# Descripción del driver ODBC.

En base de datos tenemos muchas opciones como lo son “MySQL”, “MariaDB”, “PostgreSQL”, “Microsoft SQL Server” entre otros. Cada base de datos tiene su propio archivo de configuración, lo cual no es muy conveniente, si el cliente quisiera migrar de una base de datos se tendrían que reconfigurar gran parte del Asterisk para hacer las conexiones, lo cual no es muy práctico, para ello existe un conector llamado “ODBC”, en donde Asterisk solamente se conecta al “OBDC” y este se encarga de hacer las conexiones a las diversas bases de datos.

Para comenzar es necesario una base de datos y una vez hecho eso hay que instalar el “ODBC” y una vez instalado necesitamos instalar el driver de la conexión de ODBC hacia la base de datos.

Luego una vez que el driver de la conexión de ODBC hacia la base de datos necesitamos configurar dos archivos propios, ambos se encuentran en los sigs. Directorios:

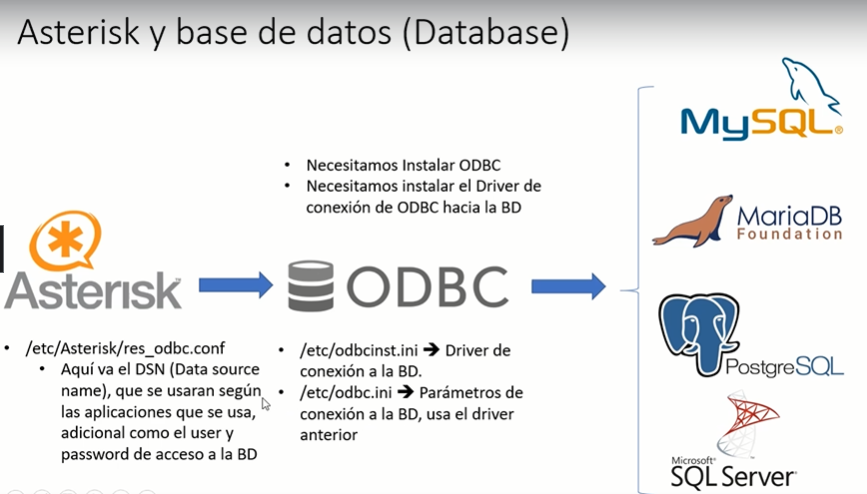
/etc/odbcinst.ini (Driver de conexión a la base de datos).

/etc/odbc.ini (Parámetros de conexión a la base de datos, usa el driver anterior).

Por último, el único archivo que necesitamos para la conexión de Asterisk “ODBC” hacia la base de datos sería el sig.:

/etc/asterisk/res\_odbc.conf

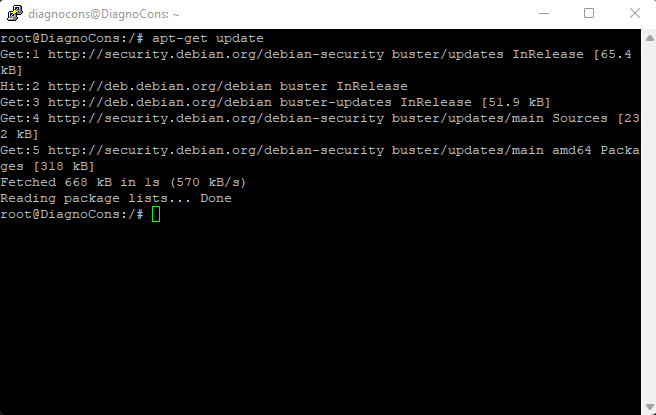
Que será donde se configura la conexión de Asterisk indicándole que va a usar el “ODBC”. Donde se coloca el DSN (Data source name) que se usaran según las aplicaciones que se usa, adicional como el “user” y el “password”.



# Instalación de base de datos.

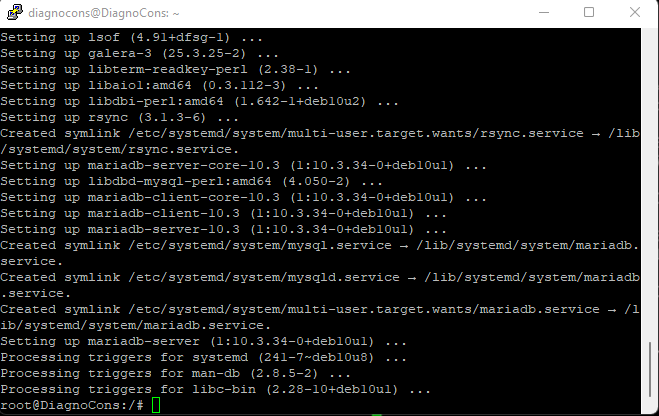
Para esta práctica instalaremos “MariaDB”, lo primero será hacer es actualizar los repositorios con el sig. comando:

apt-get update



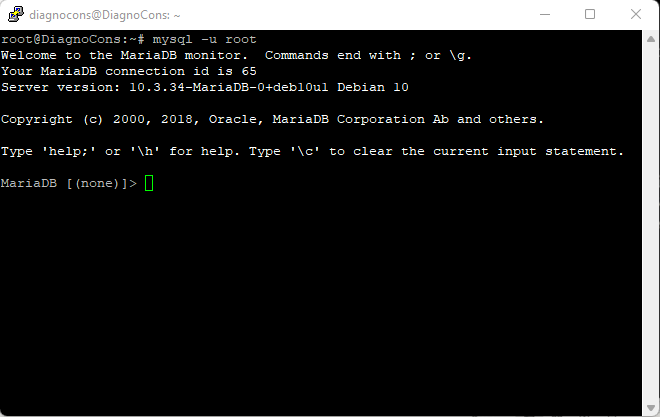
Luego para instalar el “MariaDB” usaremos el sig. comando:

apt-get install mariadb-server (le damos yes)



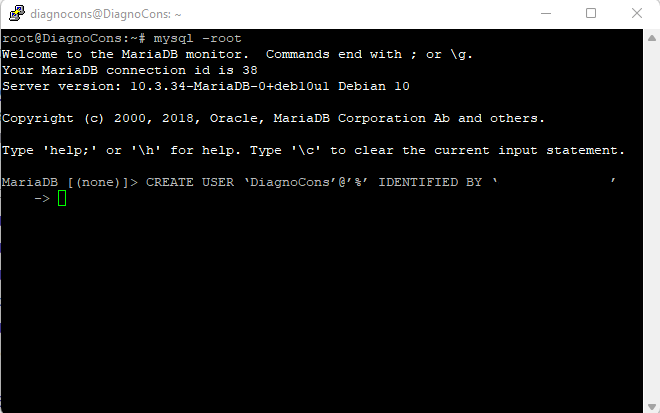
Una vez instalado para entrar a la base de datos usamos el sig. comando:

mysql -u root



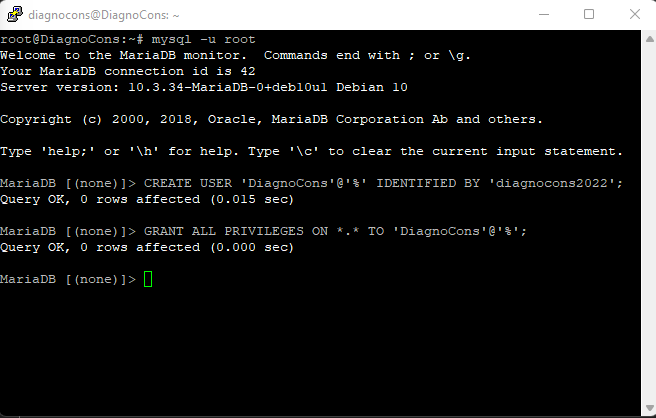
Lo primero será crear un usuario y una contraseña para ello usaremos los sigs. comandos:

CREATE USER 'usuario'@'%' IDENTIFIED BY 'contraseña';



Luego a este usuario se le va a dar todos los privilegios, para ello usamos el sig. comando:

GRANT ALL PRIVILEGES ON \*.\* TO 'DiagnoCons'@'%';



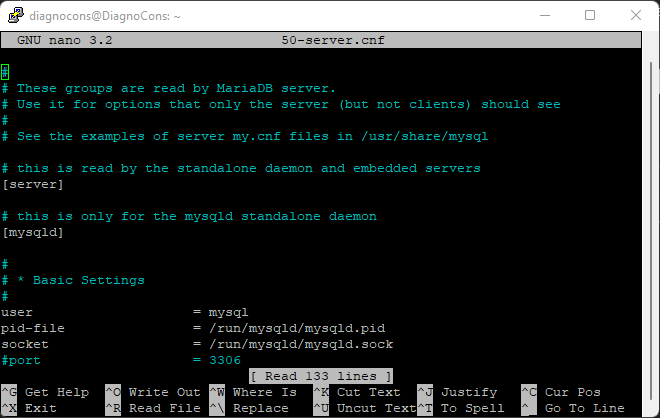
Nota: cuando se le pone (@'%') significa que este usuario puede acceder desde cualquier lugar.

Con este usuario se puede trabajar desde consola, pero no es muy practico trabajar desde consola, para esto usaremos un editor el cual podremos usar desde la PC (Windows) para poder interactuar con la base de datos. Para poder operar desde Windows hay que configurar un archivo que está en el sig. directorio:

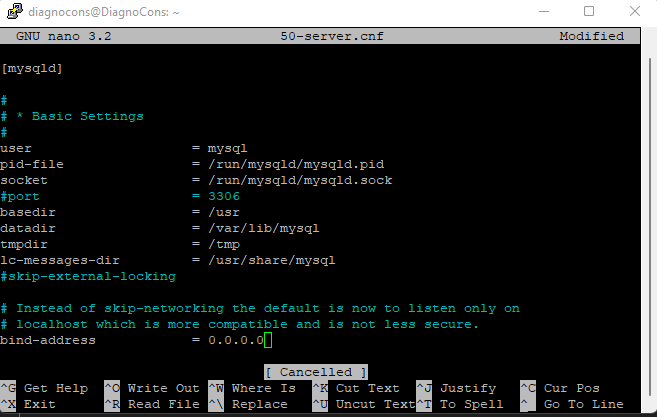
cd /etc/mysql/mariadb.conf.d/

Una vez en el directorio abrimos el archivo llamado “50-server.cnf” con el sig. comando:

nano 50-server.cnf



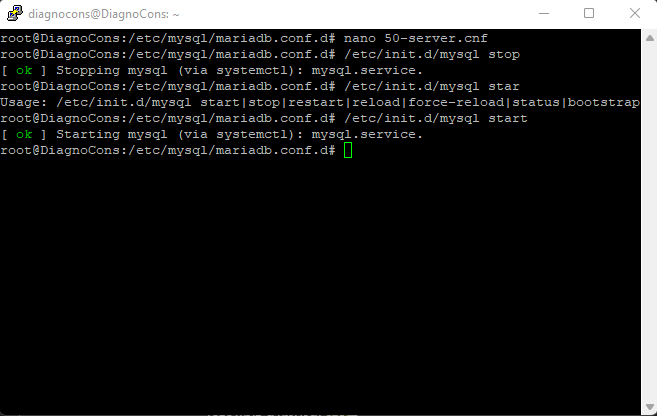
Aquí buscaremos algo llamado “bind-address”, cuando se le pone 127.0.0.1 solo se puede acceder desde la misma PC(Servidor), pero si le cambiamos a 0.0.0.0 se le puede acceder desde cualquier ubicación o un determinado rango de red, por efecto de la práctica lo dejaremos en que pueda acceder desde cualquier lugar.



Ahora tocaría reiniciar el servidor para ello usaremos los sigs. comandos:

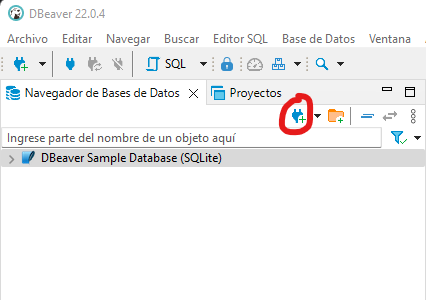
/etc/init.d/mysql stop

/etc/init.d/mysql start

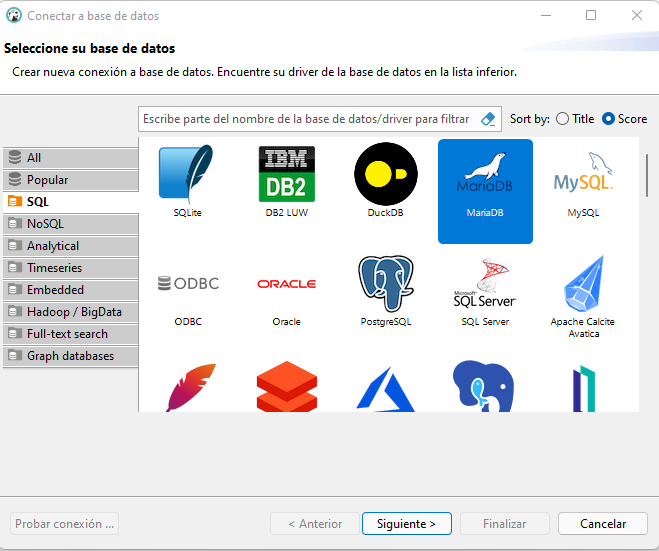


Una vez reiniciado el servicio de MySQL vamos a usar un gestor de base de datos, para ello descargaremos en la PC (Windows) un programa llamado dbeaver.

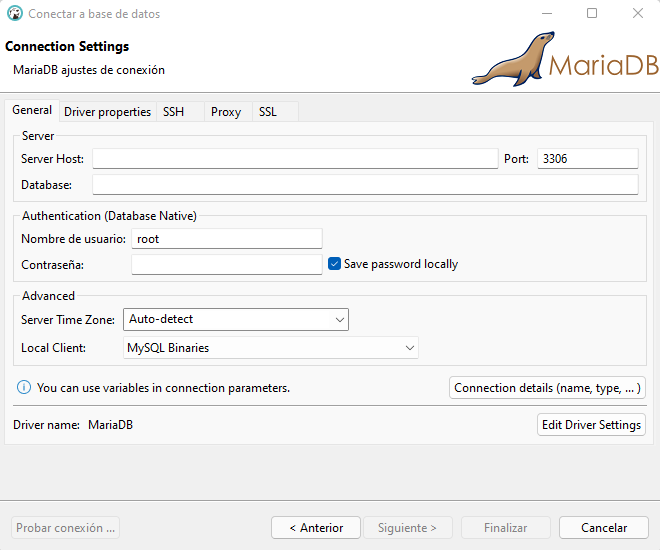
Una vez instalado el dbeaver abrimos el programa y vamos a establecer una nueva conexión, para ello le hacemos click en “nueva conexión”.



Después le hacemos click en el apartado “SQL” y seleccionamos “MariaDB” y le damos “siguiente”.



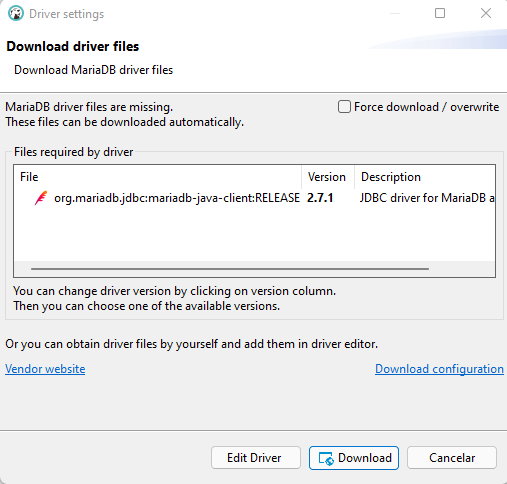
Después en la base da datos colocamos lo sig:



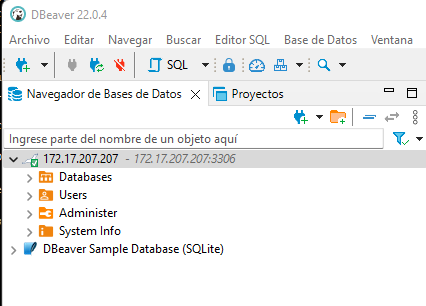
* Server Host: la IP del servidor.
* Username: usuario del servidor.
* Password: la contraseña.

Después solamente le damos en “finalizar”.

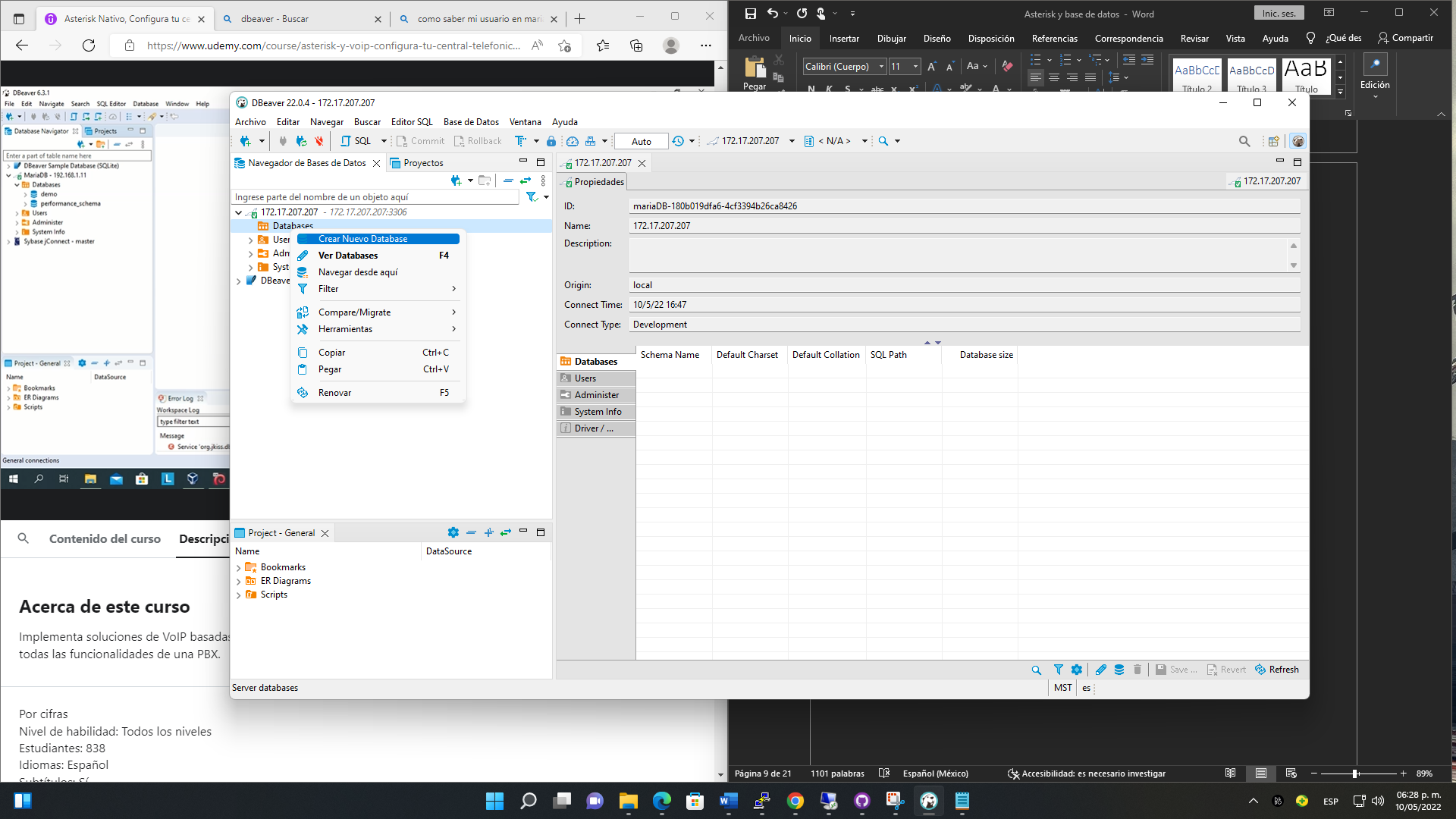
Luego saldrá una ventana llamada “Download driver files” y le hacemos click en “Download”.



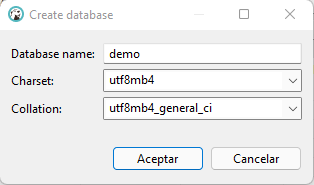
Listo con eso habremos establecido la conexión con la base de datos del servidor.



Después crearemos una base de datos llamada “demo”, para hacerlo en “Databases” le daremos click derecho y luego “Crear nuevo Database”.

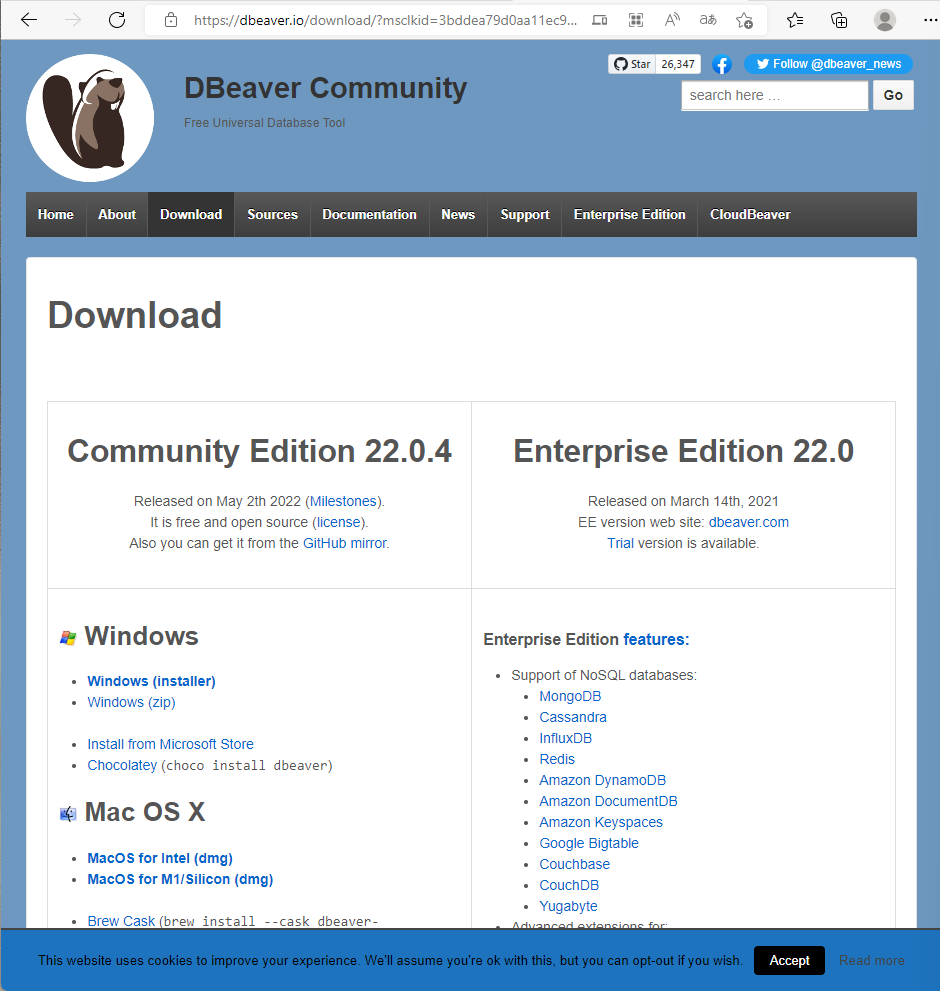


Le asignamos el nombre y le damos en “aceptar”.

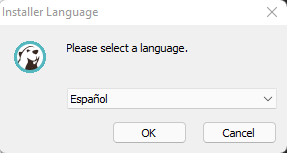


# Como instalar dbeaver.

Lo primero será ir al enlace (<https://dbeaver.io/download/?msclkid=3bddea79d0aa11ec94302426390ee7f6>) donde descargaremos el instalador.



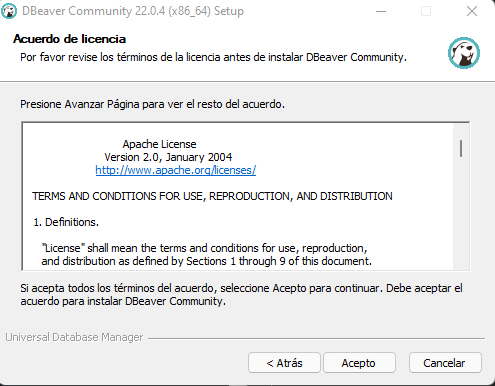
Después seleccionamos el lenguaje.



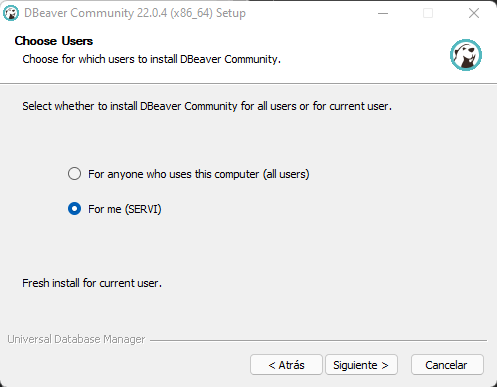
Después le damos “siguiente”.



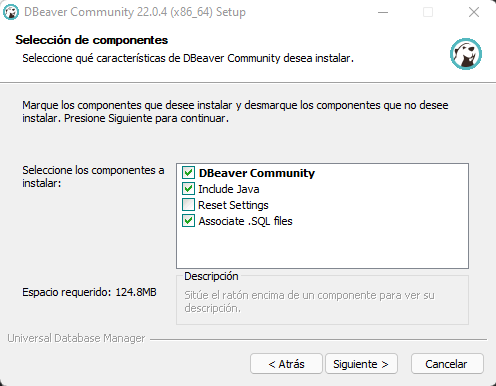
Luego aceptamos los términos de licencia.



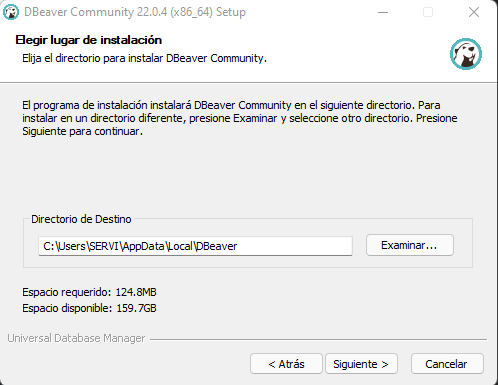
En la sig. opción le dejamos “for me” y le damos “siguiente”.



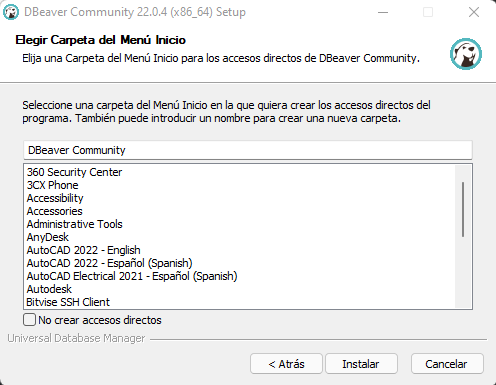
En seleccionar componentes la agregamos la opción “Associate .SQL files”.



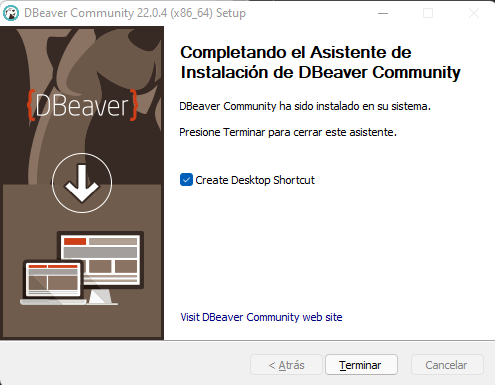
Seleccionamos la ruta donde lo vamos a instalar y le damos “siguiente”.



Por último, le damos a “instalar” y esperamos.



Listo con eso ya tendremos instalador dbeaver, ya solo le damos en “finalizar”.



# Instalación de ODBC Conector para MariaDB.

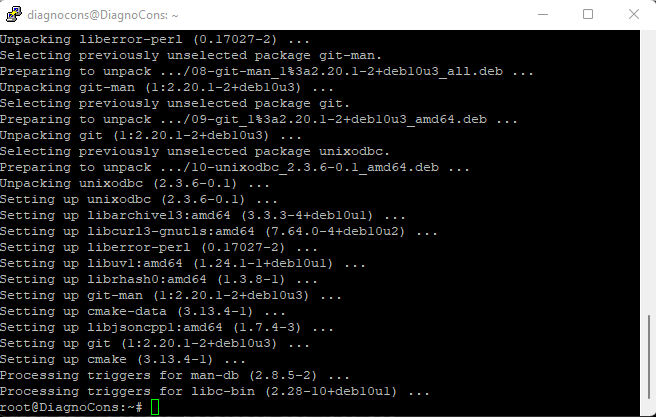
Lo primero que tenemos que hacer es instalar el ODBC y luego instalar el conector. El driver cambia de nombre de nombre dependiendo la distribución y dependiendo de la base de datos que conecta, en este caso que estamos usando Debian y MariaDB se llamara “libmaodbc.so”.

## Instalación de ODBC y paquetes previos.

Lo primero será descargar el primero paquete con este comando:

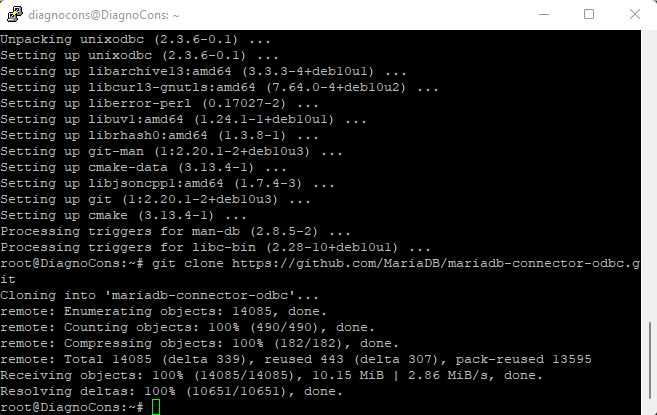
apt-get install unixodbc git cmake

* El git lo usaremos para descargar.
* EL cmake para instalar.



Después nos dirigimos al directorio “/usr/src/” después desargamos el paquete de git con el sig. comando:

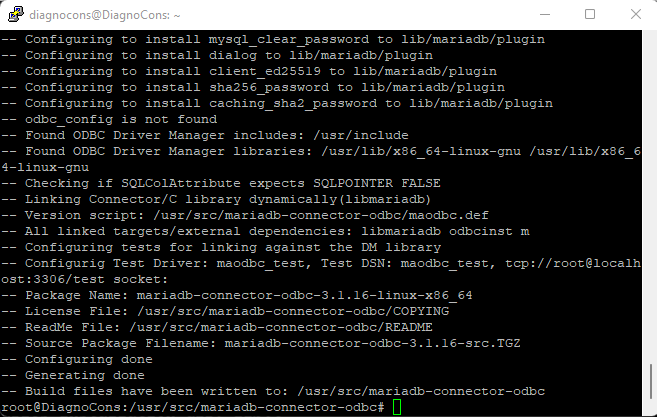
git clone https://github.com/MariaDB/mariadb-connector-odbc.git



Listo una vez descargado el git, entramos a la carpeta “mariadb-connector-odbc” y hacemos la compilación para lo cual insertaremos los sigs. comandos:

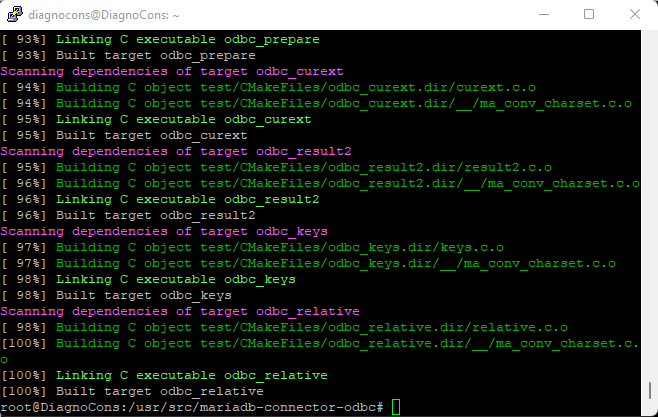
cd mariadb-connector-odbc

cmake -DCMAKE\_BUILD\_TYPE=RelWithDebInfo -DCONC\_WITH\_UNIT\_TESTS=Off -DCONC\_WITH\_MSI=OFF -DCMAKE\_INSTALL\_PREFIX=/usr/local .



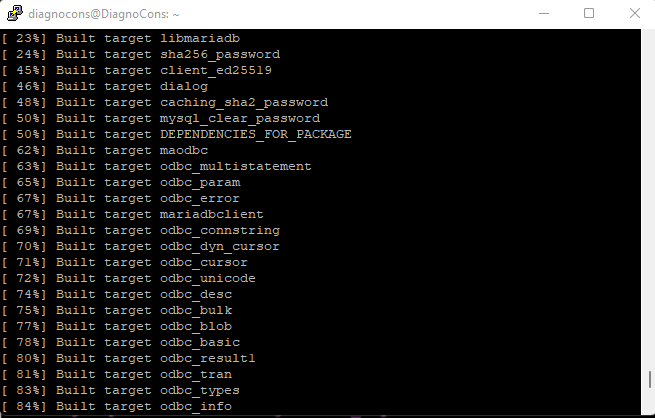
Luego insertamos el sig. comando:

cmake --build . --config RelWithDebInfo

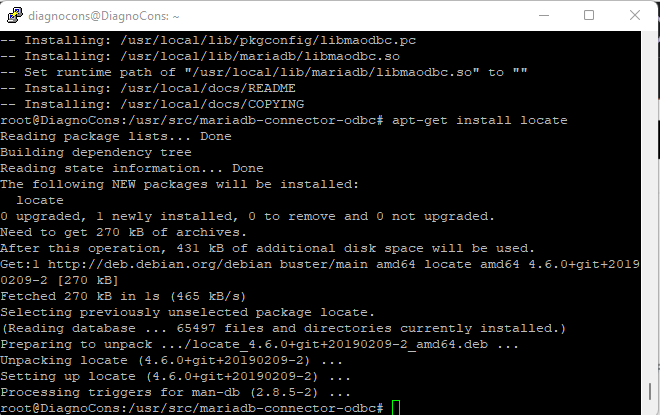


Por último, insertamos el sig. comando:

make install



Linux posee una herramienta llamada “locate” que nos sirve para buscar archivos, esta la usaremos para ubicar el archivo “libmaodbc.so”, para instalarla usamos el sig. comando:

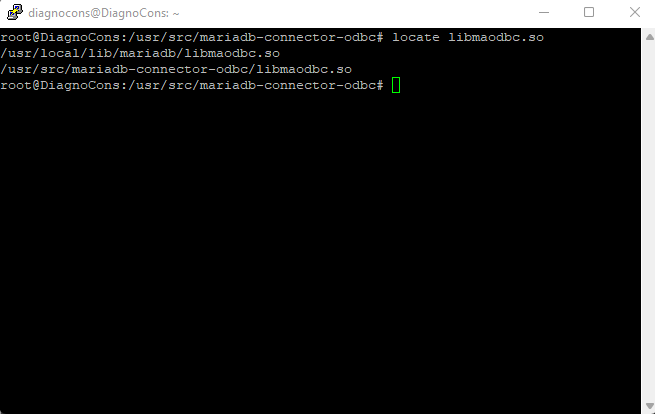


Una vez instalado actualizamos su base de datos para ello usaremos el sig. comando:

updatedb

Luego para buscar el archivo usamos el sig. comando

locate libmaodbc.so



Aparecen dos opciones, pero la que nos interesa es la primera que es donde esta ubicado, el otro es de donde se ha descargado.

/usr/local/lib/mariadb/libmaodbc.so

# Configuración de odbcinst.ini, odbc.ini y res\_odbc.conf.

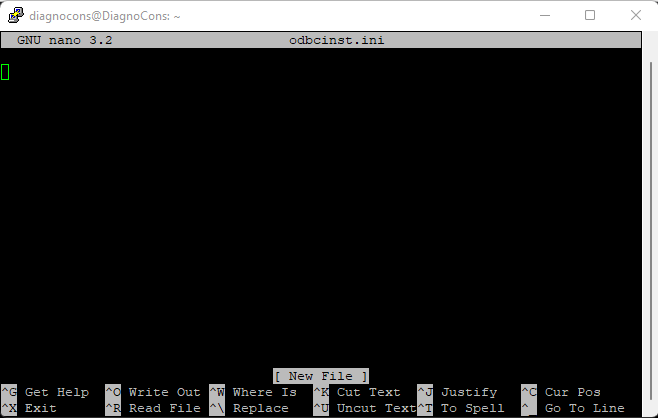
Recordemos:

* odbcinst.ini: driver de la conexión a la BD.
* odbc.ini: parámetros de conexión a al BD y usa el anterior.

## odbcinst.ini.

Lo primero será ubicarnos en el directorio “/etc/”, ahora el archivo “odbcinst.ini” no existe como tal, por lo cual tendremos que crearlo, para ello usaremos el sig. comando:

nano odbcinst.ini

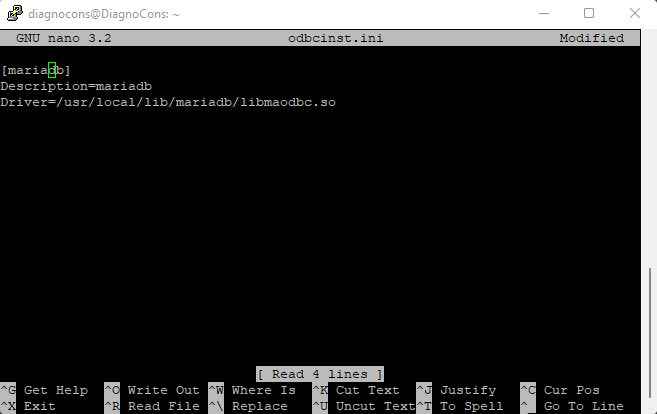


Lo primero será crear el identificador de nuestro conector, para lo cual colocaremos lo sig.:

[mariadb] (ID con el que lo vamos a identificar)

Description=mariadb (es opcional la descripción)

Driver=/usr/local/lib/mariadb/libmaodbc.so (Aquí colocamos donde se encuentra ubicado el driver)



## odbc.ini.

Después crearemos el “odbc.ini” donde vendrían los parámetros de conexión, para crearlo usaremos el sig. comando:

nano odbc.ini



Después le agregaremos lo sig.:

[mariadbcliente] (Su ID)

Description=mariadbcliente

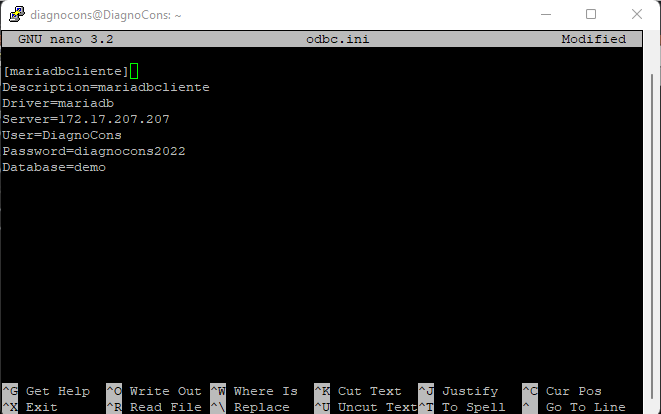
Driver=mariadb (Se coloca el ID que le colocamos al ID de “odbcinst.ini”)

Server=172.17.207.207 (Se coloca la IP del servidor)

User=DiagnoCons (El usuario que creamos para la base de datos)

Password=diagnocons2022 (La contraseña para la base de datos)

Database=demo (Nombre de la base de datos que creamos en dbeaver)

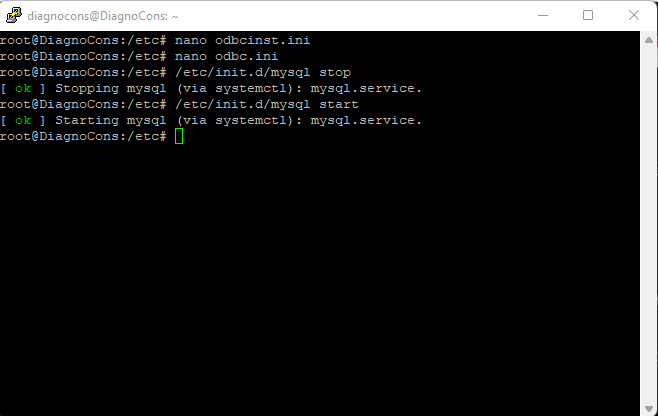


Una vez hecho todo esto salvamos.

Después reiniciamos la base de datos, para ello usamos los sigs. comandos:

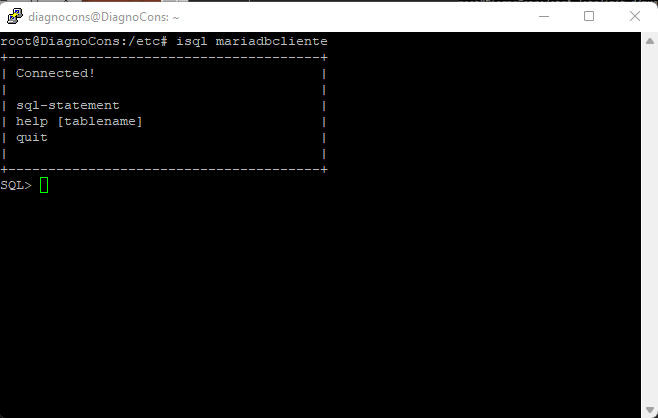
/etc/init.d/mysql stop

/etc/init.d/mysql start



Para saber si hay conexión usamos el sig. comando:

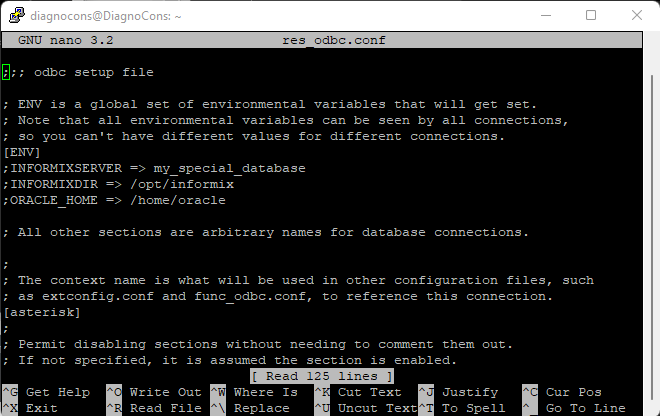
isql mariadbcliente



Nota: si esta correcta la información mostrara este cuadro.

Para que haya conexión del Asterisk hacia la base de datos hay que configurar el archivo “res\_odbc.conf” pero siempre manteniendo el orden, para ingresar a el nos ubicaremos el el directorio “/etc/asterisk” y después usaremos el sig. comando:

nano res\_odbc.conf



Como siempre lo moveremos a la carpeta bak usando el sig. comando:

mv res\_odbc.conf res\_odbc.conf.bak

Después volvemos a entrar a “res\_odbc.conf” y le vamos a crear el DSN.

[asterisk] (Le creamos su ID)

enable=yes

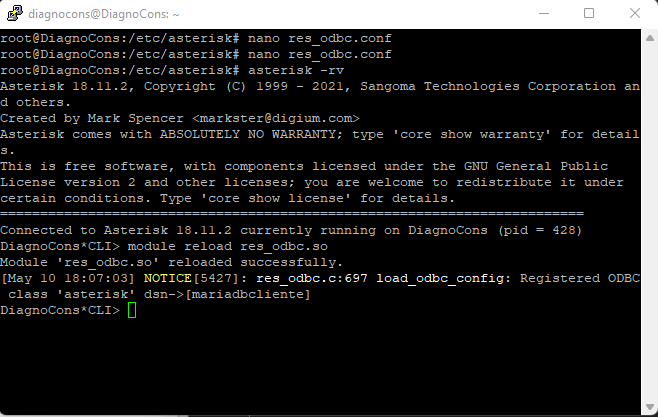
dsn=mariadbcliente (ID del “odbc.ini”)

pre-connect=yes (Para que se establezca la conexión)



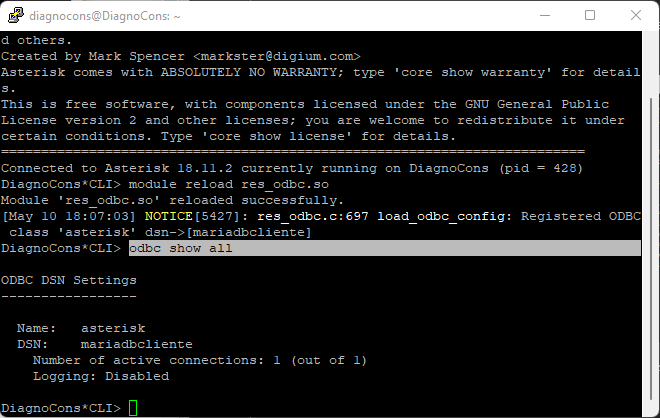
Después entramos en la consola Asterisk y recargamos el modulo con el sig. comando:

module reload res\_odbc.so



Para saber si tiene conexión la consola usamos el sig. comando:

odbc show all



Nota: nos mostrara un mensaje de que la conexión está hecha.

Con esto el Asterisk está listo para conectar con la base de datos.

# Inserción del CDR en la base de datos usando cdr\_adatative.

Una vez hecha la conexión de Asterisk a la base de datos mediante el ODBC, lo que seguiría seria poner una aplicación dentro de la base de datos, en este caso usaremos el CDR. Recordemos el CDR se encuentra en el directorio “/var/log/asterisk/cdr-csv” y después abríamos el archivo “Master.csv” y en este archivo se guardan el registro de llamadas que se efectuaba en la central telefónica, con esto haremos que los eventos se almacenen ya directamente en la base de datos, para ello necesitamos una tabla en la base de datos donde estén todos estos elementos, para ello usaremos el sig. script.

USE demo;

CREATE TABLE cdr (

id bigint(20) NOT NULL auto\_increment,

calldate datetime NOT NULL default '0000-00-00 00:00:00',

clid varchar(80) NOT NULL default '',

src varchar(80) NOT NULL default '',

dst varchar(80) NOT NULL default '',

dcontext varchar(80) NOT NULL default '',

channel varchar(80) NOT NULL default '',

dstchannel varchar(80) NOT NULL default '',

lastapp varchar(80) NOT NULL default '',

lastdata varchar(80) NOT NULL default '',

duration int(11) NOT NULL default '0',

billsec int(11) NOT NULL default '0',

disposition varchar(45) NOT NULL default '',

amaflags int(11) NOT NULL default '0',

accountcode varchar(20) NOT NULL default '',

peeraccount varchar(20) NOT NULL default '',

uniqueid varchar(32) NOT NULL default '',

linkedid varchar(80) NOT NULL default '',

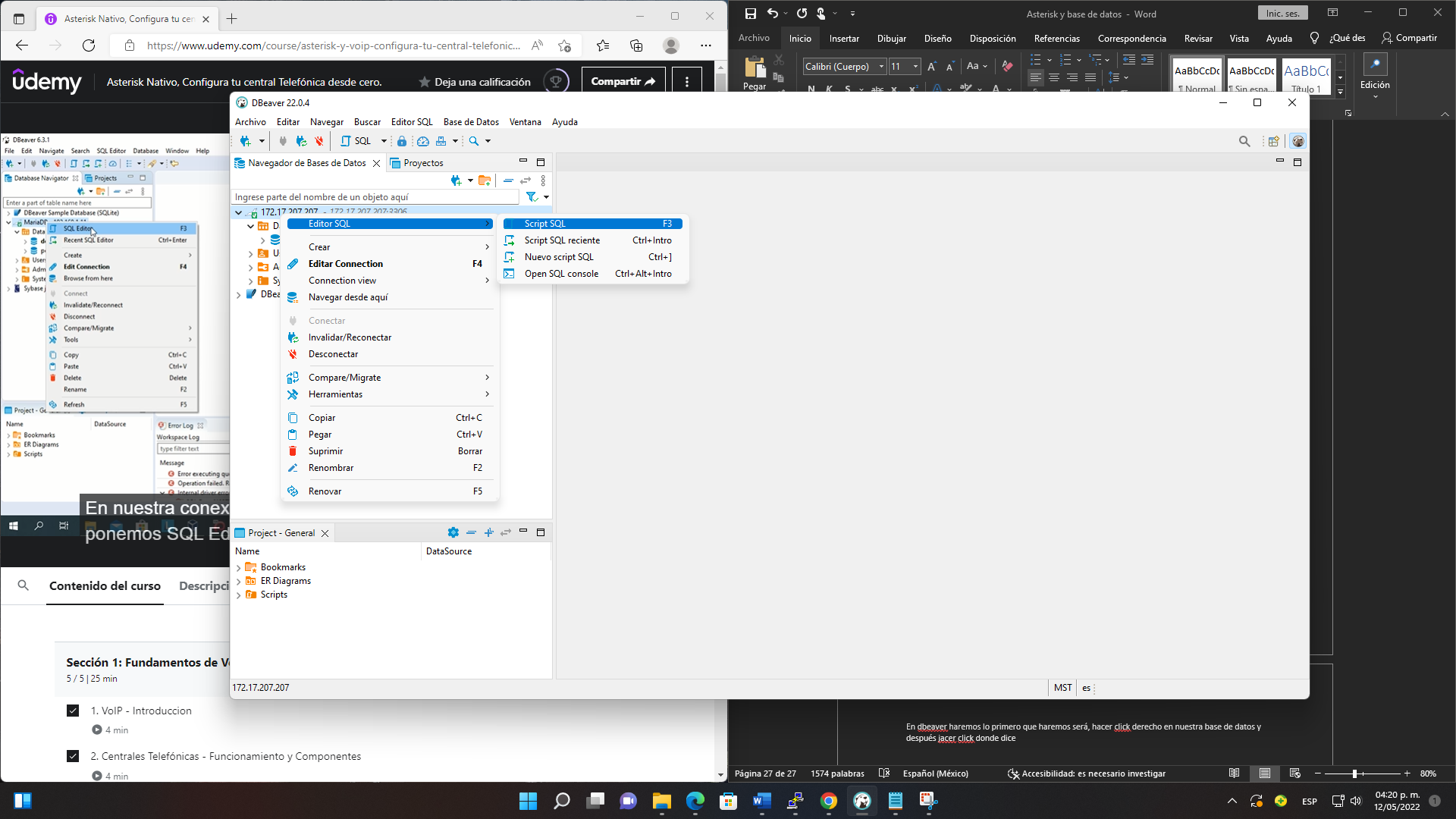
userfield varchar(255) NOT NULL default '',

PRIMARY KEY (`id`),

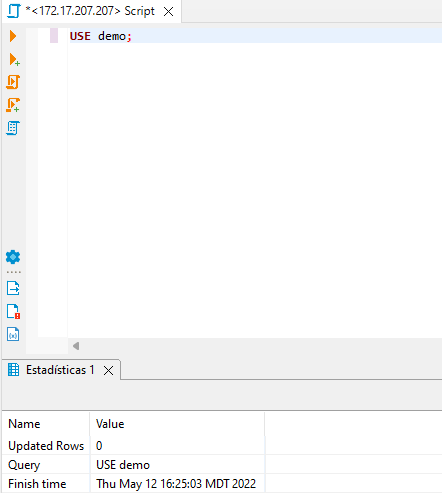
KEY callerid (clid)

);

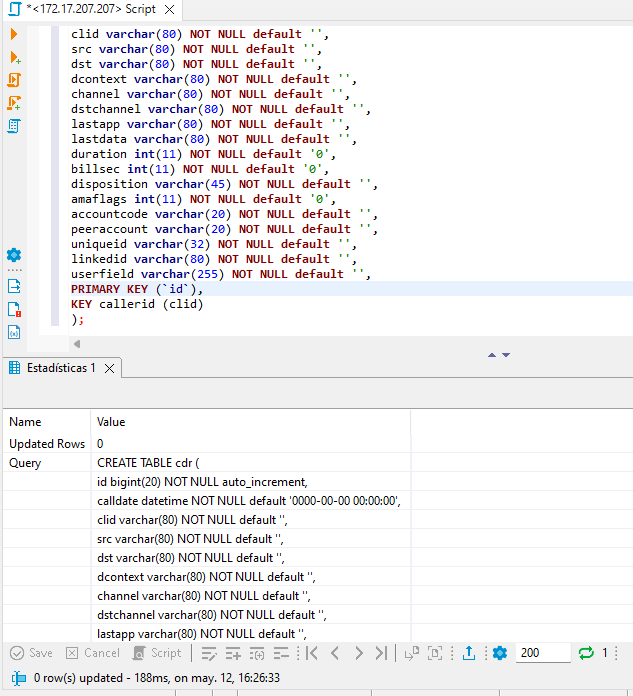
En dbeaver haremos lo primero que haremos será, hacer click derecho en nuestra base de datos y después hacer click donde dice “Editor SQL” y luego “Script SQL”



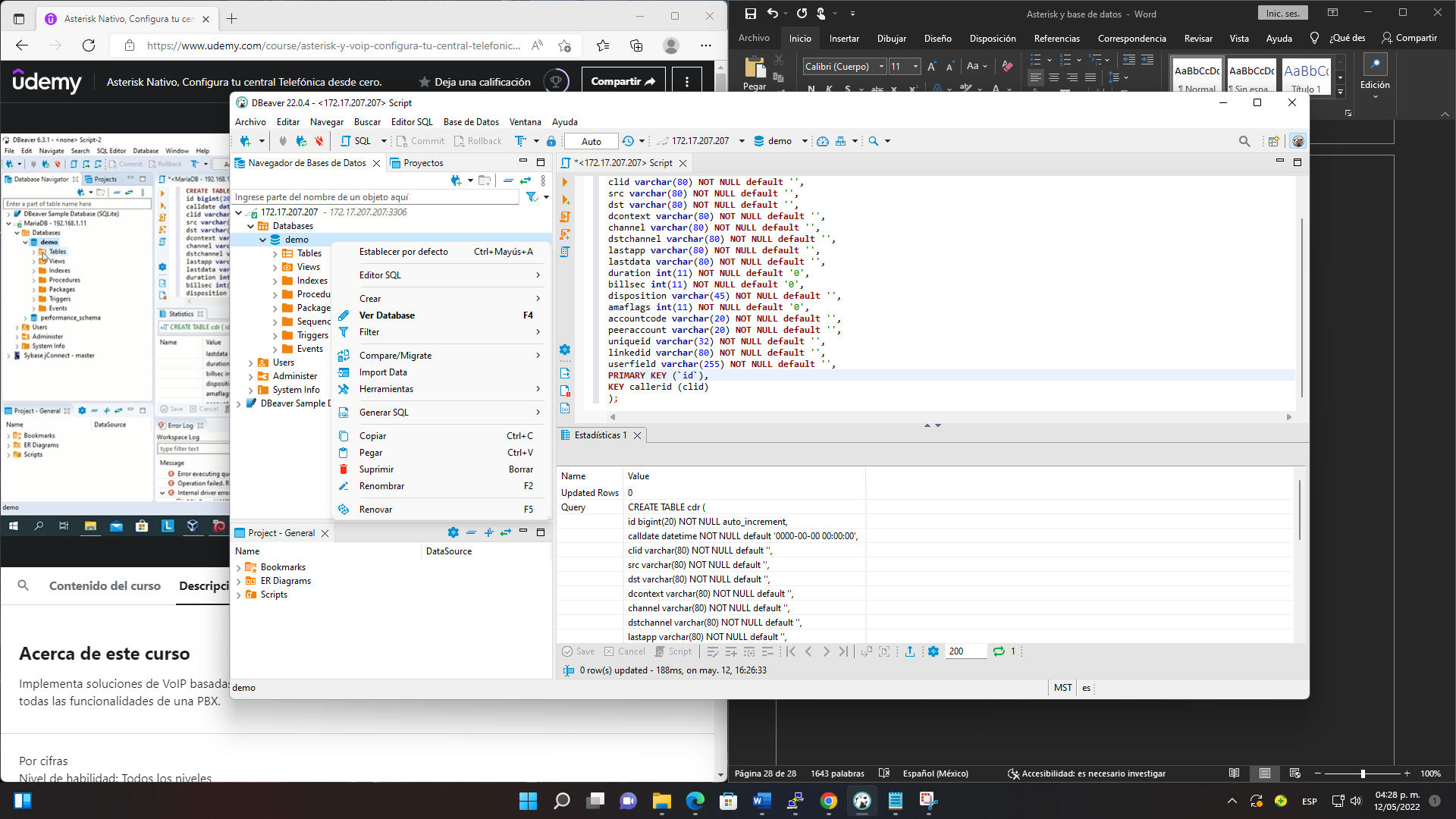
Se abrirá el editor y pegaremos el primer apartado que seria “USE demo;” que le indicaría que use la base de datos “demo”, después lo corremos con el botón de “Ejecutar comando SQL”.

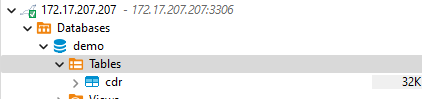


Luego ejecutamos el resto del script.

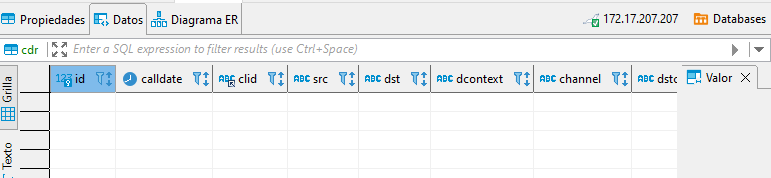


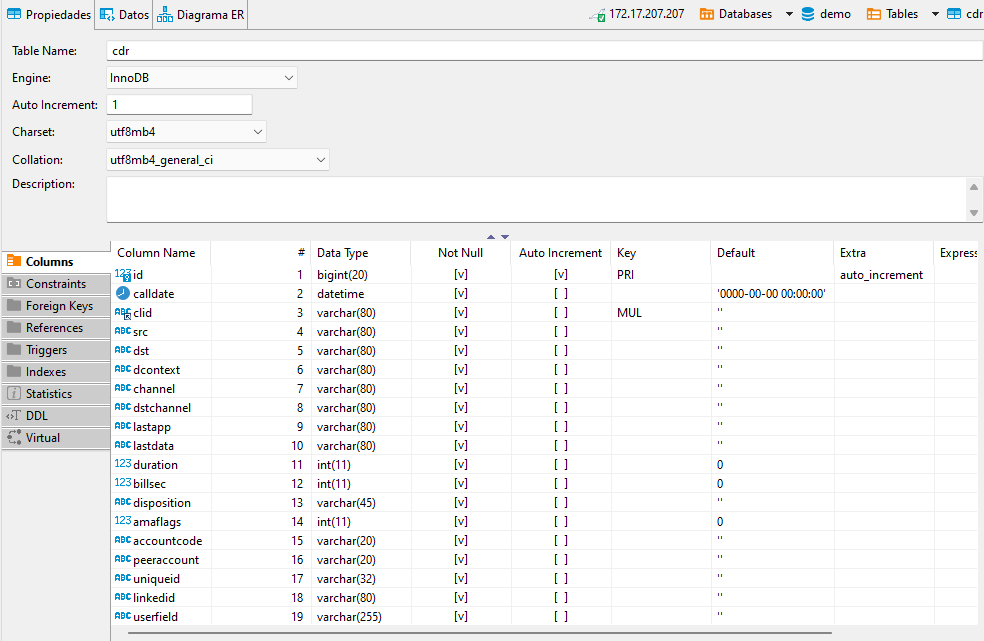
Con esto ya abremos creado nuestra tabla CDR, después en la base de datos demo le damos en “Renovar” y nos aparecerá la tabla.





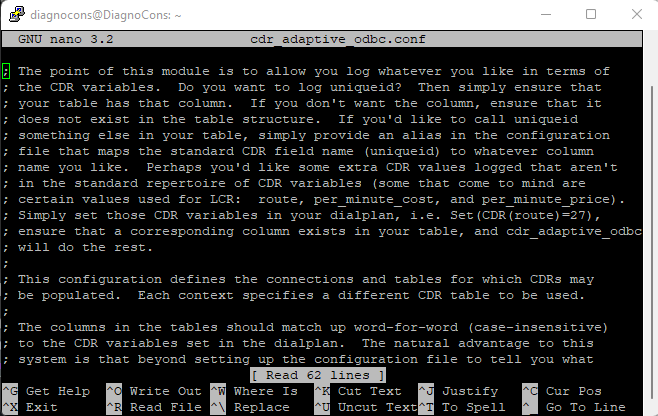
Si le hacemos click a la tabla CDR y nos vamos a “Datos” veremos que esta vacía y si vamos a “Propiedades” veremos los campos de la tabla.





Una vez creado la tabla vamos al directorio “/etc/asterisk/”, recordemos ya la conexión está hecha y dicha conexión la tenemos identificada como “asterisk” en el archivo “res\_odbc.conf”, ahora cada aplicación va a usar esta conexión “asterisk”, para grabar el CDR en la base de datos el archivo que usaremos será el “cdr\_adatative\_odbc.conf”, para ingresar al archivo usaremos el sig. comando:

nano cdr\_adaptative\_odbc.conf

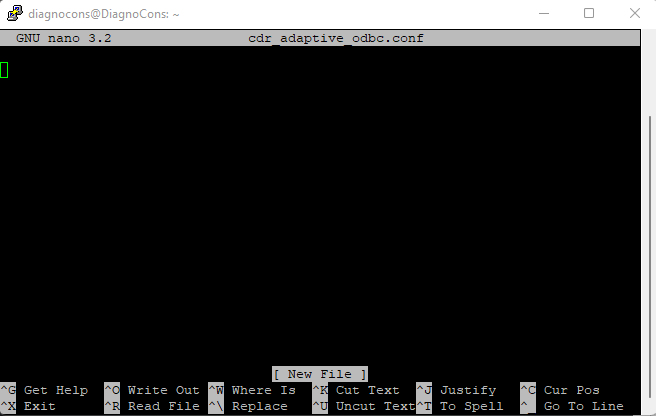


Después como siempre lo moveremos a un bak con el sig. comando:

mv cdr\_adaptative\_odbc.conf cdr\_adaptative\_odbc.conf.bak

Luego un nuevo archivo “cdr\_adaptive\_odbc.conf” con el sig. comando:

nano cdr\_adaptive\_odbc.conf



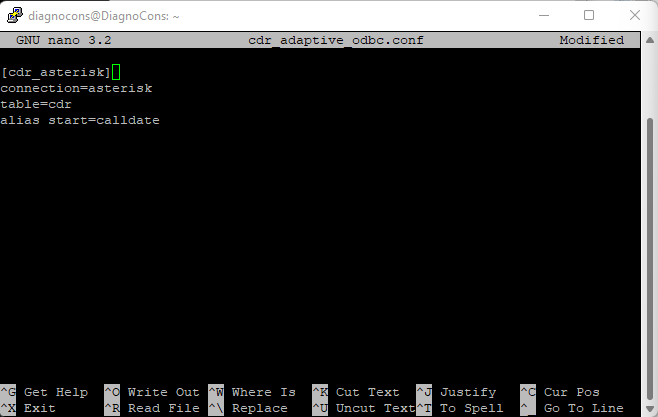
Cada aplicación se modifica su archivo o su configuración propia. Después escribiremos lo sig.:

[cdr\_asterisk] (Nombre de la conexión)

connection=asterisk (Conexión que vamos a usar)odcll

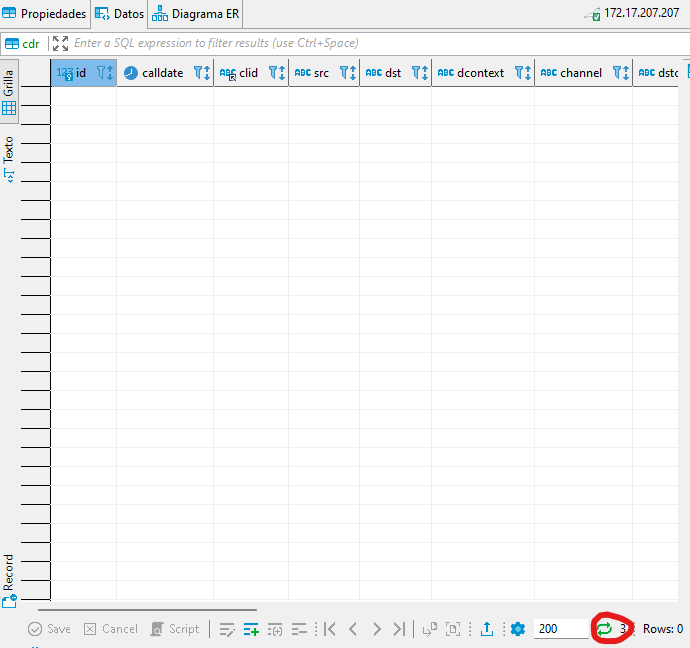
table=cdr (Tabla a la que nos vamos a conectar)

alias start=calldate (Primer campo a usar)

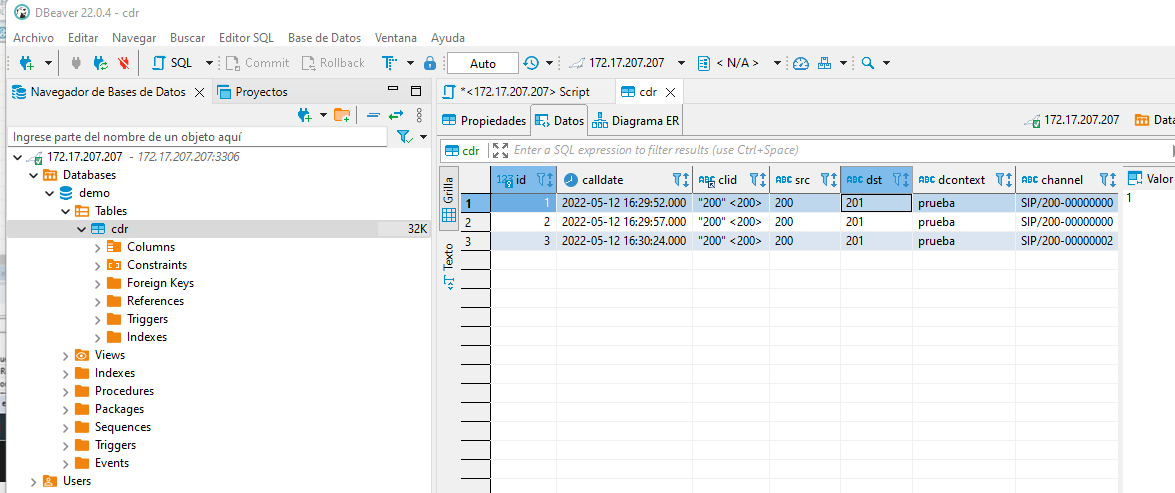


Después guardamos, entramos a la consola Asterisk y usamos un “reload” y hacemos llamadas de prueba.

Luego vamos al dbeaver refrescamos con el botón verde y veremos que tiene 3 elementos en este caso.



Después cerramos y volvemos a abrir la tabla y ya aparecerán los datos.



Entre los datos tenemos:

* calldate: fecha y hora.
* src: el origen.
* dst: el destino.
* dcontext: el contexto.
* channel y dstchannel: los canales.
* lastapp: aplicación usada.
* disposition: si la llamada fue contestada o no.
* duration: duración de la llamada.
* uniqueid: ID para identificar la llamada.

# Descripción de func\_odbc.conf para ejecutar Querys en el Dialplan.

Ahora veremos cómo le hacemos interactuar al dialplan con la base de datos. Lo que haremos para interactuar seria leer valores, insertar valores o actualizar valores en l a base de datos. Recordemos el archivo del dialplan seria “extensions.conf”. Para interactuar el dial plan con la base de datos usaremos el archivo “func\_odbc.conf” ubicado en el directorio “/etc/asterisk/”.

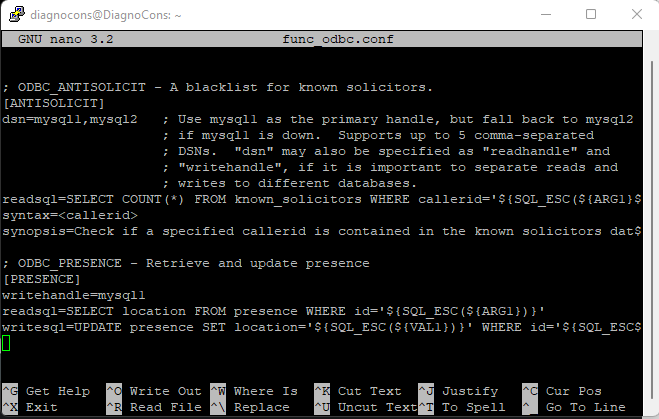
Para verlo usaremos el sig. comando:

nano func\_odbc.conf



En la parte final vemos que van los querys, los cuales son 3 principalmente:

* INSER: cuando queremos insertar registros en la base de datos.
* SELECT: cuando queremos obtener datos de la base de datos.
* UPDATE: cuando queremos actualizar los datos de la base de datos.



Primero vamos a conocer las sintaxis del archivo:

nombreFuncion

prefix=PRE (El prefijo es opcional, por defecto el prefijo es ODBC)

dsn=asterisk (Conexión de Asterisk al ODBC)

readsql= SELECT

writesql= INSERT,UPDATE

Cuando hagamos los querys tenemos dos tipos de datos (que en realidad son los mismo) que son:

* Argumentos (Se usan para cosas que no son valores como las condiciones)
* Valores (Se usa para cantidades que se van a actualizar)

En el dialplan se colocaría en “extensions.conf” lo sig.:

prefix\_nombreFuncion, OBDC\_nombreFuncion, PRE\_nombreFuncion

Después como siempre lo moveremos al bak con el sig. comando:

mv func\_odbc.conf func\_odbc.conf.bak

Volvemos a insertar el comando:  
 nano func\_odbc.conf

# Descripción de la Aplicación Read para IVRs con DB

Vamos a ver sobre la interacción de consultas en el dialplan con la base de datos, para ello se va a realizar un par ejercicios, donde se realizará los sig.:

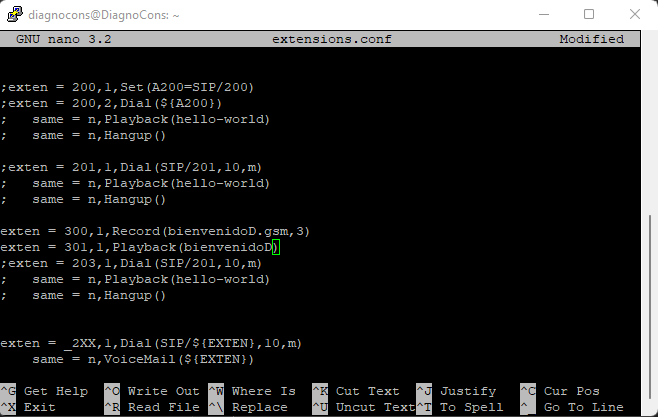
## Ejercicio 1

Un usuario llama a un numero x y el sistema le va a pedir que coloque un ID (identificador de usuario) y luego el sistema le va a proporcionar una respuesta diciéndole su deuda y le va a reproducir la cantidad de la deuda.

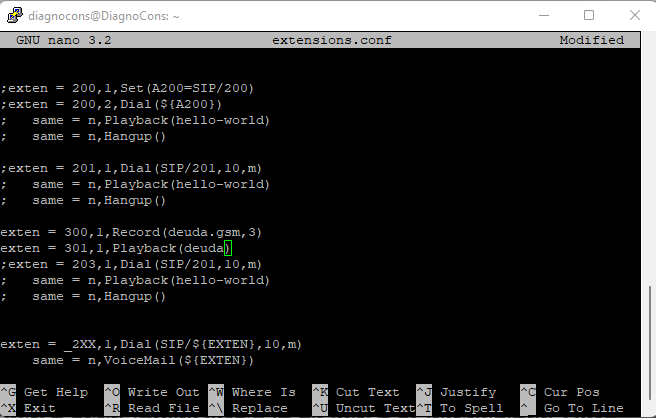
### Escenario:

Usuario llama, le piden si ID y luego le dicen su deuda.

* Audio donde le da la bienvenida y la pide su ID.
  + Vamos al Asterisk y abrimos el “extensions.conf” y revisamos los números para grabar los audios, después lo cargamos en el Asterisk y revisamos los resultados.



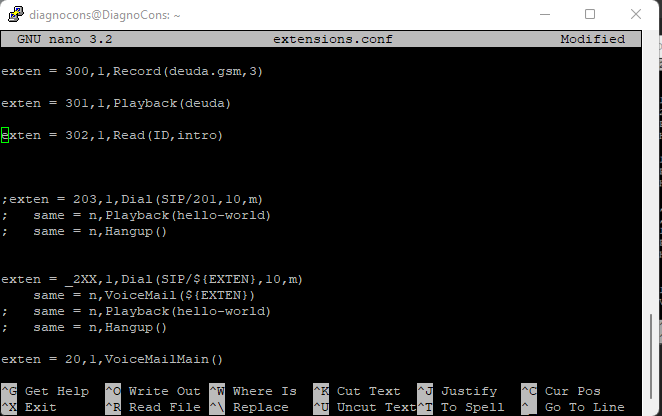
* + Grabamos audios tanto para la introducción como para la deuda.



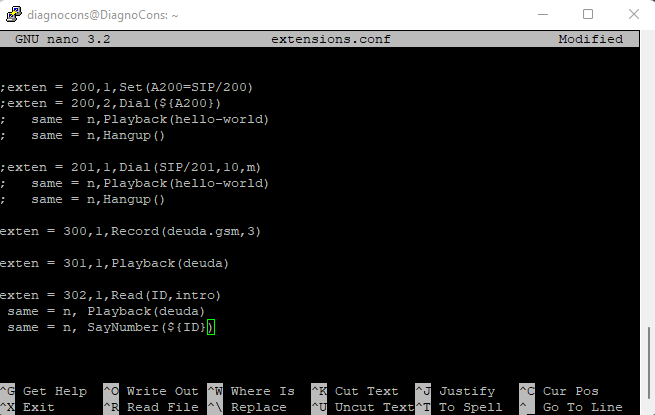
* Read () (Se usará para leer el valor que digite el usuario).
  + Sintaxis.

Read (variable,audio)

* + Abrimos el “extensions.conf” y creamos una extensión para que lea el ID y lo guarde.



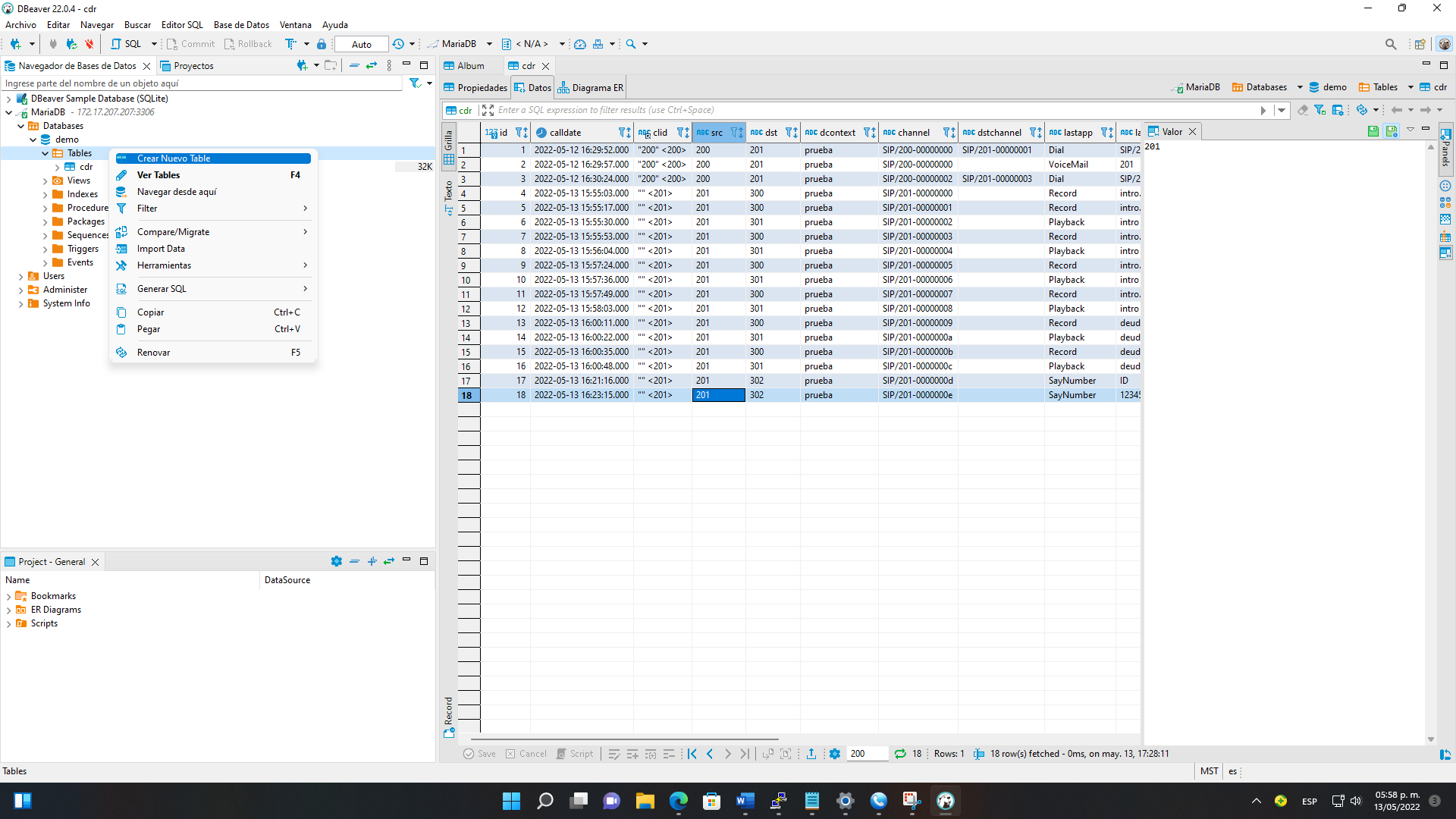
* Audio donde dice “Su deuda es de:” y después reproducir la deuda
  + Después reproducimos el audio deuda y después reproducirá el digito te la duda (Para casos del ejercicio usaremos el mismo ID).



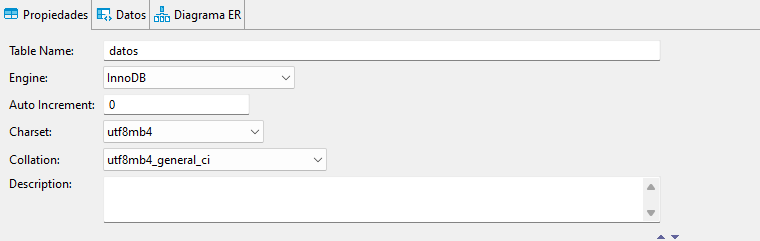
* Cargamos en el Asterisk.

### Interacción con la base de datos.

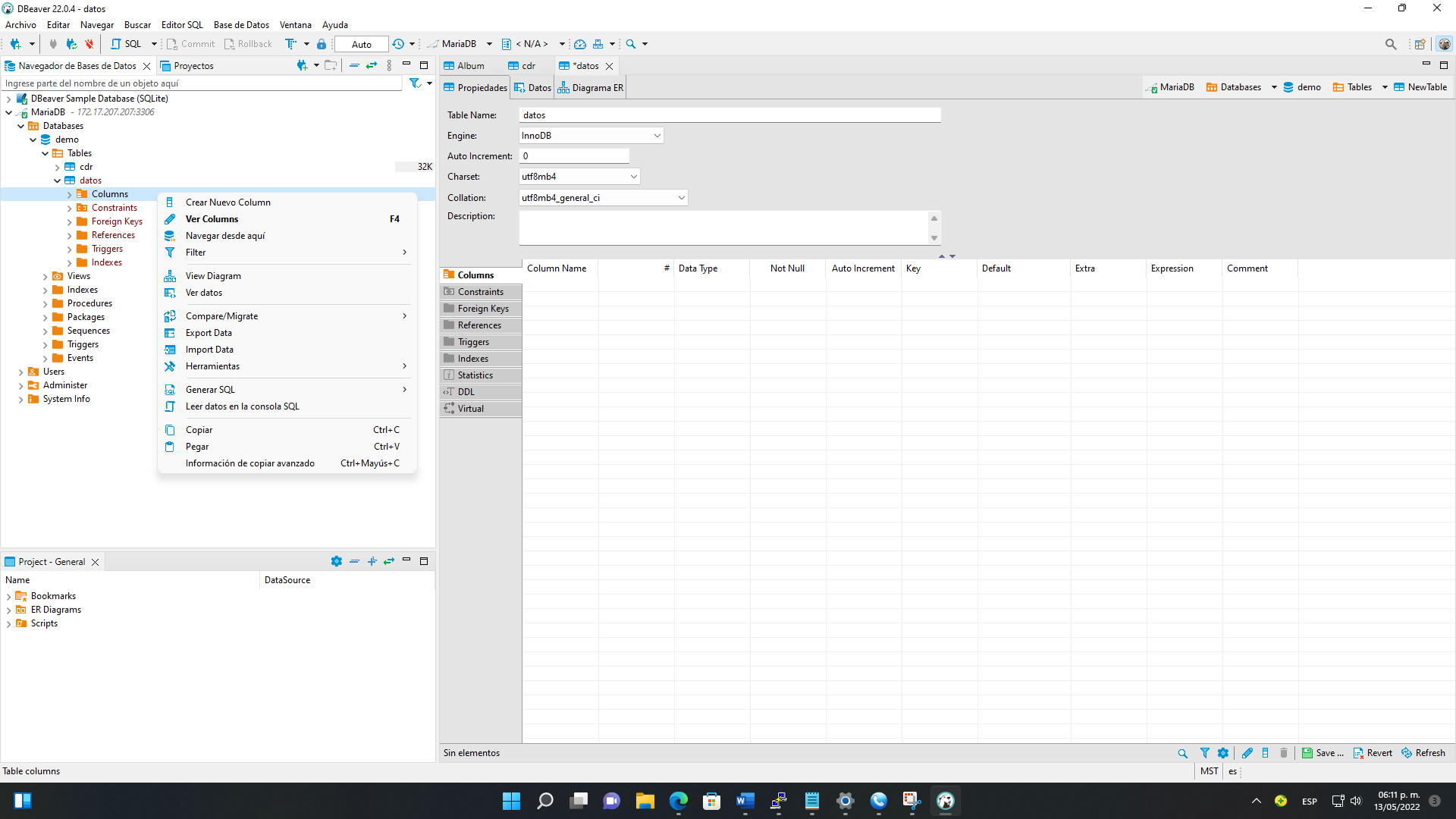
* Lo primero será tener una base de datos, en este caso usaremos la de “demo” después crearemos una nueva tabla llamada, para ello le hacemos click derecho en tablas y “Crear Nuevo Table”.



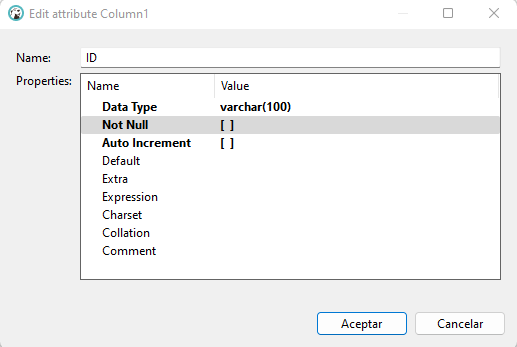
* Después la llamaremos “datos”.



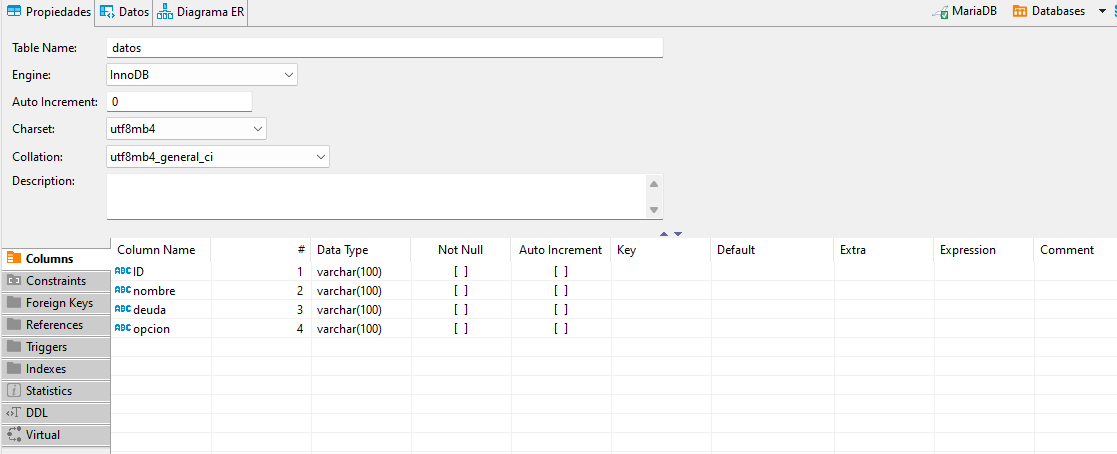
* Luego será crear columnas abrimos la tabla “datos” le hacemos click derecho en “columnas” y le damos en “Crear Nuevo Column”.



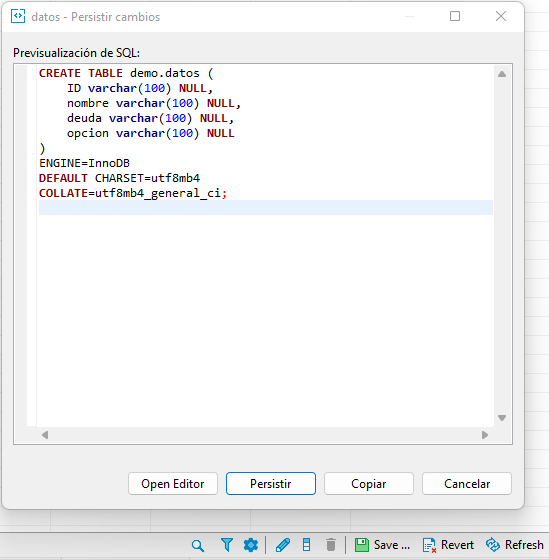
* Después le asignamos nombre y tipo de dato que vamos usar.



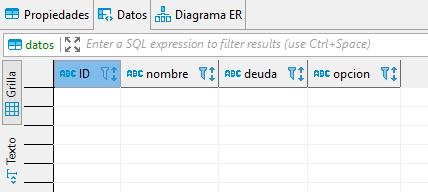
* Se realizarán las sigs. columnas:
  + ID
  + nombre
  + deuda
  + opción



* Después le damos en “Save” y saldrá un cuadro que nos mostrará el query o SQL y le damos en “Persistir”.



* Si vamos a “Datos” ya tendremos creadas nuestras columnas.



* Le asignaremos valores de prueba, para hacerlo, le hacemos click en el botón “añadir nuevo registro”.

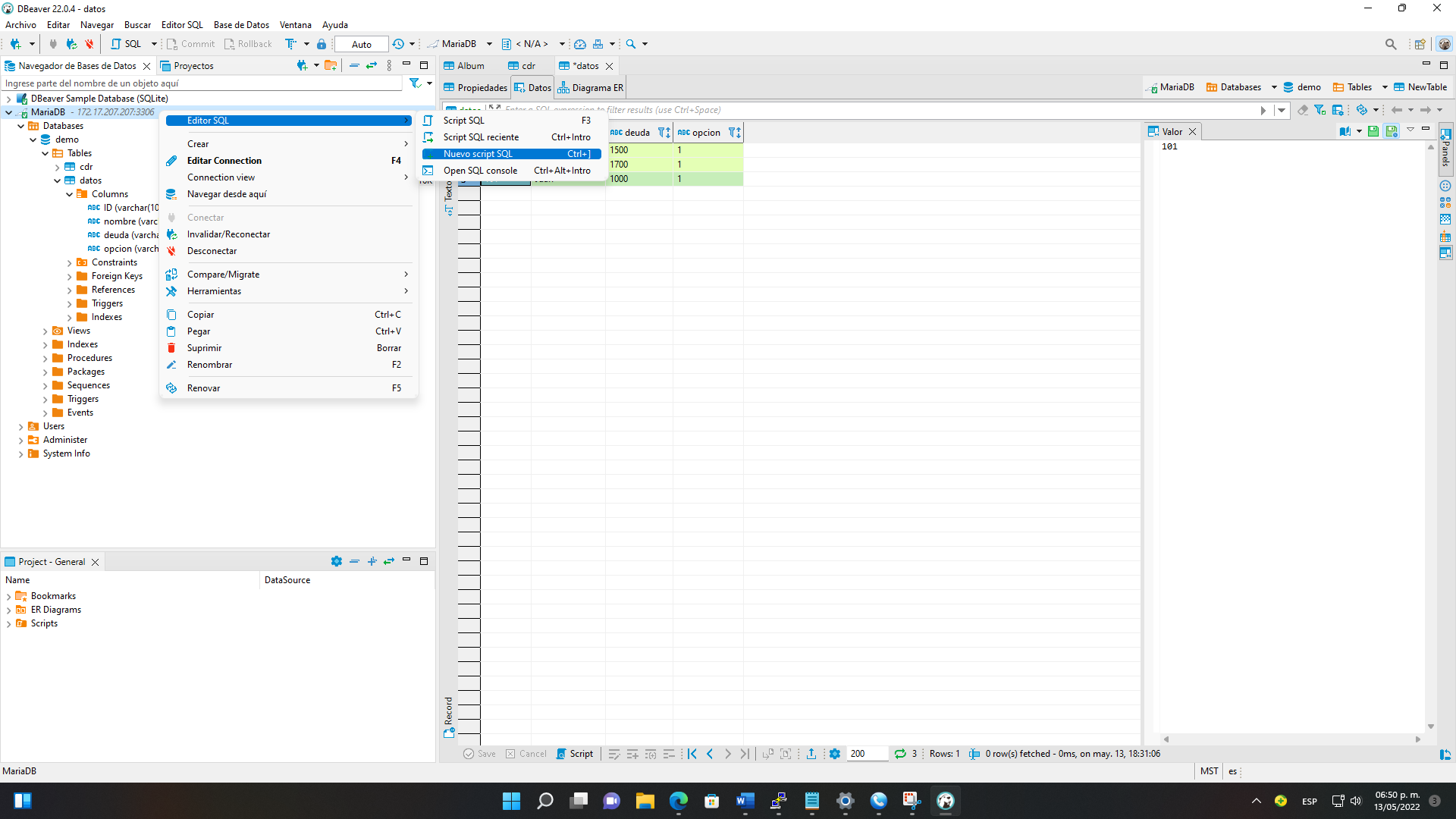


* Una vez hecho lo guardamos en el boton “Save”.



### Seleccionando datos desde la base de datos desde el dialplan.

* Para interactuar con la tabla con dbeaver, en la conexión de base de datos le hacemos click derecho, luego en “Editor SQL” y luego en “Nuevo Script SQL”

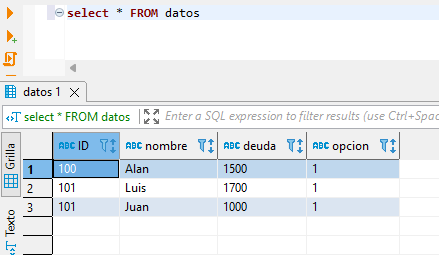


* Le escribiremos lo sig.:

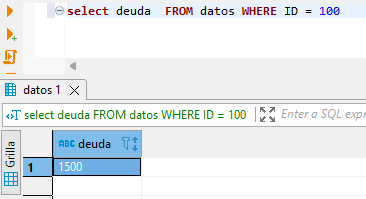
USE demo (primero ejecutar esto, donde le indicamos la base de datos que va a usar IMPORTANTE)



SELECT \* FROM datos (Nos mostrara todos los datos de la tabla)

* 
* Nota: en el asterisco podemos poner el campo o columna que deseamos ver, el asterisco significa todo.
* Si le colocamos:

SELECT deuda FROM datos WHERE ID = 100 (Nos mostrara la deuda del ususario con ID 100 en este caso)



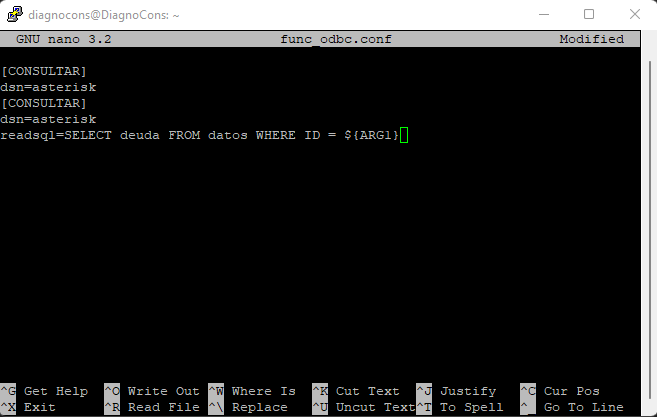
Nota: este será el query que vamos a usar

* Después nos dirigimos al archivo “func\_odbc.conf” y colocaremos lo sig.:

[CONSULTAR] (ID)

dsn=asterisk (Conexión a la base de datos)

readsql=SELECT deuda FROM datos WHERE ID = ${ARG1}



Nota: el ID será el argumento1, cuando llame a la función odbc, se pondrá “${ARG1}”. Los valores “VAL1…” van a ser cuando usemos valores que insertamos o actualizamos en la base de datos, todo lo demás será argumento.

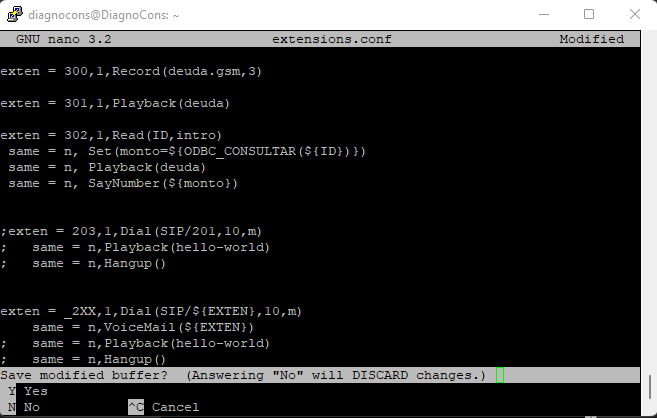
* Luego en “extensions.conf” modificaremos nuestro query, de la sig. manera:

exten = 302,1,Read(ID,intro)

same = n, Set(monto=${ODBC\_CONSULTAR(${ID})}) (Estamos hiciendo una consulta mandando llamar a la función consultar con el valor del ID, monto sirve para el valor que sale)

same = n, Playback(deuda)

same = n, SayNumber(${monto})



* Después lo cargamos en la consola Asterisk con “dialplan reload” y “reload” (si llega a dar error solo reinicie el servidor).

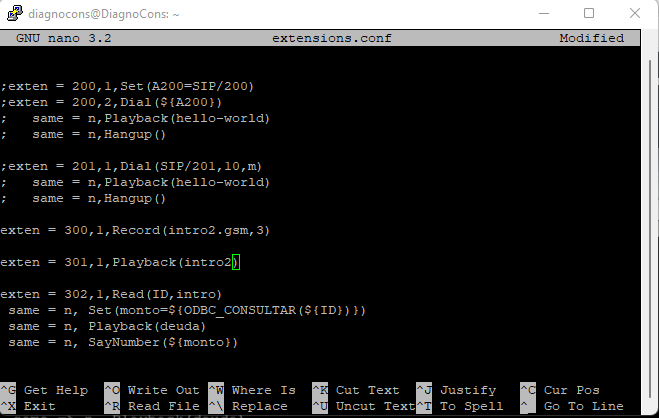
## Ejercicio 2

Un usuario llama a un numero y luego le pide su número de ID y después elegirá una opción del 1 al 9, el usuario digitará una opción y esa opción que queda se queda grabado en la base de datos

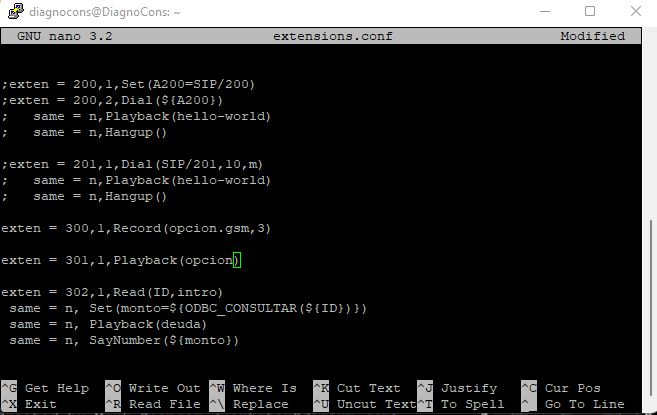
### Escenario:

Usuario llama, le piden su DI y luego la opción a escoger.

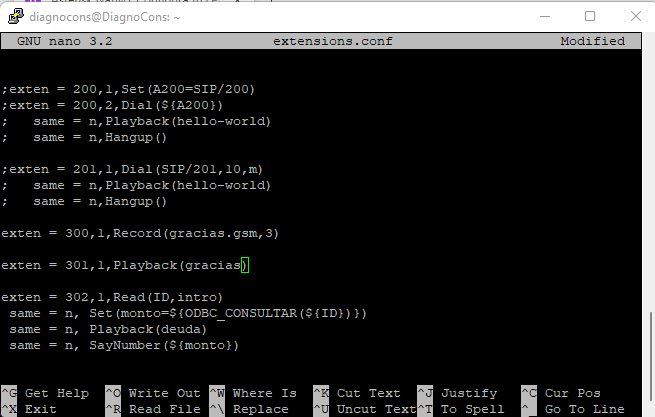
* Audio donde da la bienvenida y le pidan su ID
  + Abrimos el archivo “extensions.conf” para grabar un nuevo audio de bienvenida llamado “intro2.gsm”.



* Audio donde le pidan su opción a escoger.
  + Volvemos a abrir el archivo “extensions.conf” para grabar un audio de selección de opciones llamado “opcion.gsm”.

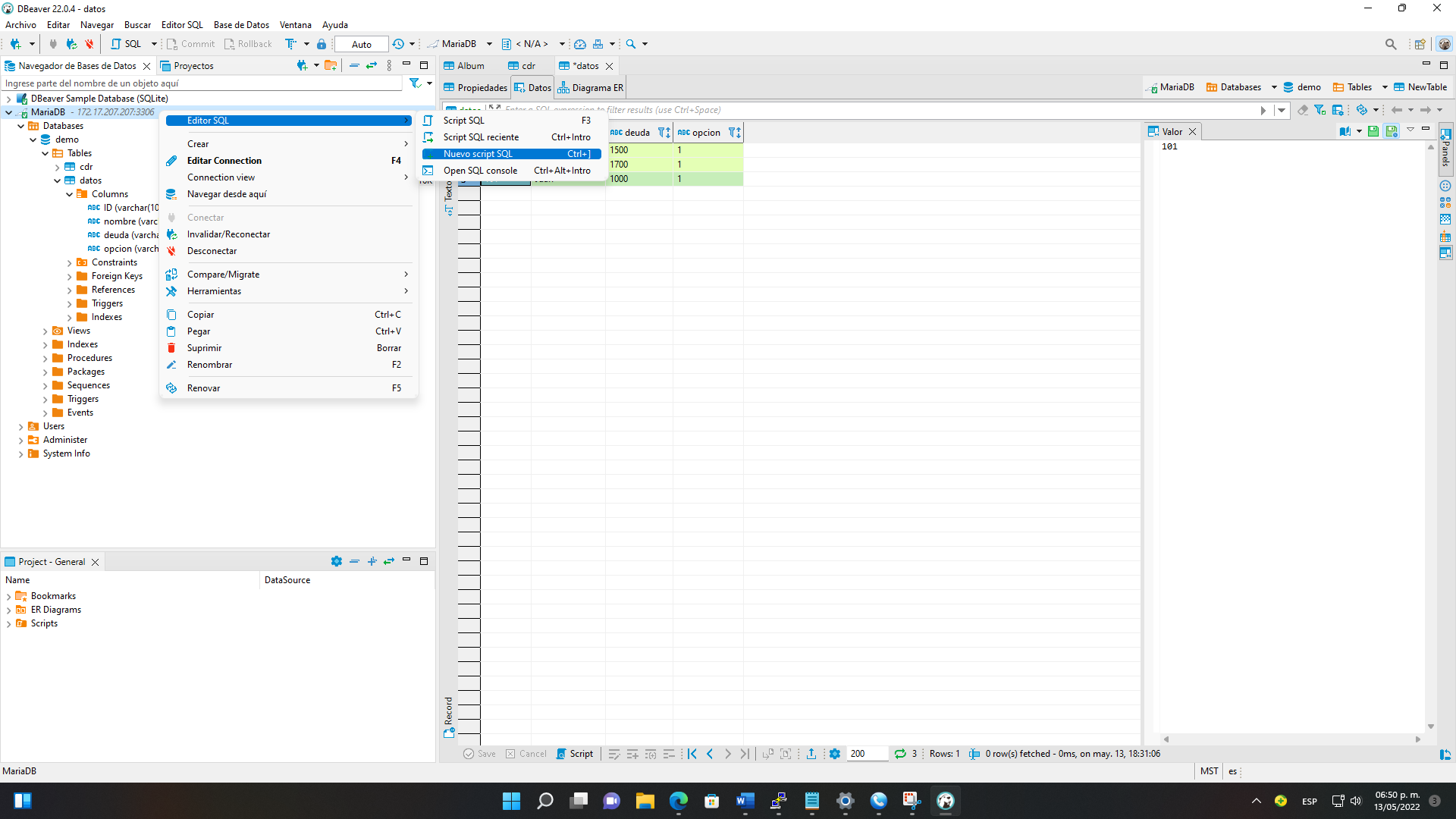


* Audio donde agradezcamos su elección.
  + Volvemos a abrir el archivo “extensions.conf” para grabar un audio donde agradezcamos su elección llamado “gracias.gsm”.



### Seleccionando datos desde la base de datos.

* Para interactuar con la tabla datos con dbeaver, en la conexión de base de datos le hacemos click derecho, luego en “Editor SQL” y luego en “Nuevo Script SQL”



* Le escribiremos lo sig.:

USE demo (primero ejecutar esto, donde le indicamos la base de datos que va a usar IMPORTANTE)

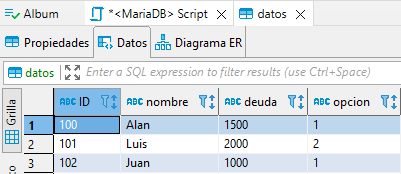


* Después vamos a alterar los valores de opción para ello ejecutaremos el sig. script:

UPDATE datos set opcion=2 WHERE ID=101 (Con esto cambiaremos el valor del campo opcion de 1 a 2 donde el ID tenga 101)

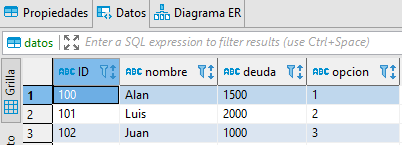


* Después actualizamos la tabla datos, la abrimos y volvemos a cerrar.



* Hacemos lo mismo, pero ahora con el ID 102 para darle un valor de 3.

UPDATE datos set opcion=3 WHERE ID=102

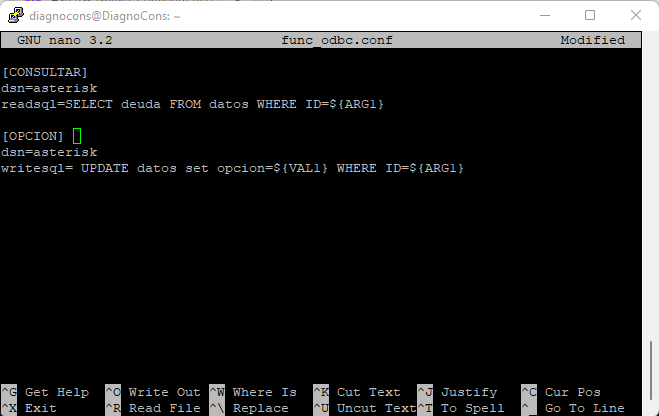


* Después nos dirigimos al archivo “func\_odbc.conf” y colocaremos lo sig.:

[OPCION] (ID)

dsn=asterisk (Conexión a la base de datos)

writesql= UPDATE datos set opcion=${VAL1} WHERE ID=${ARG1}



* Después nos dirigimos al archivo “extensions.conf” y colocaremos lo sig.:

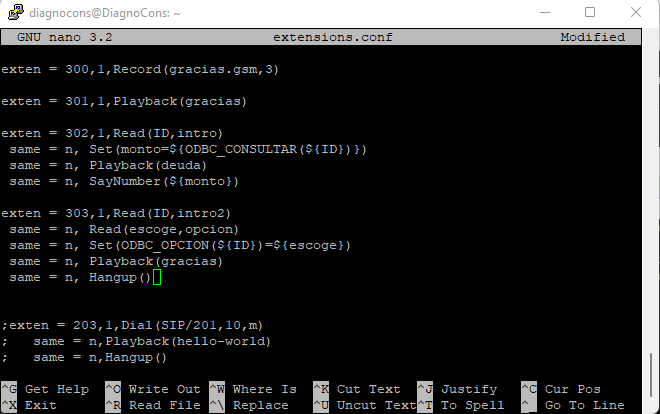
exten = 303,1,Read(ID,intro2)

same = n, Read(escoge,opcion)

same = n, Set(ODBC\_OPTION(${ID})=${escoge})

same = n, Playback(gracias)

same = n, Hangup()



* Después lo cargamos en la consola Asterisk con “dialplan reload” y “reload” (si llega a dar error solo reinicie el servidor).
* Al añadir un valor nuevo se vera reflejado en la base de datos.

