# Descripción del ARA

Nos permite poder tener archivos de configuración o poder tener la funcionalidad que nos da los archivos de configuración de una base de datos. Significa que por ejemplo lo que nos brida el “sip.conf” que básicamente es crear anexos zip, así como el “extensions.conf” que es para el plan de marcado, el “queues.conf” para crear colas y así la funcionalidad que nos da varios archivos la podríamos tener en una base de datos. Esto nos permitirá simplificar la parte de comunicación, facilitando en un futuro la creación de un programa que nos facilite esto.

El ARA (Asterisk RealTime Architecture) tiene dos tipos:

* Static realtime: básicamente se ejecuta con un reloj, es decir ponemos la data en una base de datos y mediante un reloj jalamos la información y lo tiene ya el Asterisk en su memoria, sigue el formato del “.conf”.
  + Su formato es el sig.:

[settings]

archivo.file = odbc,asterisk,table (archivo que queramos = driver a usar, conector “rest\_odbc.conf”, tabla que va a interactuar)

* Dinamic realtime: básicamente es cualquier cambio que hagamos en la base de datos ya sea agregar o quitar se va a reflejar de manera automática, por lo cual no se necesita hacer un reload, reiniciar el Asterisk ni nada por el estilo, solamente modificando un archivo de configuración.
  + Su formato es el sig:

funcionalidad = odbc,asterisk,table (archivo que querase indica la funcionalidad = driver a usar, conector “rest\_odbc.conf”, tabla que va a interactuar)

La diferencia entre el dinámico y el estático, es que el dinámico convive con los archivos de configuración, mientras que el estático prácticamente reemplaza el archivo de configuración.

# Colocando el queue\_log en la base de datos.

Vamos a ver el archivo para configurar el “real time” tanto estático como dinámico, el archivo se llama “extconfig.conf”, para acceder a el primero tendremos que situarnos en el directorio “/etc/asterisk” donde usaremos el sig. comando:

nano extconfig.conf

Como siempre lo primero que hacemos es mover el archivo al bak con el sig. comando:

mv extconfig.conf extconfig.conf.bak

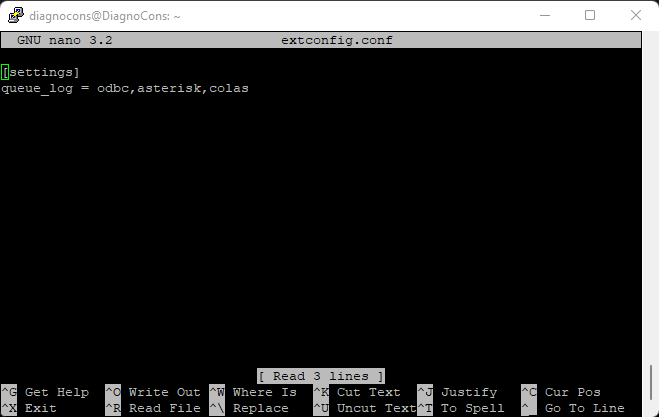
Después entramos de nuevo al “extconfig.conf”

nano extconfig.conf

Donde colocaremos lo sig.:

[settings]

queue\_log = odbc,asterisk,colas



Después guardamos el archivo.

Después de guardarlo crearemos la tabla “colas” usando el siguiente script:

CREATE TABLE colas(

id bigint(255) unsigned NOT NULL AUTO\_INCREMENT,

time varchar(26) NOT NULL DEFAULT '',

callid varchar(40) NOT NULL DEFAULT '',

queuename varchar(20) NOT NULL DEFAULT '',

agent varchar(20) NOT NULL DEFAULT '',

event varchar(20) NOT NULL DEFAULT '',

data varchar(100) NOT NULL DEFAULT '',

data1 varchar(40) NOT NULL DEFAULT '',

data2 varchar(40) NOT NULL DEFAULT '',

data3 varchar(40) NOT NULL DEFAULT '',

data4 varchar(40) NOT NULL DEFAULT '',

data5 varchar(40) NOT NULL DEFAULT '',

created timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP,

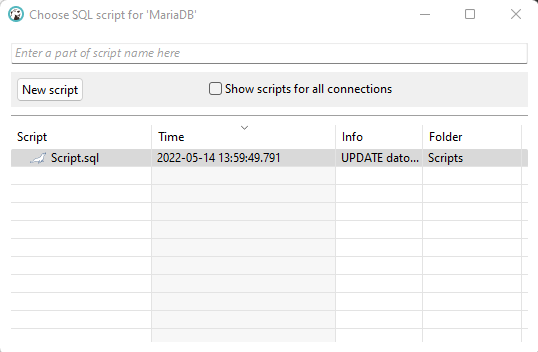
PRIMARY KEY (id),

KEY queue (queuename),

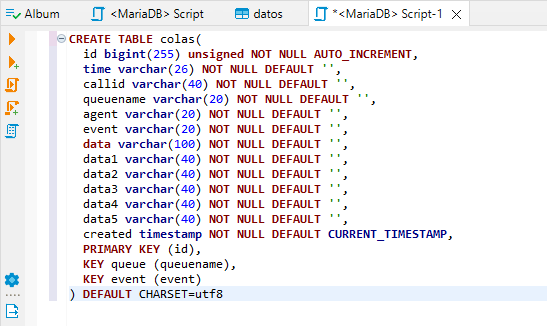
KEY event (event)

) DEFAULT CHARSET=utf8

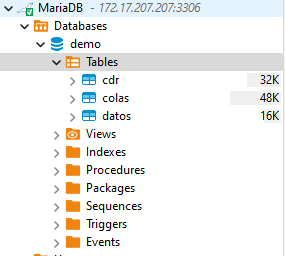
Entramos al dbeaver y en la base de datos le hacemos click derecho, luego click en “Editor SQL” y después en “Script SQL”.



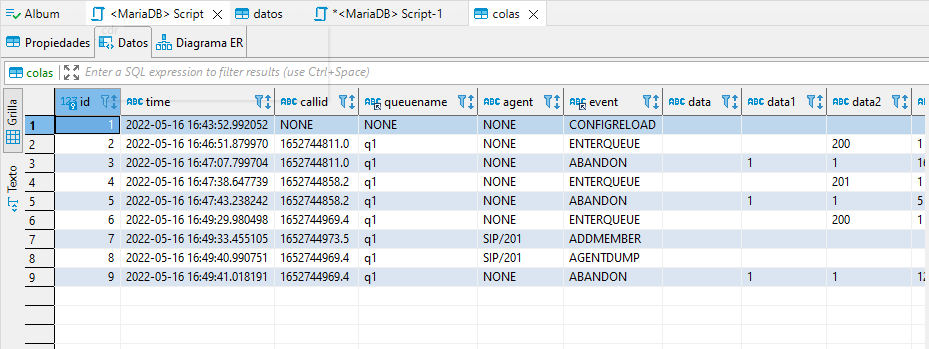
Luego en “New Script” y pegamos el script (no olvidar el USE demo al iniciar).



Luego en la base de datos “demo” en “Tables” le damos click derecho y le hacemos click en “Renovar” y nos aparecerá la nueva tabla “colas”.



Entramos al Asterisk y cargamos la consola con un “reload” y entonces ya debería de poder guardarse esa información en la base de datos. Luego hacemos llamadas a la cola (nuestra extensión es 600 en este caso) y automáticamente debería de guardarse en la base de datos.



# Ejemplo de Static Realtime, colocando queues.conf en la base de datos.

El “Static Realtime” nos permite poner un archivo de configuración en la base de datos, pero solamente va a funcionar como un reloj, es decir cuando pongamos los valores y cuando hagamos reload, recién se van a cargar en el Asterisk.

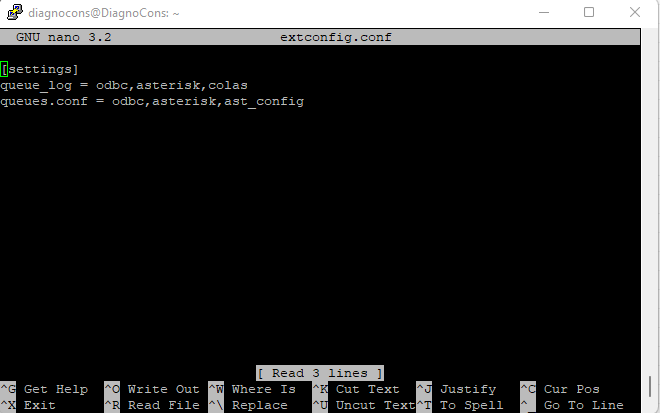
Primero entramos en el “extconfig.conf” con el sig. comando:

nano extenconfig.conf

Una vez ahí, para el tema del estático simplemente colocamos lo sig.:

[settings]

queues.conf = odbc,asterisk,ast\_config



Una vez hecho eso guardamos.

Después de guardarlo crearemos la tabla “ast\_config” usando el siguiente script:

create table ast\_config (

id int(8) primary key auto\_increment,

cat\_metric int(8), (Es lo que está entre corchetes, a una menor métrica va en la parte superior)

var\_metric int(8), (Es el valor de la métrica a una menor métrica va en la parte superior)

filename varchar(128), (Nombre del archivo)

category varchar(128), (Nombre de la categoría)

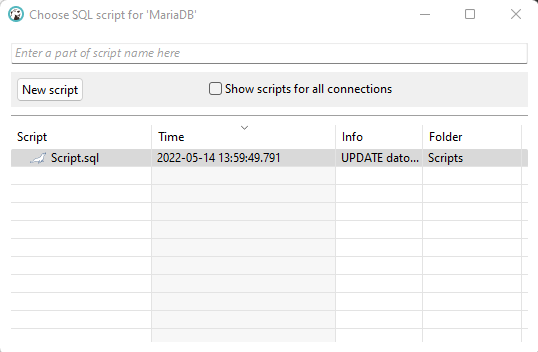
var\_name varchar(128), (Nombre de la variable)

var\_val varchar(128), (Valor de la variable)

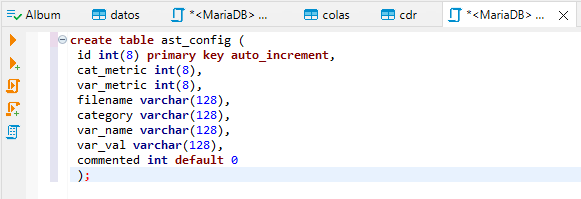
commented int default 0

);

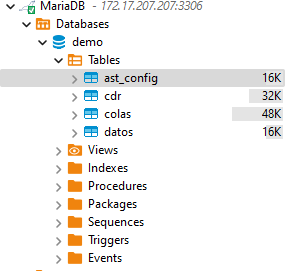
Entramos al dbeaver y en la base de datos le hacemos click derecho, luego click en “Editor SQL” y después en “Script SQL”.



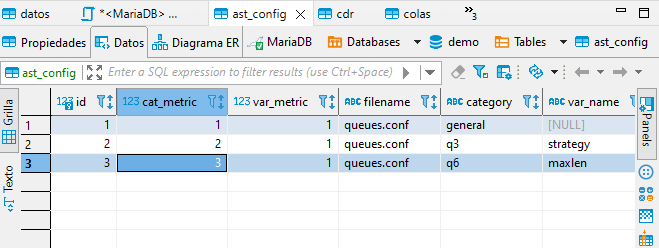
Luego en “New Script” y pegamos el script (no olvidar el USE demo al iniciar).



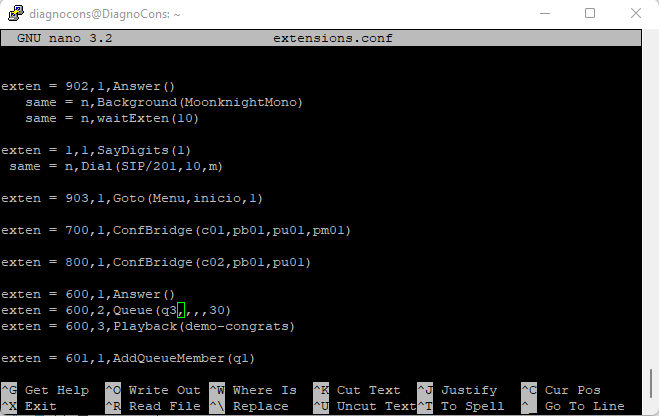
Luego en la base de datos “demo” en “Tables” le damos click derecho y le hacemos click en “Renovar” y nos aparecerá la nueva tabla “ast\_config”.



Una vez hecho eso si queremos agregar o quitar un valor, después de haberlo modificado en la base de datos, hay que hacer un “reload” en la consola Asterisk para que tenga efecto cada vez que hagamos cambios, no será de manera automática.

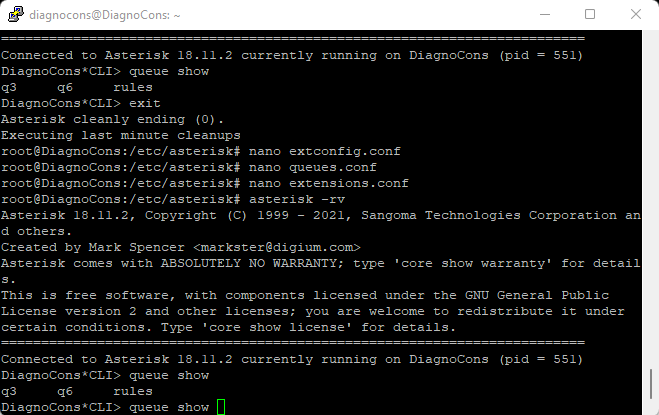


Antes de probar se tiene que hacer la configuración del nuevo queue, ya que si no dará error.



En el Asterisk ya muestra la nueva configuración queue que hicimos en la tabla, prácticamente esta ignorando el archivo “queues.conf” que se encuentra localmente. Para ver esto usaremos el comando:

queue show “TAB”



# Ejemplo de Dynamic Realtime, adicionando usuarios SIP

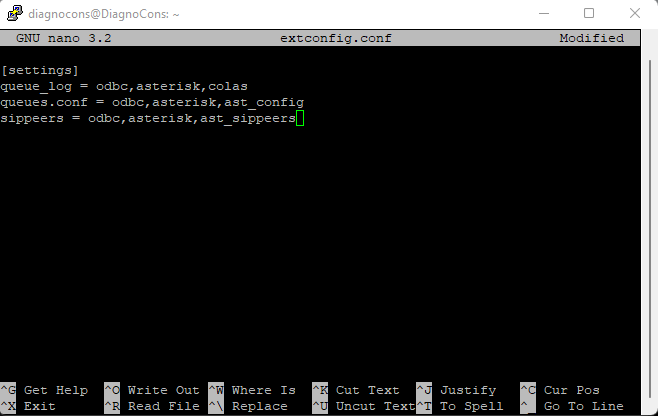
Para adicionar usuarios SIP lo primero que haremos será ingresar al archivo “extconfig.conf” con el sig. comando:

nano extconfig.con

Una vez dentro, para el tema de usuarios SIP en dinámico, escribiremos lo sig.:

[settings]

sippeers = odbc,asterisk,ast\_sippeers



Una vez hecho eso guardamos.

Después de guardarlo crearemos la tabla “ast\_sippeers”, son varios campos, muchos se llenan solos, solamente hay algunos campos que necesitamos llenar que sería:

* type
* name
* defaultuser
* host
* secret
* context

Para crear la tabla usaríamos el siguiente script:

CREATE TABLE ast\_sippeers (

type varchar(6),

name varchar(128),

secret varchar(128),

context varchar(128),

host varchar(128),

ipaddr varchar(128),

port varchar(5),

regseconds bigint(20),

defaultuser varchar(128),

fullcontact varchar(128),

regserver varchar(128),

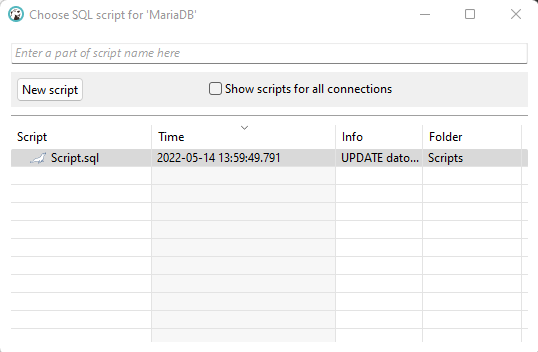
useragent varchar(128),

lastms int(11),

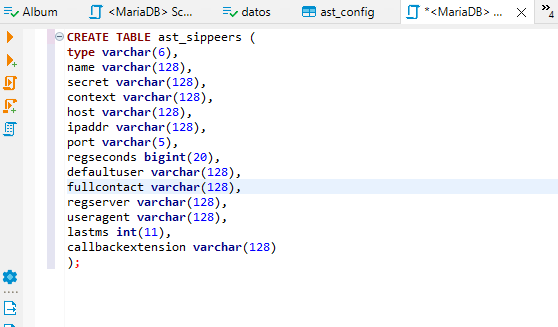
callbackextension varchar(128)

);

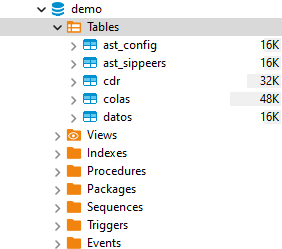
Entramos al dbeaver y en la base de datos le hacemos click derecho, luego click en “Editor SQL” y después en “Script SQL”.



Luego en “New Script” y pegamos el script (no olvidar el USE demo al iniciar).

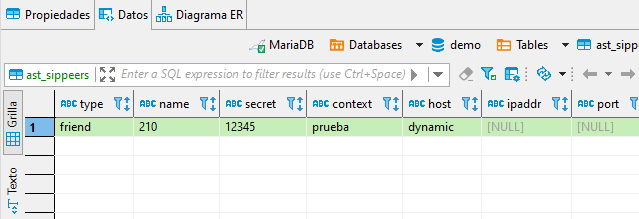


Luego en la base de datos “demo” en “Tables” le damos click derecho y le hacemos click en “Renovar” y nos aparecerá la nueva tabla “ast\_sippeers”.

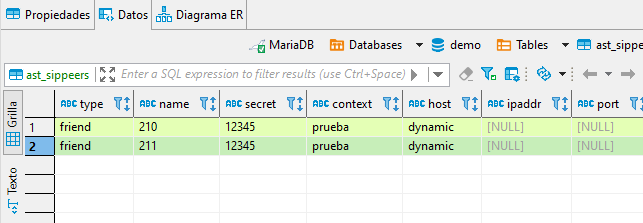


Una vez en la tabla llenamos los datos a modo de prueba que serán los sig.:

* type= friend
* name= 210
* secret= 12345
* context= prueba
* host= dynamic
* defaultuser= 210



Después crearemos el 211, para ello solo duplicamos la fila y cambiamos los campos “name” y “defaultname”.

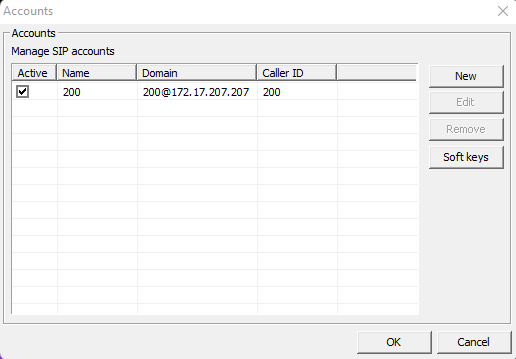


Después salvamos y realizamos un “reload” en la consola de Asterisk.

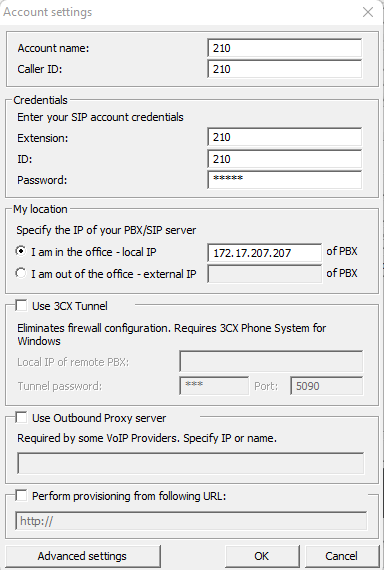
Despues para probar agregaremos el 210 y 211 a lo que seria “3XC PHONE” y “MicroSip” para ver que funciona.

En el caso de “3XC PHONE” haremos lo sig.:

* Entraremos a “accounts” y le daremos en “New”

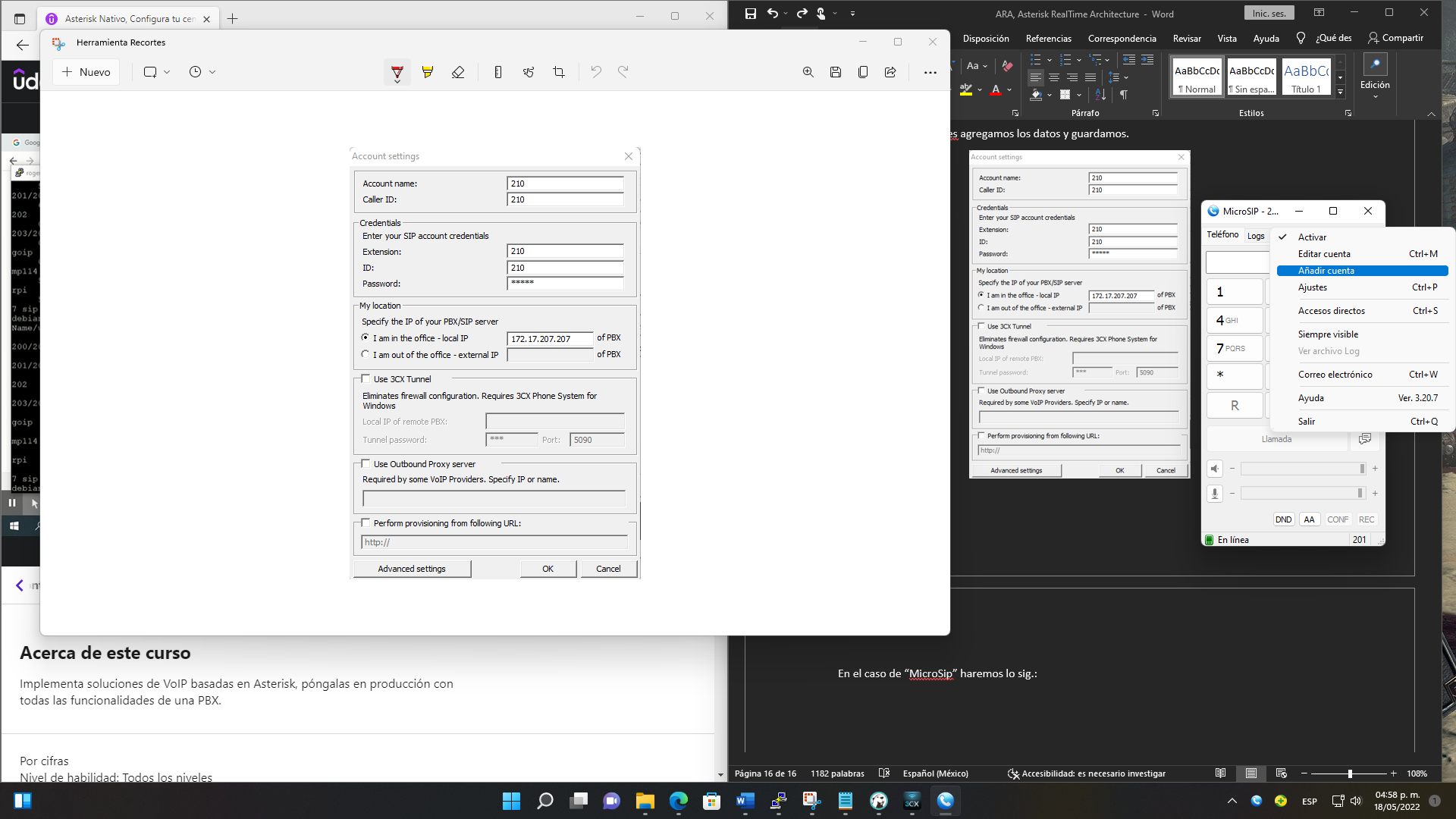


* Despues agregamos los datos y guardamos.

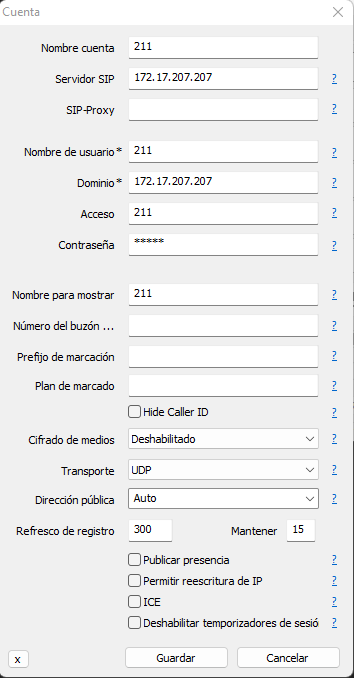


En el caso de “MicroSip” haremos lo sig.:

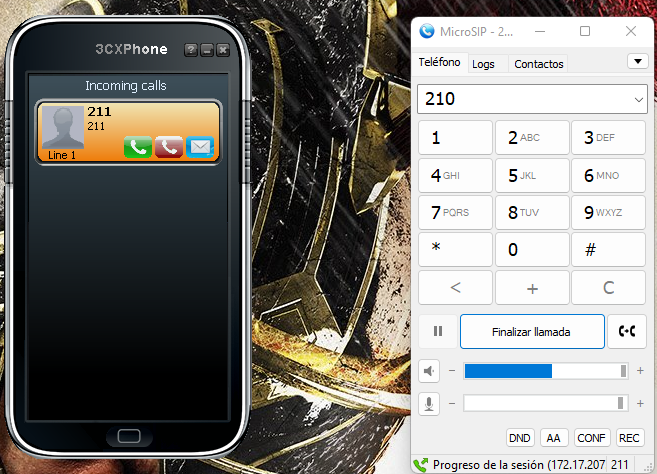
* En el menú le hacemos click en “Añadir cuenta”.

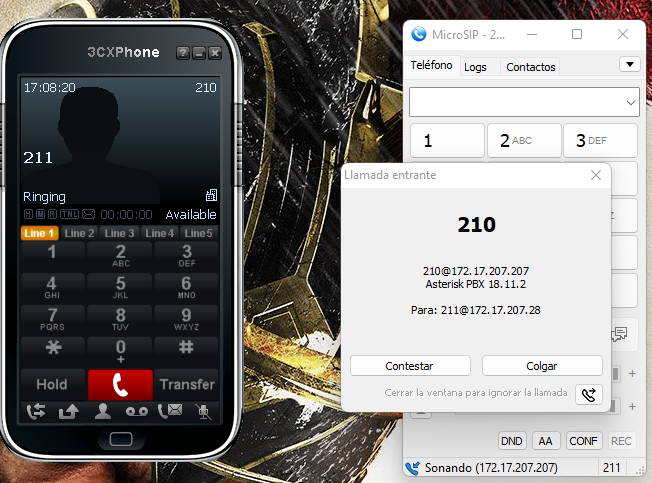


* Después llenamos los datos y guardamos.

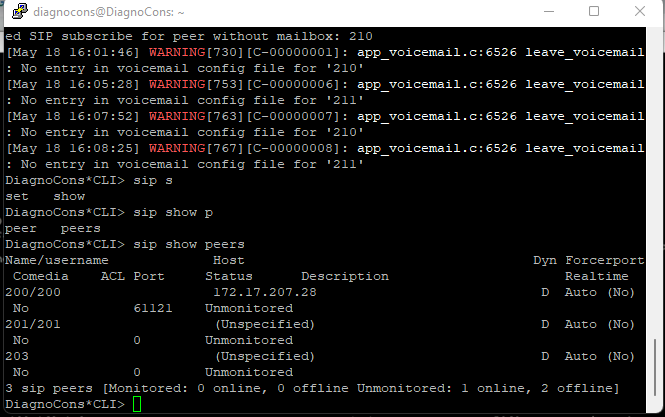


Ya solo sería cuestión de hacer pruebas y ver que funcione.



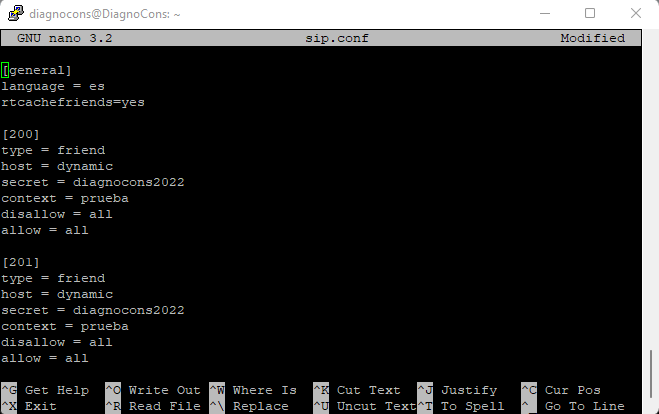


Si vamos a la consola Asterisk y colocamos el comando “sip show peers” no aparecerán los anexos registrados en la base de datos, pero ha funcionado.

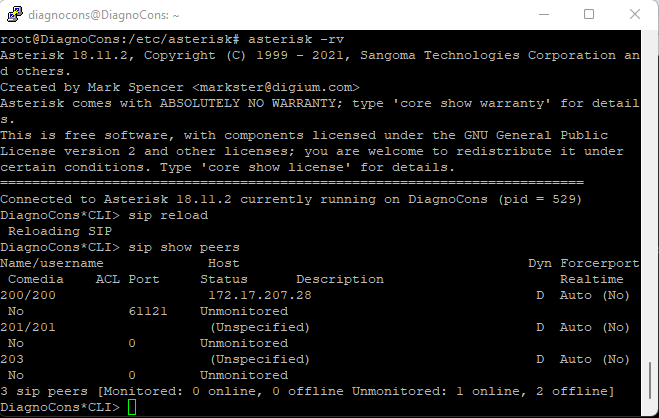


Para cambiar esto iremos al archivo “sip.conf” y en “general” agregaremos lo sig.:

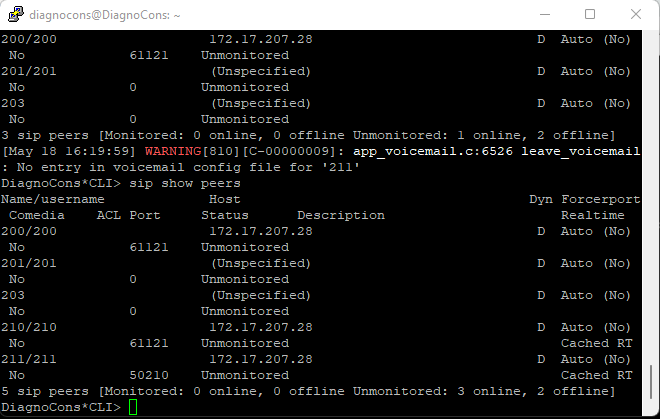
rtcachefriends=yes



Después guardamos, hacemos un “sip reload” en la consola asterisk y volvemos a insertar el comando “sip show peers”.



De inicio no saldrán, pero si realizamos una llamada a los nuevos anexos y de nuevo insertamos el comando “sip show peers” ya saldrán los nuevos anexos.



# Ejemplo de Dynamic Realtime, colocando el Extensions en la BD

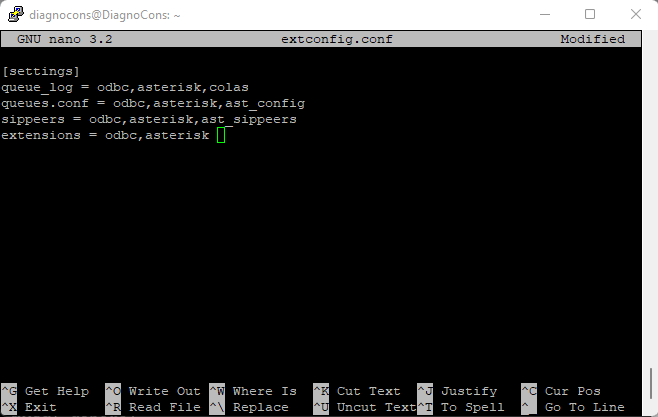
Ahora probaremos el Realtime dinámico para el “extensions.conf” en nuestro plan de marcado. Para ello ingresamos al archivo “extconfig.conf” con el sig. comando:

nano extconfig.conf

Después agregamos lo sig.:

[settings]

extensions = odbc,asterisk (Si no colocamos nombre a la table, asumirá que es el mismo que el de la funcionalidad)



Antes de configurar el extensions.conf agregaremos la tabla llamada “extensions” en la base de datos. Para crear la tabla usaríamos el siguiente script:

CREATE TABLE extensions (

id int(11) NOT NULL auto\_increment,

context varchar(20) NOT NULL default '',

exten varchar(20) NOT NULL default '',

priority tinyint(4) NOT NULL default '0',

app varchar(20) NOT NULL default '',

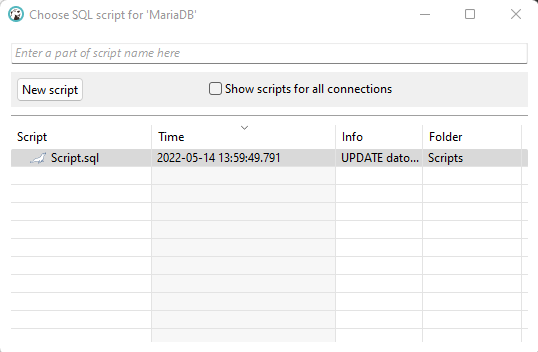
appdata varchar(128) NOT NULL default '',

PRIMARY KEY (context,exten,priority),

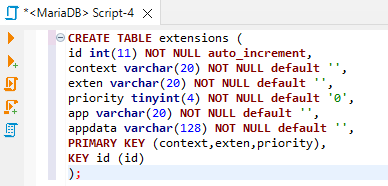
KEY id (id)

);

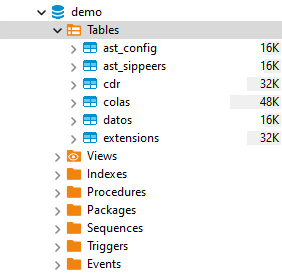
Entramos al dbeaver y en la base de datos le hacemos click derecho, luego click en “Editor SQL” y después en “Script SQL”.



Luego en “New Script” y pegamos el script (no olvidar el USE demo al iniciar).



Luego en la base de datos “demo” en “Tables” le damos click derecho y le hacemos click en “Renovar” y nos aparecerá la nueva tabla “extensions”.



Después seguiría llenar los campos con en el plan de marcado, ejemplo:

* context = pruebart
* exten = 400
* priority = 1
* app = Answer

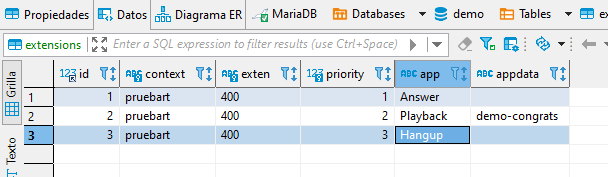
Después en otra fila:

* context = pruebart
* exten = 400
* priority = 2
* app = Answer
* appdata = demo-congrats

Luego sería:

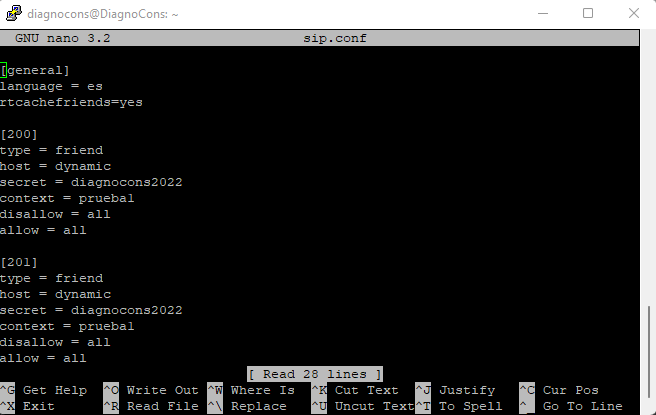
* context = pruebart
* exten = 400
* priority = 3
* app = Hangup

Por último, le daríamos en “save” para que tenga efecto.



Antes de continuar para efectos de la practica haremos unas modificaciones:

* Entramos en el “sip.conf” y a los terminales “200” y “201” el contexto lo cambiaremos a “prueba1”.



Una vez hecho eso en el archivo “extensions.conf” crearemos un contexto llamado “prueba1” de la siguiente manera:

[prueba1]

En el archivo “extensions” para activar el “Realtime” dinámico, lo que tenemos que hacer es lo sig.:

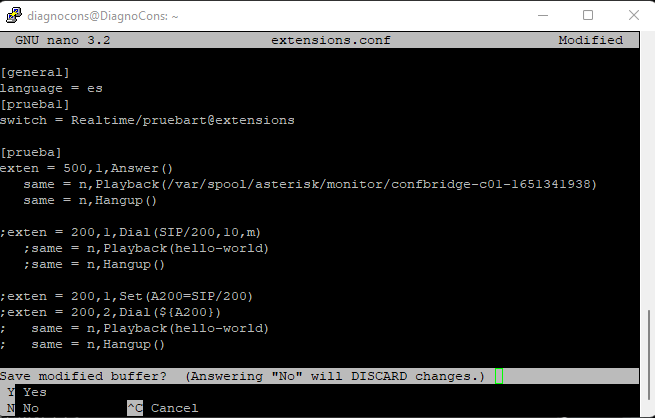
* Vamos al “extensions.conf”, en el contexto que se está trabajando lo sig.:
  + Su sintaxis es la sig.:

switch => Realtime/[[db\_context@]tablename]/<opts>

* + - db\_context@: contexto que hemos puesto en la base de datos.
    - tablename: nombre de la tabla
  + Lo que escribiríamos seria:

[prueba1]

switch = Realtime/pruebart@extensions



Después guardamos y en la consola Asterisk hacemos un “dialplan reload”, “sip reload” y “reload” para que funcionen los cambios y probamos.