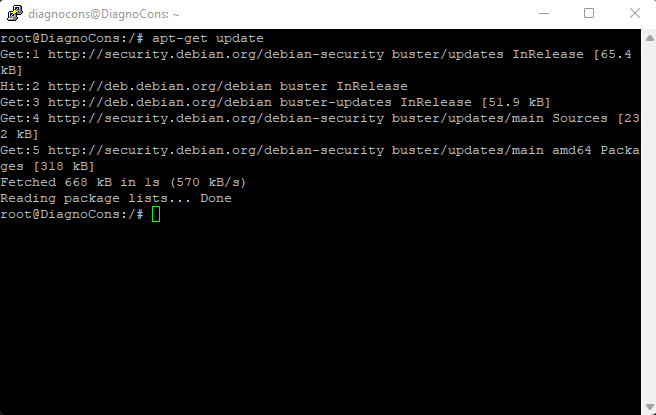
# Instalación de base de datos.

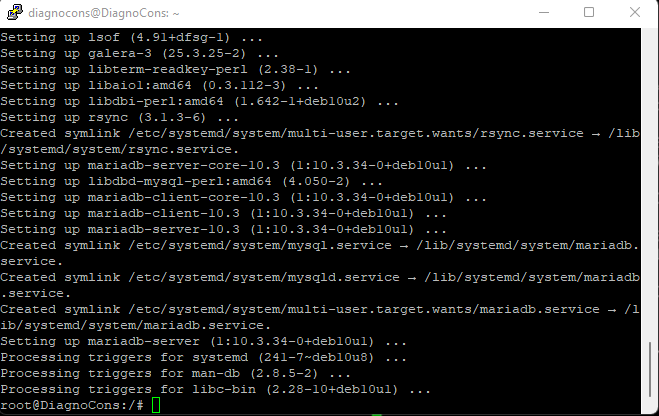
Primero instalaremos “MariaDB”, lo primero será hacer es actualizar los repositorios con el sig. comando:

apt-get update



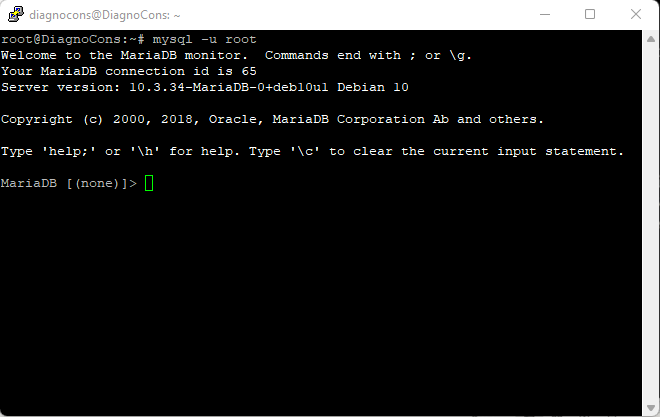
Luego para instalar el “MariaDB” usaremos el sig. comando:

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



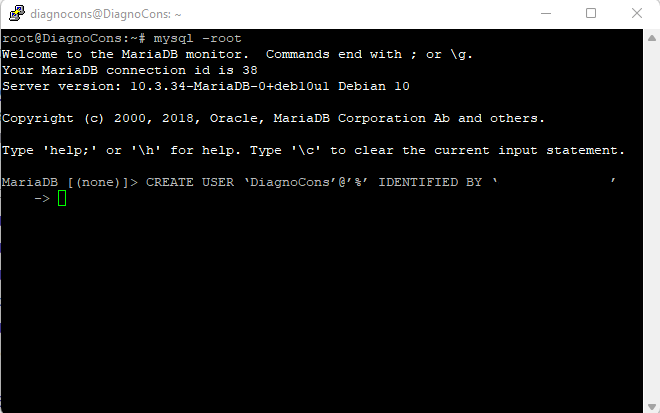
Una vez instalado para entrar a la base de datos usamos el sig. comando:

mysql -u root



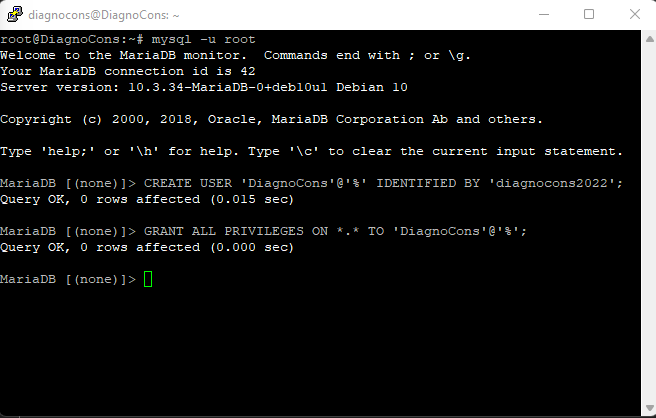
Lo primero será crear un usuario y una contraseña para ello usaremos los sigs. comandos:

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'@'%';



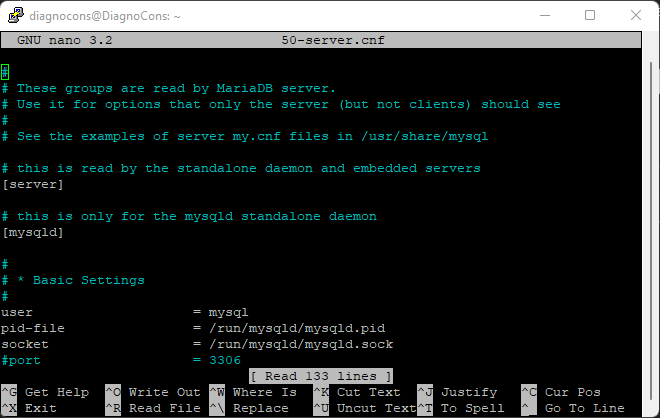
Nota: cuando se le pone (@'%') significa que este usuario puede acceder desde cualquier lugar.

Con este usuario se puede trabajar desde consola, pero no es muy práctico 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:

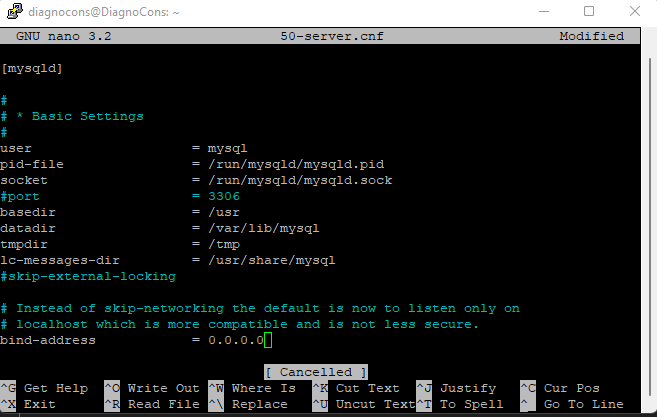
cd /etc/mysql/mariadb.conf.d/

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

nano 50-server.cnf



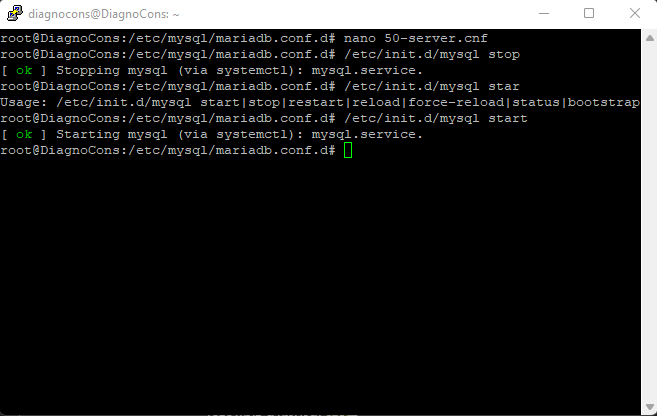
Aquí buscaremos algo llamado “bind-address”, cuando se le pone 127.0.0.1 solo se puede acceder desde la misma PC(Servidor), pero si le cambiamos a 0.0.0.0 se le puede acceder desde cualquier ubicación o un determinado rango de red, por lo pronto lo dejaremos en que pueda acceder desde cualquier lugar.



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

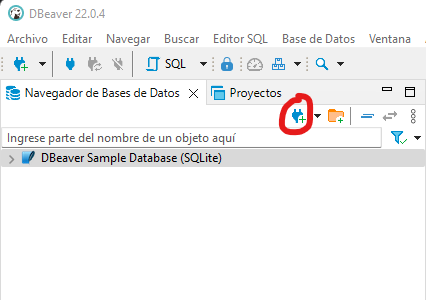
/etc/init.d/mysql stop

/etc/init.d/mysql start

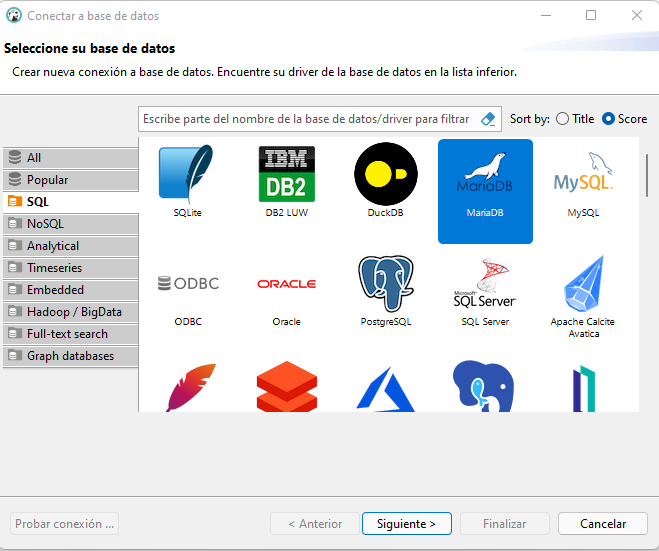


Una vez reiniciado el servicio de MySQL vamos a usar un gestor de base de datos, para ello descargaremos en la PC (Windows) un programa llamado dbeaver.

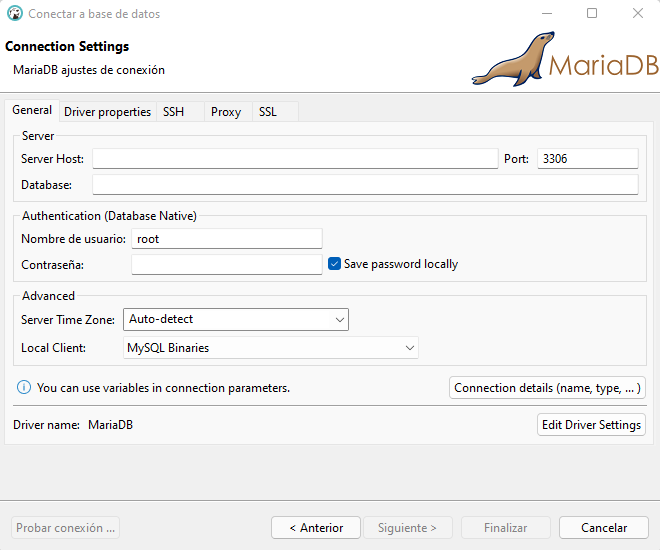
Una vez instalado el dbeaver abrimos el programa y vamos a establecer una nueva conexión, para ello le hacemos click en “nueva conexión”.



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



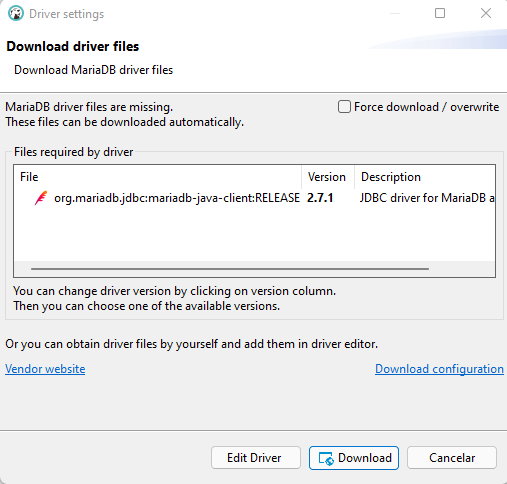
Después en la base de datos colocamos lo sig.:



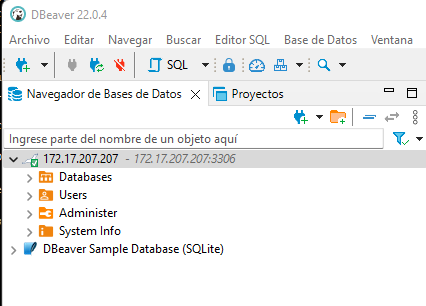
* Server Host: la IP del servidor.
* Username: usuario del servidor.
* Password: la contraseña.

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.



Instalación de ODBC Conector para MariaDB.

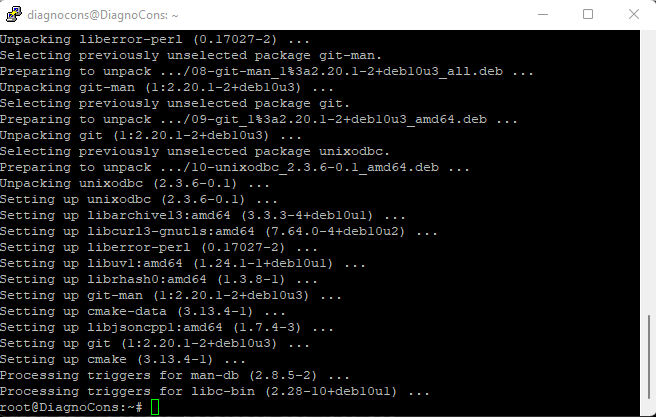
Lo primero que tenemos que hacer es instalar el ODBC y luego instalar el conector. El driver cambia de nombre de nombre dependiendo la distribución y dependiendo de la base de datos que conecta, en este caso que estamos usando Debian y MariaDB se llamara “libmaodbc.so”.

Instalación de ODBC y paquetes previos.

Lo primero será descargar el primero paquete con este comando:

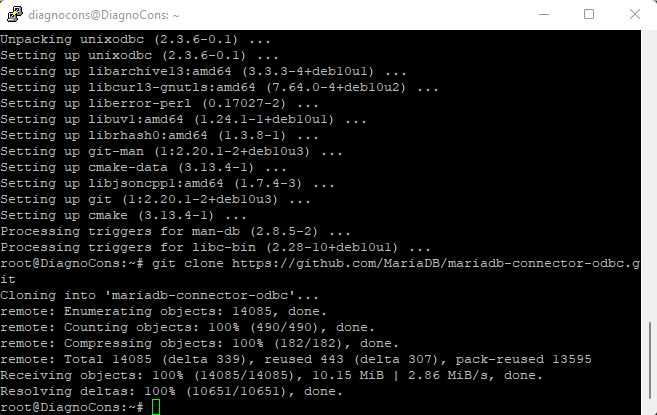
apt-get install unixodbc git cmake

* 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:

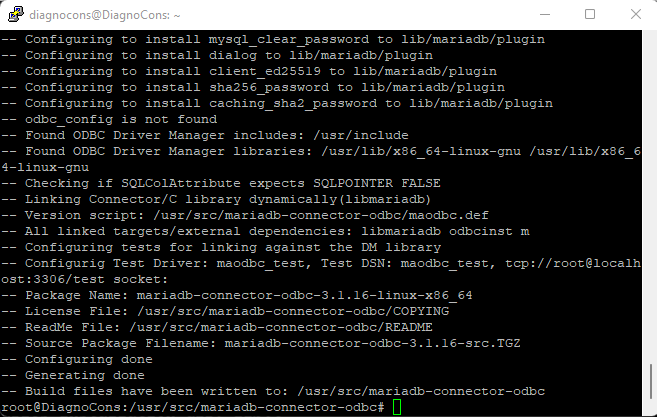
git clone https://github.com/MariaDB/mariadb-connector-odbc.git



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

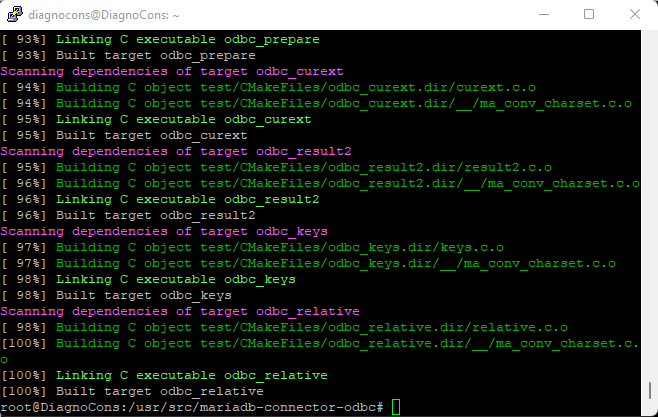
cd mariadb-connector-odbc

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



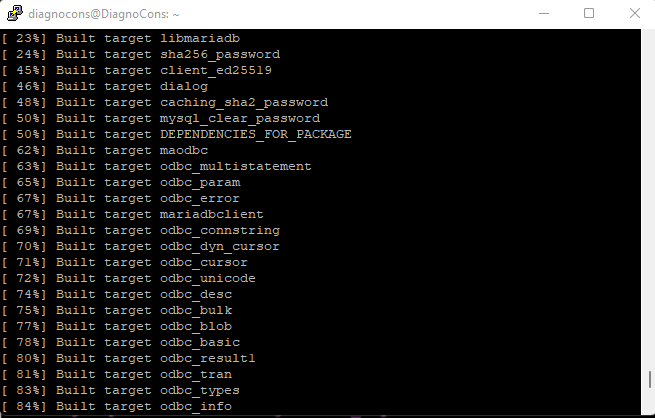
Luego insertamos el sig. comando:

cmake --build . --config RelWithDebInfo

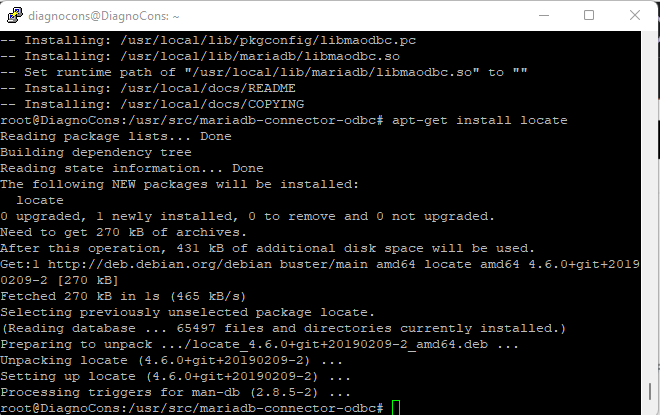


Por último, insertamos el sig. comando:

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:

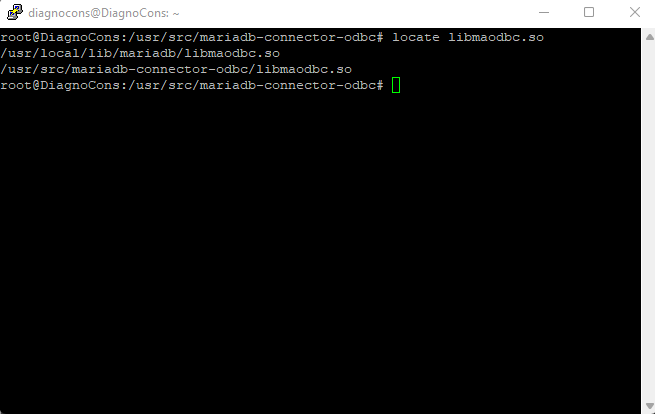


Una vez instalado actualizamos su base de datos para ello usaremos el sig. comando:

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.

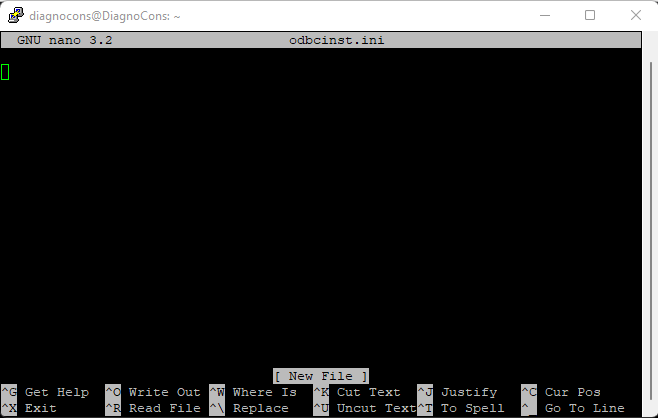
Recordemos:

* 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

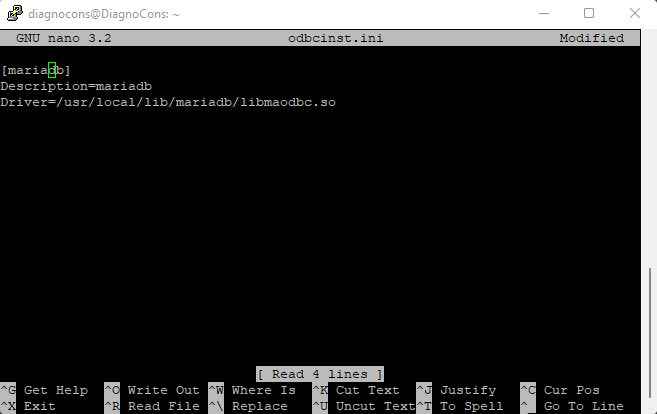


Lo primero será crear el identificador de nuestro conector, para lo cual colocaremos lo sig.:

[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

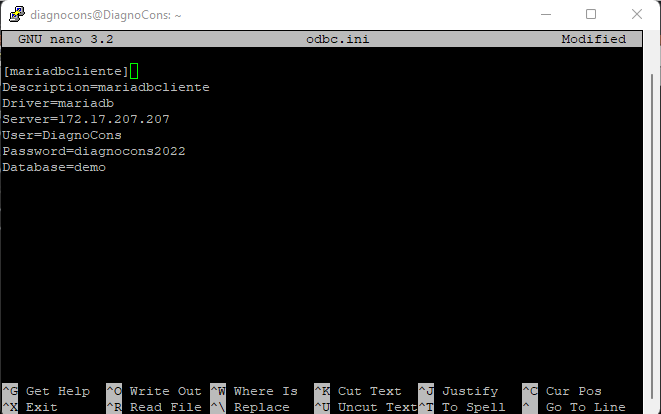
Driver=mariadb (Se coloca el ID que le colocamos al ID de “odbcinst.ini”)

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)

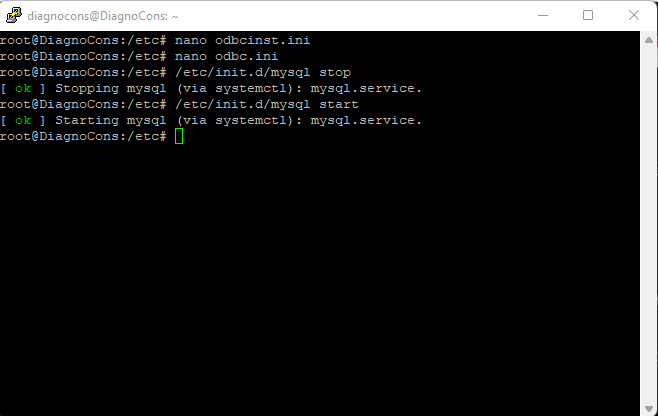


Una vez hecho todo esto salvamos.

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

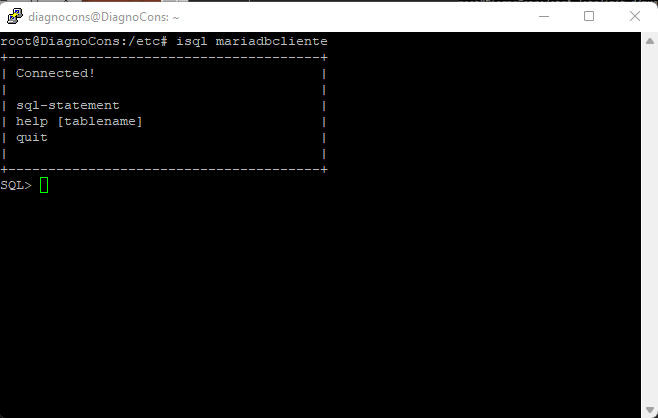
/etc/init.d/mysql stop

/etc/init.d/mysql start



Para saber si hay conexión usamos el sig. comando:

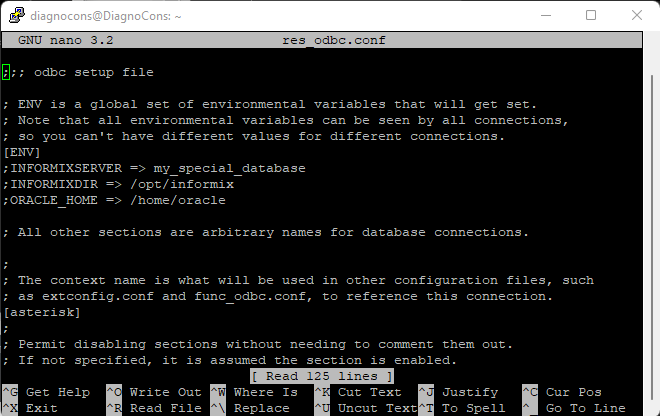
isql mariadbcliente



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

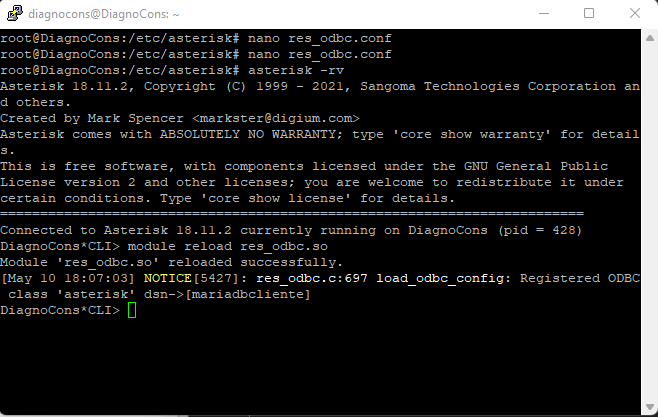
dsn=mariadbcliente (ID del “odbc.ini”)

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



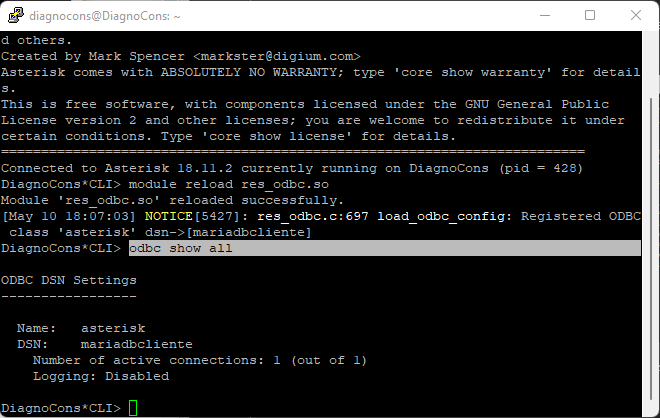
Después entramos en la consola Asterisk y recargamos el modulo con el sig. comando:

module reload res\_odbc.so



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

odbc show all

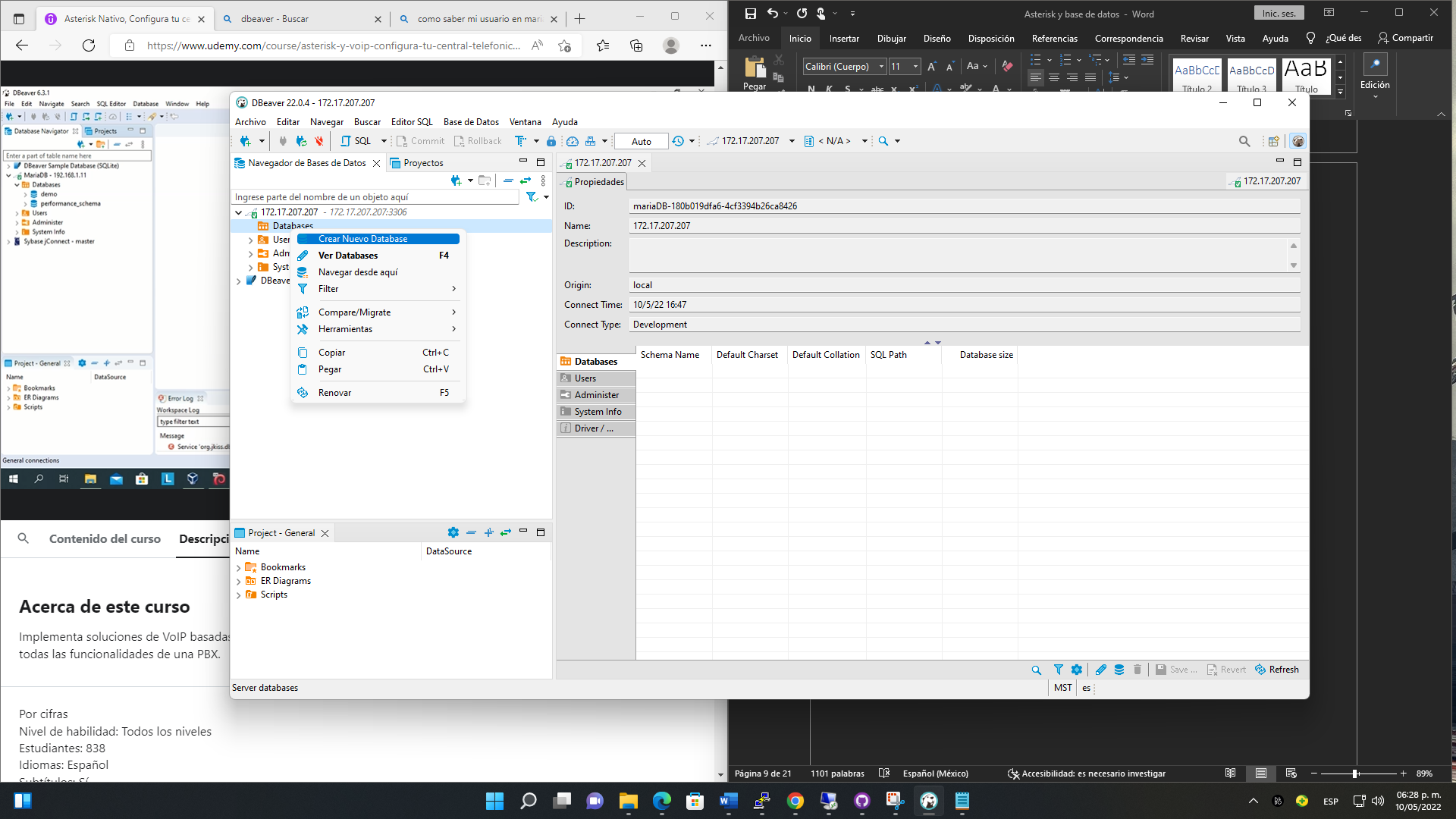


Nota: nos mostrara un mensaje de que la conexión está hecha.

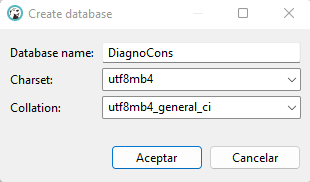
Con esto el Asterisk está listo para conectar con la base de datos.

# Crear base de datos

Lo primero que haremos será crear una base de datos llamada “DiagnoCons” en “MariaDB”, para hacerlo en “Databases” le daremos click derecho y luego “Crear nuevo Database”.



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



# 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 DiagnoCons;

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 '',

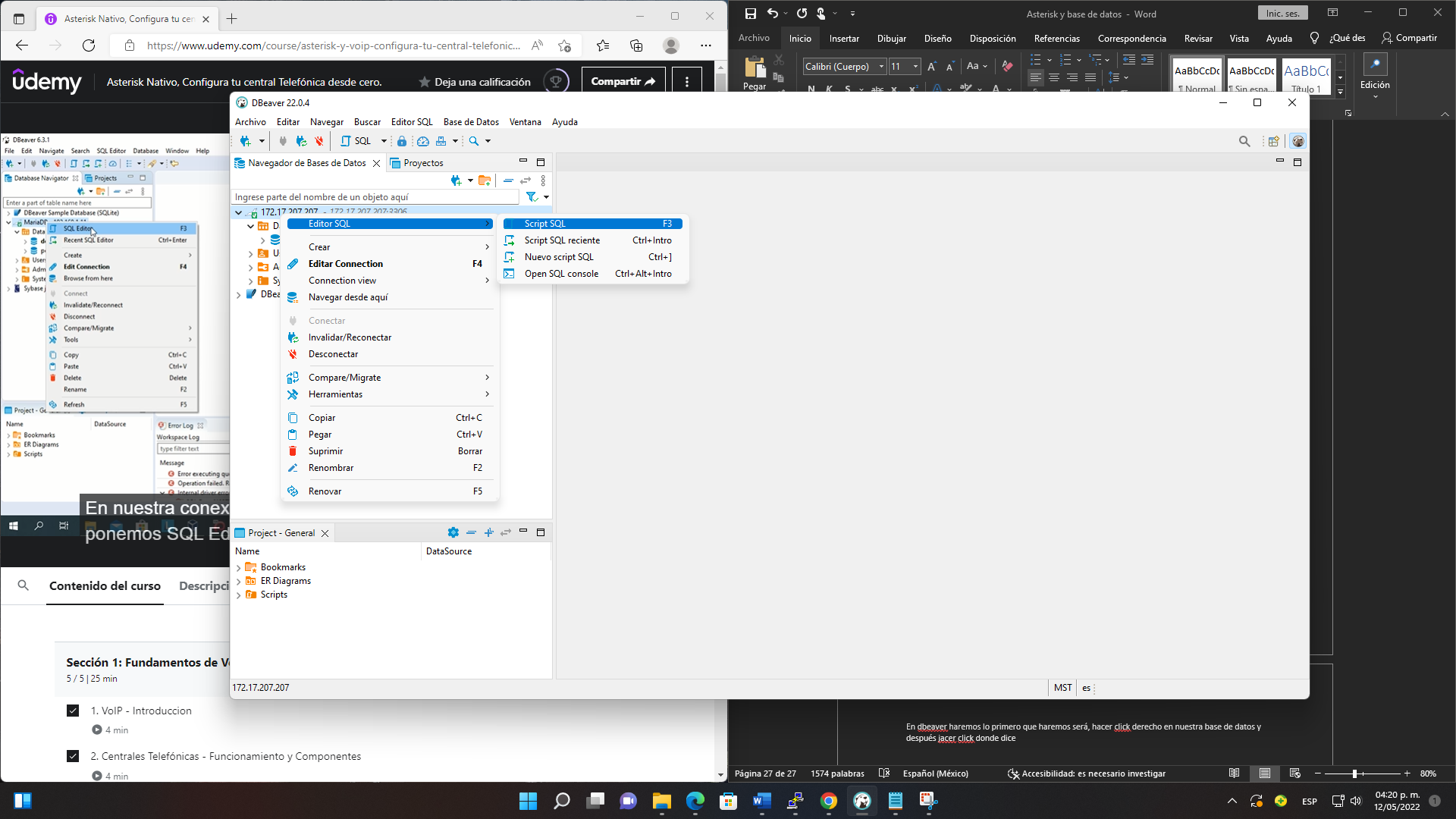
userfield varchar(255) NOT NULL default '',

PRIMARY KEY (`id`),

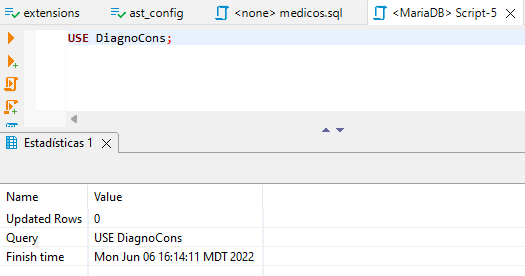
KEY callerid (clid)

);

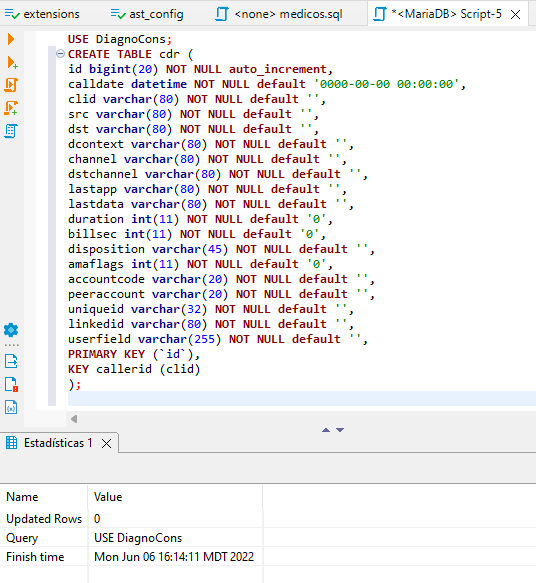
En dbeaver haremos lo primero que haremos será, hacer click derecho en nuestra base de datos y después hacer click donde dice “Editor SQL” y luego “Script SQL”



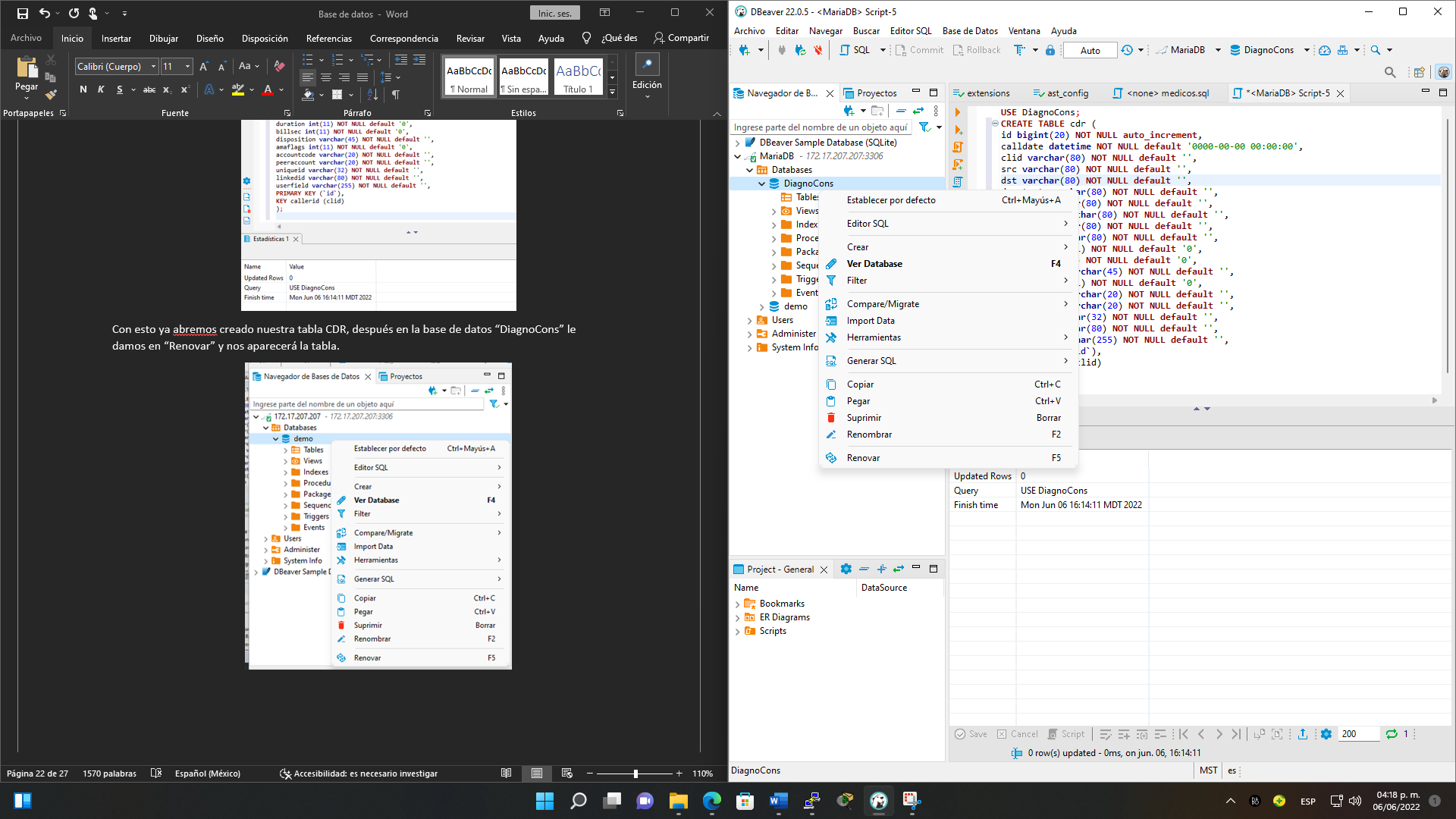
Se abrirá el editor y pegaremos el primer apartado que seria “USE demo;” que le indicaría que use la base de datos “demo”, después lo corremos con el botón de “Ejecutar comando SQL”.

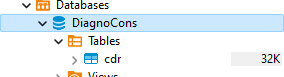


Luego ejecutamos el resto del script.

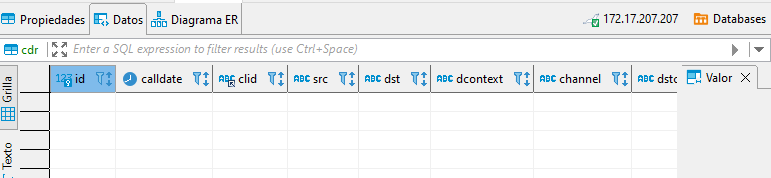


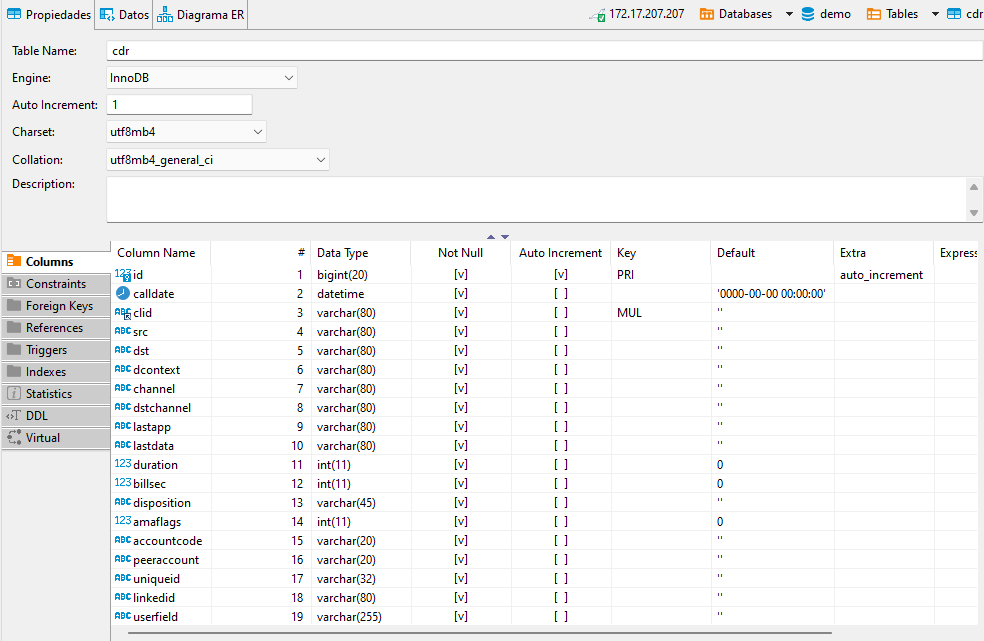
Con esto ya abremos creado nuestra tabla CDR, después en la base de datos “DiagnoCons” le damos en “Renovar” y nos aparecerá la tabla.





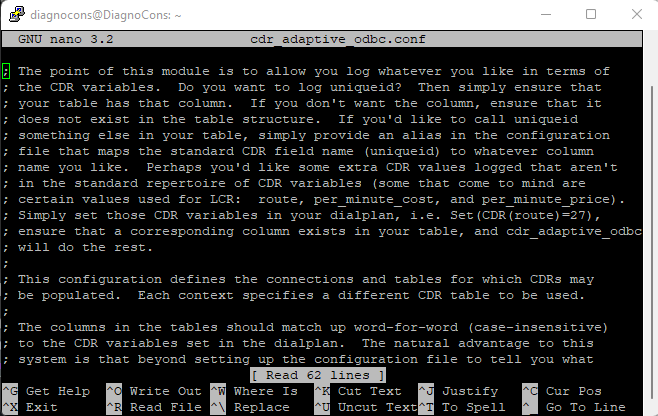
Si le hacemos click a la tabla CDR y nos vamos a “Datos” veremos que está vacía y si vamos a “Propiedades” veremos los campos de la tabla.





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

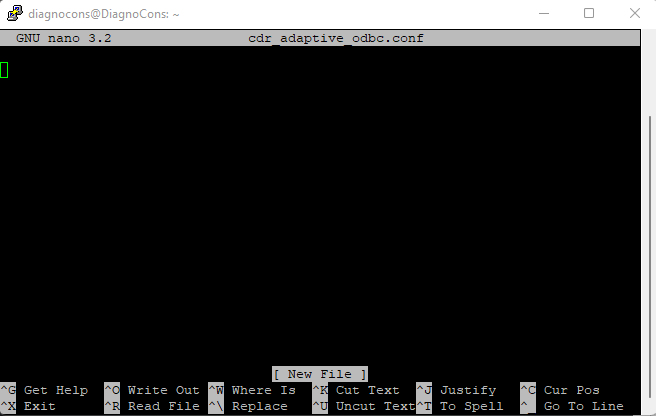


Después como siempre lo moveremos a un bak con el sig. comando:

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



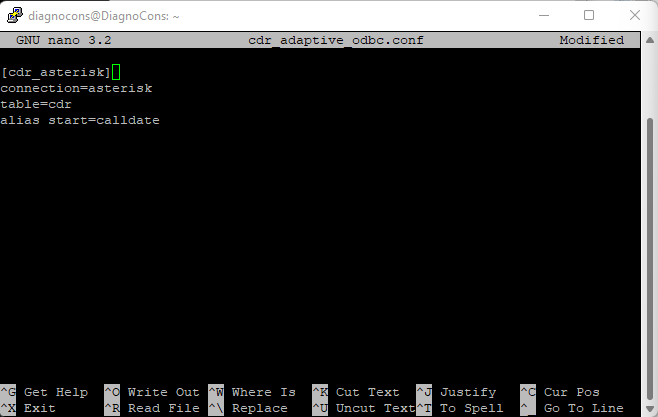
Cada aplicación se modifica su archivo o su configuración propia. Después escribiremos lo sig.:

[cdr\_asterisk] (Nombre de la conexión)

connection=asterisk (Conexión que vamos a usar)odcll

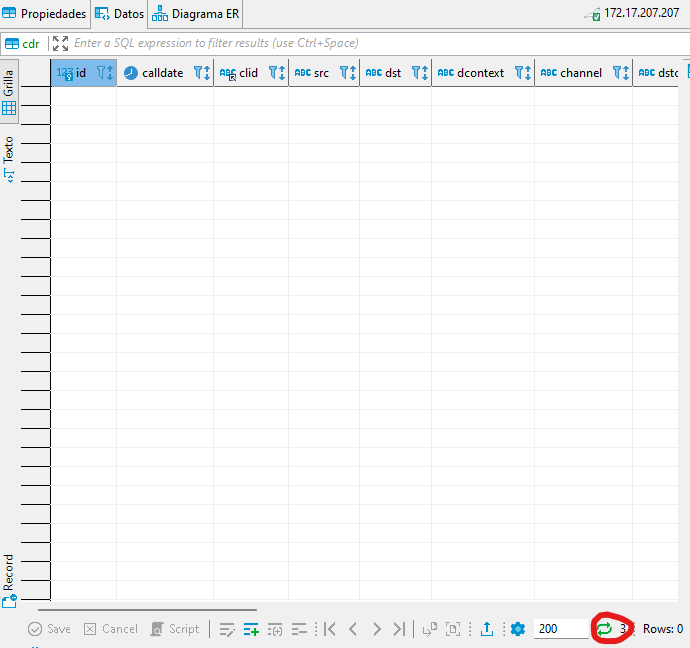
table=cdr (Tabla a la que nos vamos a conectar)

alias start=calldate (Primer campo a usar)

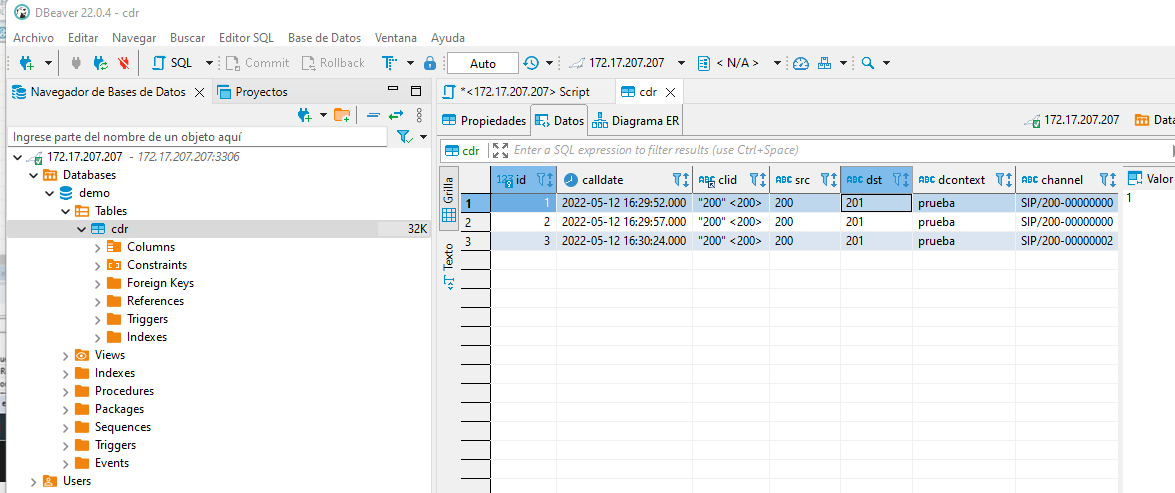


Después guardamos, entramos a la consola Asterisk y usamos un “reload” y hacemos llamadas de prueba.

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.