a la lógica del funcionamiento de las colas, como por ejemplo conectarlas a una base de datos que almacene información sobre el tiempo de espera, cuanto tardan las llamadas y una recolección de datos con fines estadísticos.

El parámetro **anunciador** se utiliza para reproducir un mensaje una vez que el usuario entra en la cola, en este caso no se ha utilizado porque previamente se reproduce un mensaje con **Playback()** antes de que el usuario entre en la cola.

Por otra parte, el parámetro de **tiempo de espera** establece el tiempo máximo que está el usuario en el sistema de colas, si en ese tiempo no le han contestado los agentes, la lógica del programa pasa a ejecutar la siguiente prioridad. En este caso, la siguiente prioridad es reproducir un mensaje que informa al cliente de que no se ha podido contactar con el soporte técnico.

Por último recordaras, que en el archivo queues.conf se configuró un máximo de usuarios en cola para seguir con dicha lógica se ha utilizado la aplicación Gotolf(\${QUEUESTATUS} = "FULL"?overflow,1). **Gotolf()** es una función condicionada de Asterisk que permite saltar a otras extensiones y prioridades si se cumple la condición indicada. En este programa lee el estatus de queues y en caso de estar **full** (completas), es decir, sobrepasar ese número máximo, salta el **overflow** (desbordamiento) prioridad 1 y se reproduce un mensaje que informa al cliente de que los agentes están ocupados y no pueden atenderle en ese momento.

Conexión entre PBXs:

Para establecer una conexión entre dos PBXs distintas se ha de configurar los archivos **pjsip.conf** y **extensions.conf** en los que se establecerá la IP del otro servidor mediante un troncal y los usuarios que queremos conectar entre sí. Para ello se comenzará configurando el troncal del otro servidor en el **pjsip.conf** siguiendo los siguientes pasos:

Configura el **endpoint**, debes poner el contexto que hayas asignado a los usuarios con los que deseas llamar a la otra PBX. Además, se añaden los parámetros **outbound_auth** define credenciales para autenticación saliente y **direct_media** que activa o desactiva la conexión directa entre extremos, útil si hay NAT involucrado.

- [servidorA]
type=endpoint
context=office-phone
transport=transport-udp
disallow=all
allow=ulaw, alaw, gsm
aors=servidorA-aor

Configura la autenticación (**auth**), aquí el único campo que no se ha visto anteriormente es **realm** que se emplea para identificar la autenticación válida dentro del otro servidor.

- [servidorA-auth]
type=auth
auth_type=userpass
username=servidorA
password=Hola123
realm=servidorA

Configura el **aor** (Address of Record), en **contact** debes poner la ip del otro servidor.

- [servidorA-aor]
type=aor
contact=sip:192.168.1.78

Configura el registro (**registration**) de la otra PBX.

- [servidorA-registration]
type=registration
outbound_auth=servidorA-auth
server_uri=sip:192.168.1.78
client_uri=sip:servidorA@192.168.1.78
retry_interval=60

Configura la identidad (**identify**) del otro servidor para permitir que las llamadas desde el otro servidor sean identificadas por tu PBX

- [servidorA-identify]

type=identify

endpoint=servidorA

match=192.168.1.78

En el otro servidor se hará lo mismo de forma análoga.

Una vez configurado esto, pasaremos a configurar el **extensions.conf**. En este caso se ha configurado de forma que cualquier número que comience con 2XXX o 6XXX se marcará a través de servidorA. Esto se ha configurado de esta forma, ya que los usuarios que se quieren conectar son los 2XXX con los 6XXX del otro servidor.

- [office-phone]

...

exten => _2XXX,1,Dial(PJSIP/\${EXTEN}@servidorA,25)

same => n,Hangup()

exten => _6XXX,1,Dial(PJSIP/\${EXTEN}@servidorA,25)

same => n,Hangup()

Solo queda iniciar Asterisk recargar los servicios y comprobar que funcione llamando a los usuarios 6001, 6002...

Instalación de MariaDB:

Para instalar y configurar la base de datos para que esta registre los diferentes parámetros cada vez que se realiza una llamada se deben seguir los siguientes pasos:

Se deben actualizar los paquetes del sistema, esto lo puedes realizar con el siguiente comando:

```
sudo apt update && sudo apt upgrade
```

Se procede a la instalación de MariaDB y su habilitación.

```
sudo apt install mariadb-server mariadb-client
sudo systemctl enable mariadb
```

Puedes comprobar que está activa con el siguiente comando:

```
sudo systemctl status mariadb
```

```
mariadb.service - MariaDB 10.11.8 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-03-30 19:10:22 UTC; 2min 53s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
Main PID: 1934 (mariadb)
   Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 14998)
     Memory: 78.8M (peak: 81.8M)
        CPU: 240ms
       CGroup: /system.slice/mariadb.service
               └─1934 /usr/sbin/mariadb

mar 30 19:10:22 asterisk mariadb[1934]: 2025-03-30 19:10:22 0 [Note] Plugin 'FEEDBACK' is disabled.
mar 30 19:10:22 asterisk mariadb[1934]: 2025-03-30 19:10:22 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
mar 30 19:10:22 asterisk mariadb[1934]: 2025-03-30 19:10:22 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.
mar 30 19:10:22 asterisk mariadb[1934]: 2025-03-30 19:10:22 0 [Note] Server socket created on IP: '127.0.0.1'.
mar 30 19:10:22 asterisk mariadb[1934]: 2025-03-30 19:10:22 0 [Note] InnoDB: Buffer pool(s) load completed at 250330 19:10:22
mar 30 19:10:22 asterisk mariadb[1934]: 2025-03-30 19:10:22 0 [Note] /usr/sbin/mariadb: ready for connections.
mar 30 19:10:22 asterisk mariadb[1934]: Version: '10.11.8-MariaDB-10.11.8-24.04.1' socket: '/run/mysqld/mysqld.sock' port: 3306  Ubuntu 24.04
mar 30 19:10:22 asterisk systemd[1]: Started mariadb.service - MariaDB 10.11.8 database server.
mar 30 19:10:22 asterisk /etc/mysql/debian-start[1992]: Upgrading MariaDB tables if necessary.
mar 30 19:10:22 asterisk /etc/mysql/debian-start[1993]: Checking for insecure root accounts.
```

Imagen 28: Comprobación de que MariaDB se encuentra activa



A continuación, se configura MariaDB para proteger la base de datos.

```
sudo mysql_secure_installation
```

Establece una contraseña y contesta a las diferentes preguntas, se recomiendan las siguientes configuraciones:

- Set root password? → Sí
- Remove anonymous users? → Sí
- Disallow root login remotely? → Sí
- Remove test database? → Sí
- Reload privilege tables now? → Sí

Accede a MariaDB y modifica la contraseña del usuario root

```
sudo mysql -u root  
UPDATE mysql.user SET PASSWORD=PASSWORD('root') WHERE USER='root';  
flush privileges;
```

Ahora configura MariaDB para que siempre nos pida la contraseña al entrar

```
UPDATE mysql.user SET plugin = " WHERE user = 'root';
```

Usa **quit** para salir y reinicia MariaDB

```
sudo systemctl restart mysql
```

Accede a MariaDB y crea la base de datos para Asterisk

```
Sudo mysql -u root -p  
CREATE DATABASE asterisk;
```

Crea un usuario y dale privilegios

```
CREATE USER 'asterisk'@'localhost' IDENTIFIED BY 'contraseña';  
GRANT ALL PRIVILEGES ON asterisk.* TO 'asterisk'@'localhost';  
FLUSH PRIVILEGES;
```

Puedes comprobar que lo has creado correctamente con este comando:

```
SELECT user, host FROM mysql.user;
```

```

MariaDB [mysql]> select user, host from user;
+-----+-----+
| User   | Host    |
+-----+-----+
| asterisk | localhost |
| mariadb.sys | localhost |
| mysql     | localhost |
| root      | localhost |
+-----+-----+
4 rows in set (0,001 sec)

MariaDB [mysql]> █

```

Imagen 29 Comprobación de creación del usuario asterisk dentro de MariaDB

Y a su vez comprobar que le has dado correctamente los permisos con:

```
SHOW GRANTS FOR 'asterisk'@'localhost';
```

```

MariaDB [mysql]> SHOW GRANTS FOR 'asterisk'@'localhost';
+-----+
| Grants for asterisk@localhost          |
+-----+
| GRANT USAGE ON *.* TO `asterisk`@`localhost` IDENTIFIED BY PASSWORD '*E6B7BA6C
6A533AA2D9B38A010964B5588EBA402D' |
| GRANT ALL PRIVILEGES ON `asterisk`.* TO `asterisk`@`localhost`           |
+-----+
2 rows in set (0,000 sec)

```

Imagen 30: Comprobación de permisos del usuario asterisk

Por consiguiente, accede a la base de datos y crea una tabla para que almacene los datos de las llamadas, esta tabla se puede modificar para que te muestre únicamente los campos que a ti te interesen o que guarde solo estos cuando tengas determinados valores. Por ejemplo, en el campo **duration**, a lo mejor te interesa tener un registro de todas las llamadas, aunque estas no sean contestadas, entonces podrías quitar el parámetro **NOT NULL**. Un ejemplo de los campos que se pueden incluir en la tabla, llamados también CDRs (Call Detail Records) es el siguiente:

Campo	Descripción
calldate	Fecha y hora en que comenzó la llamada.
clid	Caller ID (número del que llama).
src	Número de origen de la llamada.
dst	Número de destino de la llamada.
dcontext	Contexto del dialplan donde ocurrió la llamada.
channel	Canal de origen de la llamada (ejemplo: SIP/101-0000001).
dstchannel	Canal de destino de la llamada (ejemplo: SIP/202-0000002).
lastapp	Última aplicación ejecutada en el dialplan (ejemplo: Dial).
lastdata	Argumentos de la última aplicación ejecutada.
start	Hora de inicio de la llamada (igual a calldate).

answer	Hora en que la llamada fue contestada.
end	Hora en que la llamada finalizó.
duration	Duración total de la llamada (en segundos).
billsec	Duración facturable de la llamada (en segundos).
disposition	Estado final de la llamada (ANSWERED, BUSY, FAILED, NO ANSWER).
amaflags	Bandera de tarificación (DOCUMENTATION, BILLING, IGNORE).
accountcode	Código de cuenta (usado en sistemas de facturación).
uniqueid	Identificador único de la llamada.
linkedid	Identificador único que agrupa llamadas relacionadas (por ejemplo, transferencias).
sequence	Número secuencial del CDR.
userfield	Campo personalizable para agregar información adicional.
peeraccount	Cuenta asociada al usuario remoto.
hangupcase	Código de causa de colgado (según estándar Q.850).
hangupsource	Indica qué lado colgó la llamada (ejemplo: SIP/101).
peer	Nombre del dispositivo remoto.
callerid	Identificador de llamada.

Tabla 3: Opciones de los CDRs que pueden ser recogidos de Asterisk en MariaDB

El siguiente comando se usa para crear la tabla:

```
use asterisk
CREATE TABLE cdr (
    id INT PRIMARY KEY AUTO_INCREMENT,
    calldate DATETIME NOT NULL,
    clid VARCHAR(80) NOT NULL,
    src VARCHAR(80) NOT NULL,
    dst VARCHAR(80) NOT NULL,
    dcontext VARCHAR(80) NOT NULL,
    channel VARCHAR(80) NOT NULL,
    dstchannel VARCHAR(80),
    lastapp VARCHAR(80) NOT NULL,
    lastdata VARCHAR(80) NOT NULL,
    duration INT NOT NULL,
    billsec INT NOT NULL,
    disposition VARCHAR(45) NOT NULL,
    amaflags INT NOT NULL,
    accountcode VARCHAR(20),
    uniqueid VARCHAR(32) NOT NULL,
    userfield VARCHAR(255),
    peeraccount VARCHAR(80),
    linkedid VARCHAR(32),
    sequence INT
);
```

Comprueba que se ha creado correctamente

```
DESCRIBE cdr;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
calldate	datetime	NO		NULL	
clid	varchar(80)	NO		NULL	
src	varchar(80)	NO		NULL	
dst	varchar(80)	NO		NULL	
dcontext	varchar(80)	NO		NULL	
channel	varchar(80)	NO		NULL	
dstchannel	varchar(80)	YES		NULL	
lastapp	varchar(80)	NO		NULL	
lastdata	varchar(80)	NO		NULL	
duration	int(11)	NO		NULL	
billsec	int(11)	NO		NULL	
disposition	varchar(45)	NO		NULL	
amaflags	int(11)	NO		NULL	
accountcode	varchar(20)	YES		NULL	
uniqueid	varchar(32)	NO		NULL	
userfield	varchar(255)	YES		NULL	
peeraccount	varchar(80)	YES		NULL	
linkedid	varchar(32)	YES		NULL	
sequence	int(11)	YES		NULL	

20 rows in set (0,001 sec)

Imagen 31: Comprobación de la creación de la tabla cdr

Sal de MariaDB con **quit;**

Una vez realizada la configuración de MariaDB debemos asegurarnos de que estén instaladas las siguientes dependencias en caso de no haberlas instalado anteriormente:

```
sudo apt install build-essential uuid-dev libjansson-dev libxml2-dev libsqlite3-dev
sudo apt install libmariadb-dev libmariadb-dev-compat
```

A se debe instalar un conectorODBC que se encargará de conectar la base de datos de MariaDB con el servicio de Asterisk.

Para ello te recomiendo que descargas el conector más adecuado para la versión de tu sistema operativo en esta página <https://dev.mysql.com/downloads/connector/odbc/>

```
sudo apt-get install unixodbc unixodbc-dev
sudo wget https://dev.mysql.com/get/Downloads/Connector-ODBC/9.2/mysql-
connector-odbc_9.2.0-1ubuntu24.04_amd64.deb
sudo dpkg -i mysql-connector-odbc_9.2.0-1ubuntu24.04_amd64.deb
sudo apt install odbcinst
```

A continuación, configura el menú de selección si no lo hiciste o configuraste en la instalación de Asterisk:

```
cd /usr/src/asterisk-22.2.0-rc2
sudo make menuselect
```

Comprueba que tengas marcadas las siguientes casillas, sobre todo aquellas en las que ponga mysql u odbc. Lo puedes comprobar ([aqui](#))

Pulsa en **Save & Exit** y en caso de no tener alguno de los módulos activados anteriormente instálalos.

```
sudo make install
```

A continuación, se detallan los archivos clave a modificar para configurar la integración entre Asterisk y la base de datos MySQL (MariaDB):

/etc/asterisk/cdr_adaptive_odbc.conf

Este archivo contiene la configuración para la conexión adaptativa de ODBC en Asterisk. Se usa para almacenar los registros de llamadas y sus detalles en la tabla que se ha configurado en la base de datos de MariaDB.

El parámetro **connection** se refiere a la conexión definida en el archivo **res_odbc.conf**, que debe coincidir con el nombre del DSN. El campo **table** define la tabla donde se guardarán los CDRs (Call Detail Records) y el campo **alias start** relaciona el parámetro **calldate** de la tabla creada al alias start.

```
[asterisk]
connection=asterisk
table=cdr
alias start => calldate
```

/etc/asterisk/cdr_odbc.conf

En este archivo se especifica cómo Asterisk interactuará con la base de datos para almacenar los registros de llamadas (CDR), estableciendo el DSN, la tabla y otros parámetros de la conexión.

El parámetro **dns** define el nombre del DNS configurado en **/etc/odbc.ini**, este indica la conexión a la base de datos. Los campos **username** y **password** son las credenciales de acceso configuradas para el usuario anteriormente creado (al cual le dimos privilegios) en la base de datos. Los parámetros **loguniqueid**, **dispositionstring** y **usegmttime** indican si se debe registrar el identificador único de la llamada, si se debe registrar el estado final de la llamada y si se debe usar la hora GMT para registrar la hora de la llamada respectivamente. Por último, el campo **table** se comporta de la misma forma que en el archivo anterior.

```
[global]
dsn=MySQL-test
username=asterisk
password=contraseña
loguniqueid=yes
dispositionstring=yes
table=cdr
usegmttime=no
```

/etc/asterisk/cdr.conf

Aquí se configuran las opciones generales para los registros de llamadas, como la utilización de la zona horaria GMT o la localización de las fechas y horas de las llamadas.

El campo **enabled** se usa para habilitar o deshabilitar el registro de CDRs y al igual que antes se emplea **usegmttime** si se deben usar o no las horas GMT.

```
[general]
enabled = yes
usegmttime = no
```

/etc/asterisk/extconfig.conf

En este archivo se define cómo Asterisk gestionará los registros de llamadas a través de ODBC, utilizando el DSN configurado previamente.

El parámetro **cdr** muestra que los CDRs deben ser gestionados por ODBC, usando el DSN (asterisk) y la tabla (cdr).

```
[settings]
cdr => odbc,asterisk,cdr
```

/etc/asterisk/res_odbc.conf

En este archivo se especifica la configuración de la conexión ODBC entre Asterisk y MariaDB, incluyendo detalles de autenticación y conexión.

El campo **enabled** habilita la conexión ODBC. Los parámetros **dns**, **username** y **password** se emplean igual que antes. Y, por último, **pre-connect** establece la conexión ODBC antes de realizar cualquier operación.

```
[asterisk]
enabled => yes
dns => MySQL-asterisk
username => asterisk
password => contraseña
pre-connect => yes
```

/etc/odbc.ini

Este archivo se emplea para configurar el DSN para la conexión de ODBC con la base de datos de Asterisk, incluyendo el servidor, la base de datos, y las credenciales de acceso.

El parámetro **Driver** define el controlador ODBC que Asterisk usará para conectarse a la base de datos, este deberá ser el que se ha instalado previamente. El campo **Description** es simplemente una breve descripción de la conexión. El parámetro **Server** es el servidor en el que se encuentra la base de datos, generalmente localhost si MariaDB está en la misma máquina.

El campo **Database** es el nombre de la base de datos creada en MariaDB. Los parámetros **user** y **password** se utilizan de la misma forma que anteriormente. En **Port** se define el puerto de conexión (por defecto 3306 para MySQL/MariaDB).

El campo **Socket** define el socket utilizado por MariaDB para la conexión local, puede saber cuál es tu socket con **sudo vim /etc/mysql/my.cnf**. Por último, el parámetro **Option** se usa porque mejora la estabilidad de la conexión y la precisión en las actualizaciones de datos.

```
[MySQL-asterisk]
Driver=MySQL ODBC 9.2 Unicode Driver
Description=MariaDB connection to 'asterisk' database
Server=localhost
Database=asterisk
User=asterisk
Password=contraseña
Port=3306
Option = 3
Socket=/var/run/mysqld/mysqld.sock
```

/etc/odbcinst.ini

Configura el controlador de ODBC necesario para establecer la conexión con la base de datos MySQL (MariaDB).

En el campo **Driver** debes definir la ruta hacia el controlador que te hayas instalado y el parámetro **UsageCount** indica cuántas veces se está utilizando dicho controlador.

```
[MySQL ODBC 9.2 Unicode Driver]
DRIVER=/usr/lib/x86_64-linux-gnu/odbc/libmyodbc9w.so
UsageCount=1
```

Con el siguiente comando puedes comprobar que se ha conectado correctamente:

```
isql -v MySQL-asterisk
```

```
chelseafh@lrss:~/archivosP2$ isql -v MySQL-asterisk
+-----+
| Connected!
|
| sql-statement
| help [tablename]
| echo [string]
| quit
|
+-----+
SQL> 
```

Imagen 32: Comprobación de que MariaDB se ha conectado correctamente a Asterisk

Utiliza **quit** para salir.

A continuación, reinicia asterisk y comprueba si está bien conectado

```
sudo systemctl restart asterisk
sudo asterisk -rx "odbc show"
```

```
chelseafh@lrss:~/archivosp2$ sudo asterisk -rx "odbc show"
ODBC DSN Settings
-----
Name: asterisk
DSN: MySQL-asterisk
Number of active connections: 1 (out of 1)
Logging: Disabled
```

Imagen 33: Comprobación de que el conector ODBC está bien conectado a Asterisk

Por último, prueba a hacer una llamada y cuando esta finalice revisa si esta se ha almacenado correctamente en la base de datos, esto lo puedes hacer con el siguiente comando dentro de la base de datos **asterisk** creada anteriormente:

```
SELECT * FROM cdr ORDER BY calldate DESC;
```

	id	calldate	clid	src	dst	dcontext	channel	dstchannel	lastapp	laststate	duration	billsec	disposition	meetflags	accountcode	uniqueid	userfield	peeraccount	linkedid	sequence
2	2025-03-13 19:44:59	2001-softphone	2001-softphone	1009	soporte-tecnico	PJSIP/2001-softphone-00000003	PJSIP/2001-softphone-00000003	PJSIP/2001-softphone-00000003	NULL	Answered	61	61	ANSWERED	2	NULL	1741895909.0	NULL	1741895909.0	2	
1	2025-03-13 19:23:32	2002-softphone	2002-softphone	2081	oficina-phone	PJSIP/2002-softphone-00000008	PJSIP/2002-softphone-00000008	PJSIP/2002-softphone-00000008	None	Ring	9	9	ANSWERED	2	NULL	1741893813.0	NULL	1741893813.0	0	

Imagen 34: Ejemplo de cómo deben salir los datos que se almacenan en la base de datos tras las llamadas

Tarificación:

En Marchel se han establecido unos precios/min de tarificación teniendo en cuenta el horario y el área de localización dentro de la cual donde se realiza la llamada.

El primer paso es crear una tabla en MariaDB con los campos que se van a necesitar para establecer los precios según las condiciones que se quieran tener en cuenta:

- CREATE TABLE tarifas (
 id INT PRIMARY KEY AUTO_INCREMENT,
 destino VARCHAR(80) NOT NULL,
 horario VARCHAR(20) NOT NULL,
 costo_por_minuto DECIMAL(10,4) NOT NULL
);

Una vez que se tiene creada la tabla es hora de dar valores a los campos. En el caso de Marchel se considerarán llamadas nacionales las realizadas dentro de la misma pbx e internaciones las realizadas entre pbxs.

- INSERT INTO tarifas (destino, horario, costo_por_minuto) VALUES
 ('nacional', 'normal', 0.05),
 ('nacional', 'nocturno', 0.02),
 ('internacional', 'normal', 0.15),
 ('internacional', 'nocturno', 0.10);

Una vez que se tiene registrado en la base de datos los precios, se va a pasar a crear una tabla que recoja la información necesaria para crear una facturación del total de las



llamadas que realiza cada usuario. Por ejemplo, a quien llama, el precio, la hora, la fecha, la duración, y el coste ya calculado por llamada.

- CREATE TABLE facturacion (
id INT PRIMARY KEY AUTO_INCREMENT,
uniqueid VARCHAR(32) NOT NULL,
src VARCHAR(80) NOT NULL,
dst VARCHAR(80) NOT NULL,
duracion INT NOT NULL,
costo DECIMAL(10,4) NOT NULL,
fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP

Con la tabla de facturación ya creada es momento de ir rellenando los campos juntando la información de la tabla de registros y la tabla de tarificación. De forma tal que tenemos 3 tablas una dinámica con todos los registros de llamadas de asterisk, una estática con la información de las tarifas y otra dinámica con la información filtrada y necesaria para la facturación. Como necesitamos que esta tabla se rellene de forma dinámica, se va a crear un tarificacion.py que automatice el proceso.

Tarificacion.py se encarga de realizar una lectura en ambas tablas y una escritura en facturación mediante estas consultas:

- SELECT
c.uniqueid,
c.src,
c.dst,
c.duration,
t.costo_por_minuto
FROM cdr c
JOIN tarifas t
ON (CASE
WHEN c.dst LIKE '2%' OR c.dst LIKE '200%' THEN 'nacional'
WHEN c.dst LIKE '6%' OR c.dst LIKE '600%' THEN 'internacional'
END) = t.destino
AND (CASE
WHEN HOUR(c.calldate) BETWEEN 22 AND 23 OR HOUR(c.calldate)
BETWEEN 0 AND 6 THEN 'nocturno'
ELSE 'normal'
END) = t.horario
WHERE NOT EXISTS (SELECT 1 FROM facturacion f WHERE f.uniqueid = c.uniqueid);

```
INSERT INTO facturacion (uniqueid, src, dst, duracion, costo, fecha)
VALUES (%s, %s, %s, %s, %s, %s);
```

Para que la ejecución de tarificacion.py se realice de forma periódica y automatizada se va a utilizar un sistema de programación de tareas en Linux que permite ejecutar comandos o scripts en intervalos de tiempo definidos automáticamente.

```
crontab -e -> en la terminal de ubuntu
```

```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m. every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
*/5 * * * * /usr/bin/python3 /home/chelseafh/p2/tarificacion.py
```

Imagen 35: Interfaz CRON

Agregamos la siguiente línea que hará que se ejecute cada 5 mins. Se ha tomado este numero como consideración inicial para que no se sobrecargue el sistema, pero al mismo tiempo sea eficiente y rápida la recarga de información.

```
*/5 * * * * /usr/bin/python3 /home/chelseafh/p2/tarificacion.py
```

Una vez que tenemos la tabla de facturación actualizada con el coste de cada llamada que hace cada usuario. Se puede obtener con la siguiente consulta el coste total de las llamadas de un usuario durante el mes.

- SELECT
src,
SUM(costo) AS costo_total
FROM facturacion
WHERE src = '2001-softphone'
AND fecha >= CURDATE() - INTERVAL 1 MONTH
GROUP BY src;

[www.marchel.com:](http://www.marchel.com)

En Marchel hemos pensado que al tener tantos datos almacenados en la base de datos podríamos utilizarlos para ofrecer un servicio de página web, en la cual el usuario puede consultar informaciones estadísticas sobre su actividad dentro de la central.

En primer lugar, es una página que recoge **información personalizada** y protegida, el usuario debe introducir su numero de abonado y una contraseña proporcionada por la compañía.



Una vez dentro el usuario puede consultar varias pestañas. En la **pestaña principal** obtendrá información resumida como costo mensual acumulado o el total de llamadas.

The image shows the main panel of the Marchel web panel. The title 'Panel de Usuario' is at the top. Below it is a navigation bar with tabs: 'Panel Principal' (which is active and highlighted in green), 'Estadísticas', 'Tarifas', 'Historial', and 'Tendencias'. The main content area displays three summary statistics in boxes: 'COSTO TOTAL DEL MES' (0.1927 €), 'DURACIÓN TOTAL' (231 seg), and 'LLAMADAS REALIZADAS' (16). At the bottom right of the panel is a 'Cerrar Sesión' (Logout) button.

Imagen 37: Web Panel Principal

En la pestaña de **estadísticas** podrá consultar gráficos estadísticos que reúnen información como el impacto del usuario en la red (compara sus llamas con las del resto de usuarios), su utilización de la red según los días de la semana, su utilización por horas del día y un gráfico comparando a que usuarios suele llamar más.

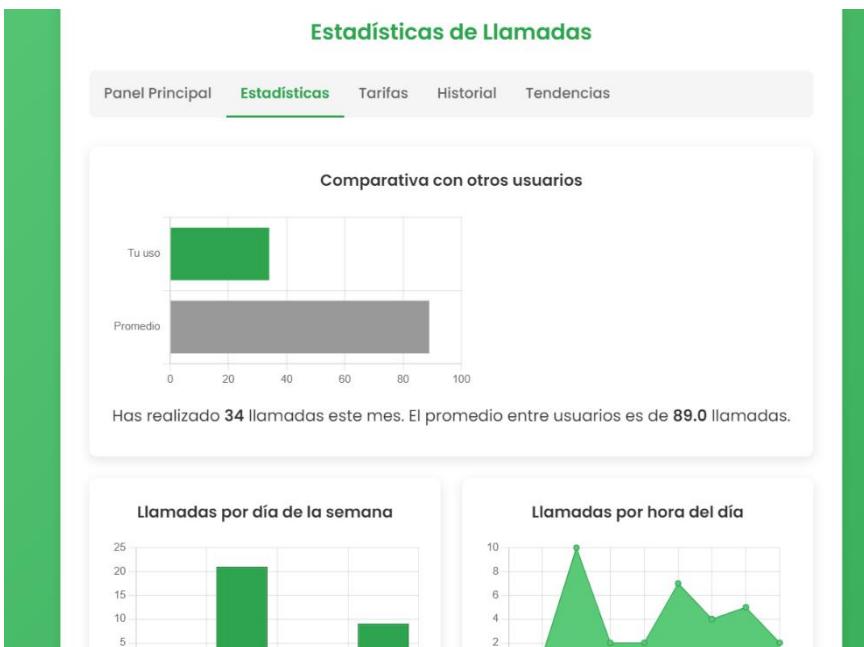


Imagen 38: Web Estadísticas

En la pestaña **tarifas** el usuario puede consultar los precios que ofrece la compañía

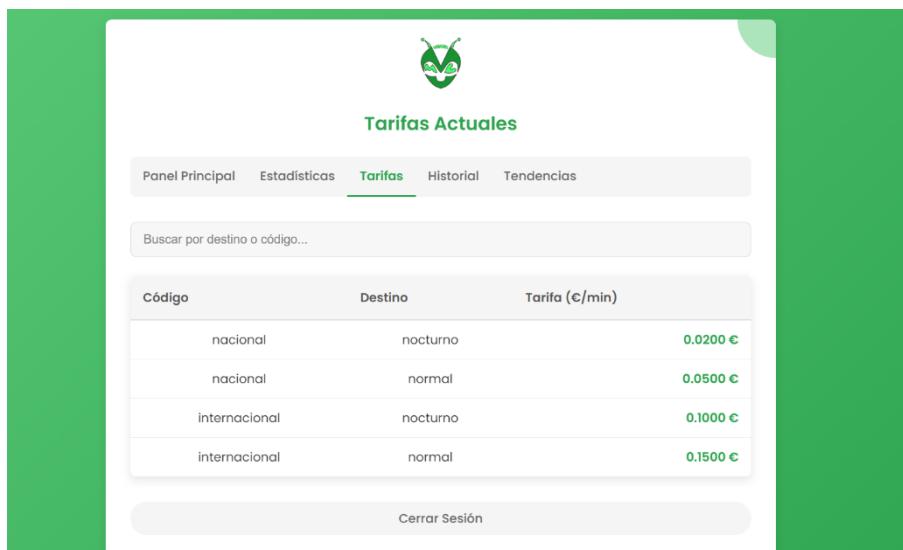


Imagen 39: Web Tarifas

En la pestaña de **historial** el usuario puede consultar todas las llamadas que ha realizado, la duración y el coste que han tenido cada una de ellas



Histórico de Llamadas

Fecha	Destino	Duración	Costo	Estado
02/04/2025 17:10	2001	262 seg	0.22 €	Contestada
02/04/2025 16:50	2001	297 seg	0.25 €	Contestada
30/03/2025 22:15	2	30 seg	0.03 €	Contestada
30/03/2025 22:15	2	47 seg	0.04 €	Contestada
29/03/2025 20:05	2001	50 seg	0.04 €	Contestada

En la pestaña de tendencias el usuario puede consultar datos estadísticos mensuales, diferencia de coste mensual, tendencias de llamadas, como varía la media de la duración de sus llamadas según el mes y un resumen final de precios, duración y total de llamadas de los últimos 6 meses

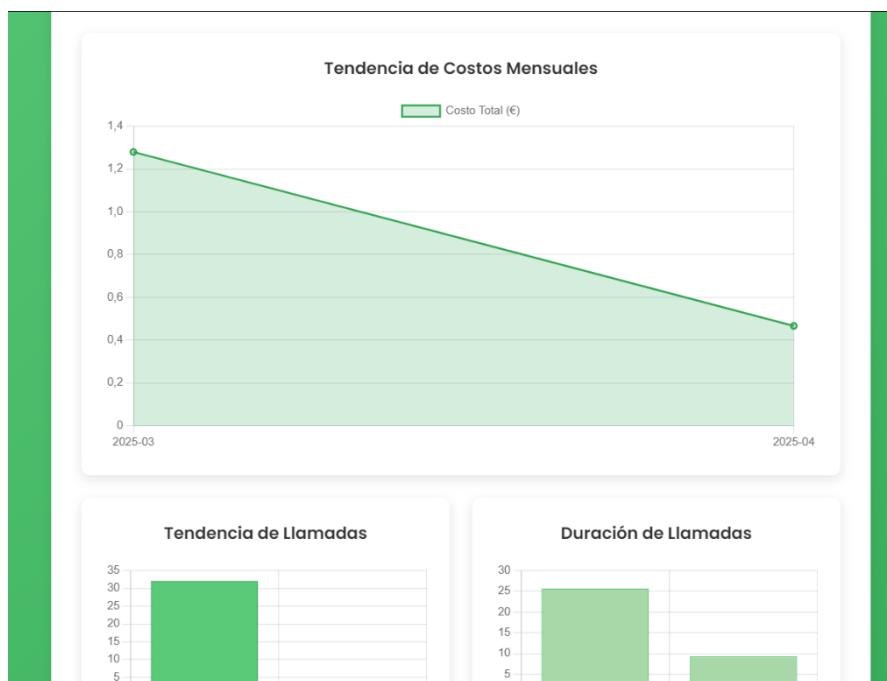


Imagen 41: Web Tendencias

El desarrollo de esta página web se realizó mediante la utilización de **Flask**. Flask es un framework web ligero para Python que permite crear aplicaciones web de manera sencilla y flexible. Se ha recogido en un app.py todas las consultas y lógicas necesarias para la creación de la aplicación web.

Conclusión:

La creación de Marchel como una central telefónica basada en Asterisk ha demostrado ser un proyecto completo y multifacético que abarca desde la configuración básica hasta funciones avanzadas de comunicación VoIP. A lo largo del desarrollo, se hizo evidente la importancia de una planificación cuidadosa y una configuración detallada para garantizar el correcto funcionamiento del sistema.

Uno de los aspectos más destacables ha sido la configuración inicial de usuarios y extensiones, ya que sentó las bases para el desarrollo del resto de funcionalidades. Por otra parte, la utilización de aplicaciones como Zoiper ha permitido validar rápidamente las llamadas entre extensiones, confirmando que la estructura básica de la PBX funciona correctamente.

La implementación de características avanzadas, como el buzón de voz, la música en espera y el menú IVR, añadieron funcionalidades básicas al proyecto. Estas, no solo mejoraron la experiencia del usuario, sino que también demostraron la flexibilidad de Asterisk para adaptarse a diferentes necesidades. Además, la configuración de transferencias de llamadas y las salas de conferencia han demostrado cómo el sistema puede facilitar la comunicación tanto en entornos empresariales como educativos.

La incorporación de una base de datos MariaDB para registrar detalladamente las llamadas (CDRs) ha sido un paso crucial para la gestión y análisis del tráfico telefónico. La automatización de procesos, como la tarificación mediante scripts en Python y su programación con CRON, ha destacado la capacidad de Asterisk para integrarse con otras herramientas y optimizar tareas repetitivas.

El desarrollo de una interfaz web utilizando Flask permitió visualizar datos de manera clara y accesible, ofreciendo a los usuarios información relevante como estadísticas de llamadas y costos. Este componente añadió un valor significativo al proyecto, mostrando cómo los datos recopilados pueden transformarse en información útil para los usuarios y empresas interesadas.

En términos generales, el proyecto ha reforzado la importancia de la documentación y la metodología estructurada. Cada desafío encontrado durante la configuración sirvió como oportunidad para profundizar en el funcionamiento de Asterisk y sus módulos. El resultado final es un sistema robusto y escalable, capaz de adaptarse a diversas necesidades de comunicación.

Esta experiencia no solo consolidó conocimientos técnicos en VoIP y administración de sistemas, sino que también destacó el potencial de Asterisk como solución integral para centralitas telefónicas. El proyecto sienta las bases para futuras mejoras, como la integración de APIs avanzadas o la implementación de medidas de seguridad adicionales.

Bibliografía:

- <https://zadarma.com/es/support/instructions/asteriskpjisp/>
- <https://www.voipdo.com/wp-content/uploads/2017/04/Laboratorio-de-contextos-especiales.pdf>

- https://github.com/javierorp/Asterisk_Tutorial?tab=readme-ov-file#48--simular-llamadas-desde-el-exterior-redirección-a-un-contexto-diferente
- <https://stackoverflow.com/questions/37422603/building-channels-trunks-between-two-asterisk-servers-with-pjsip>
- <https://medium.com/asterisk-tips-101/troncal-pjsip-entre-servidores-asterisk-48e6468ecc8e>
- <https://blackhold.nusepas.com/2020/07/23/instalar-asterisk-con-odbc-configurar-odbc-mariadb/>
- <https://lavariega.com/guia-para-usar-mariadb-en-asterisk/>
- <https://github.com/franjvasquezg/festival-spanish-voices>
- <https://www.voztovoice.org/?q=comment/484#comment-484>