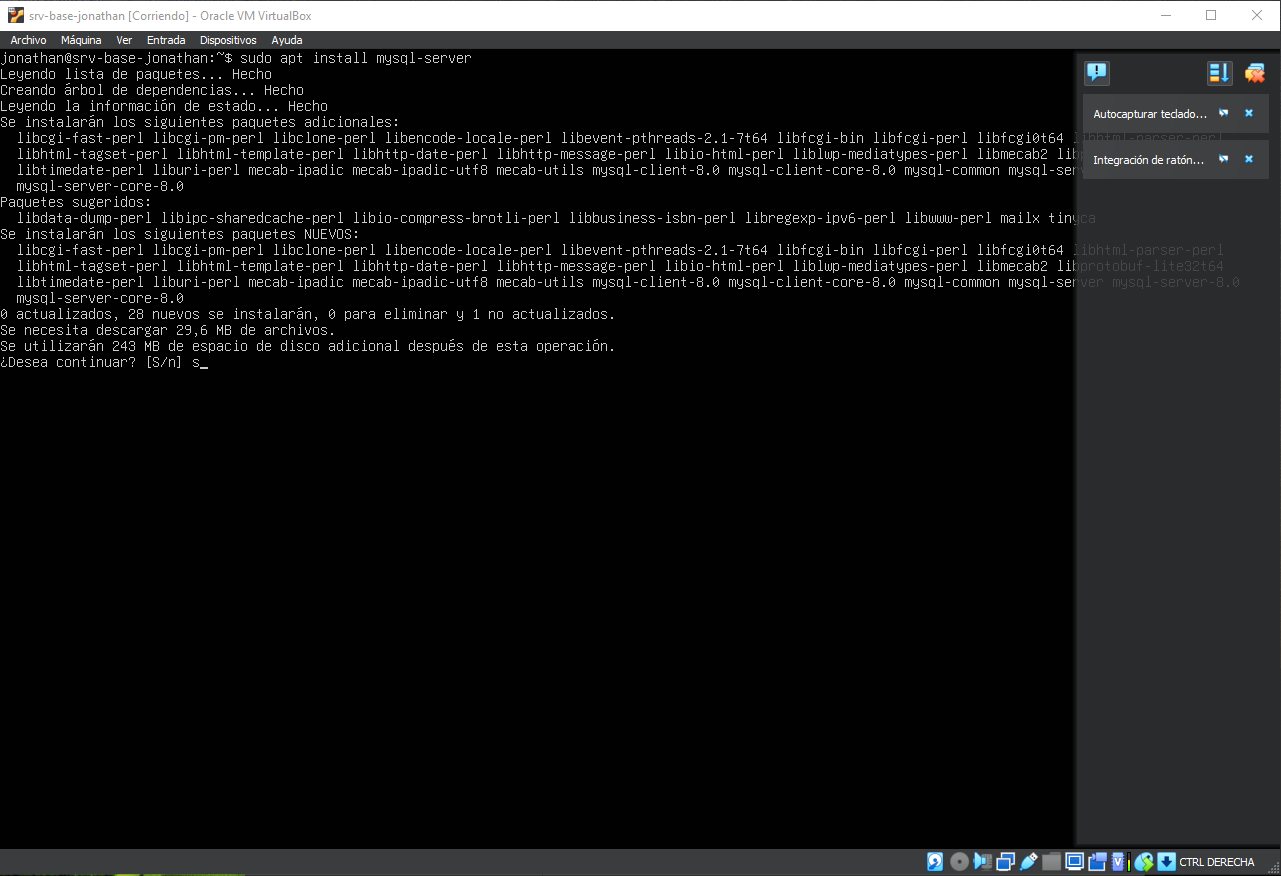
**Implementación de un Servidor de Bases de Datos para Codearts Solutions**

# **Fase 1: Instalación y configuración del servidor de bases de datos**

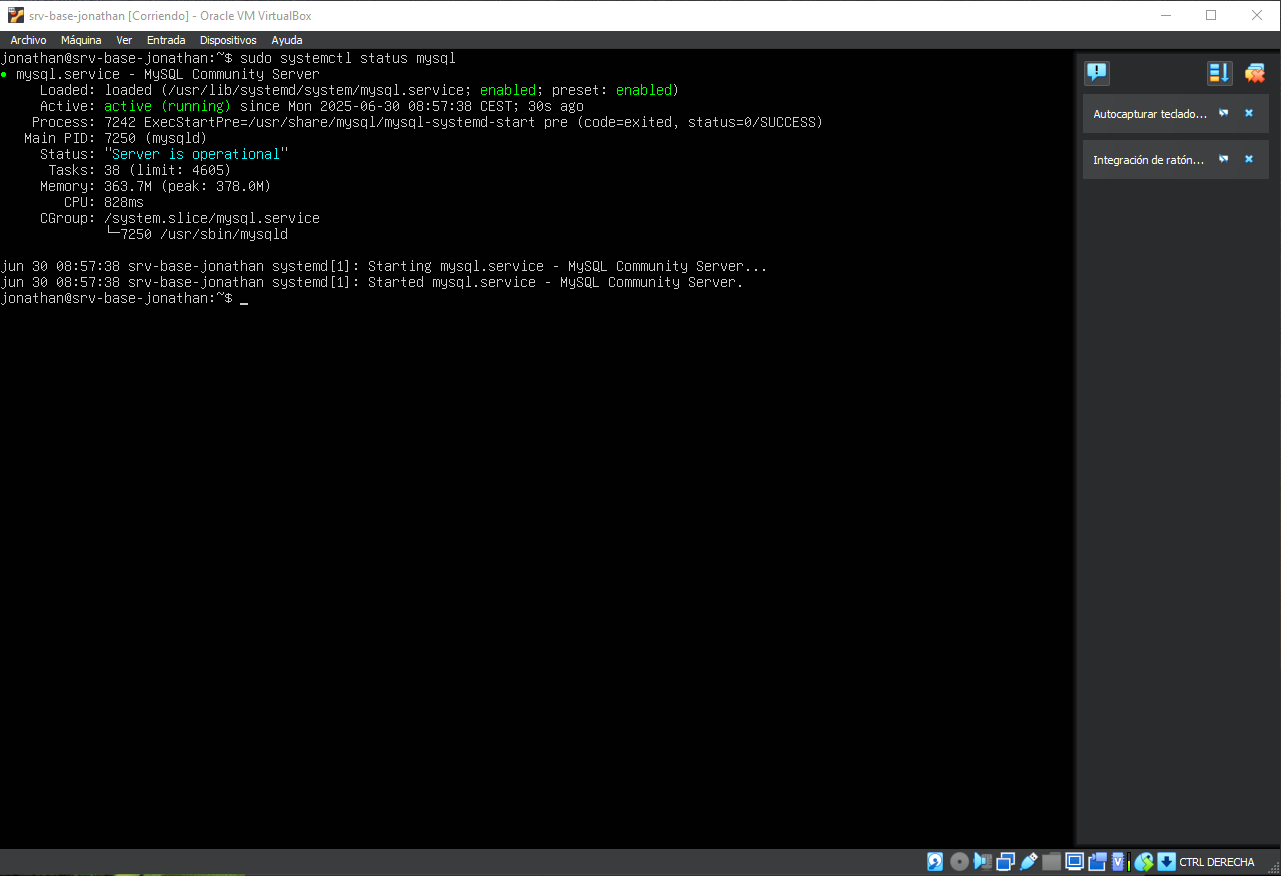
Instalación de MySQL en el servidor a través del comando:

sudo apt install mysql-server



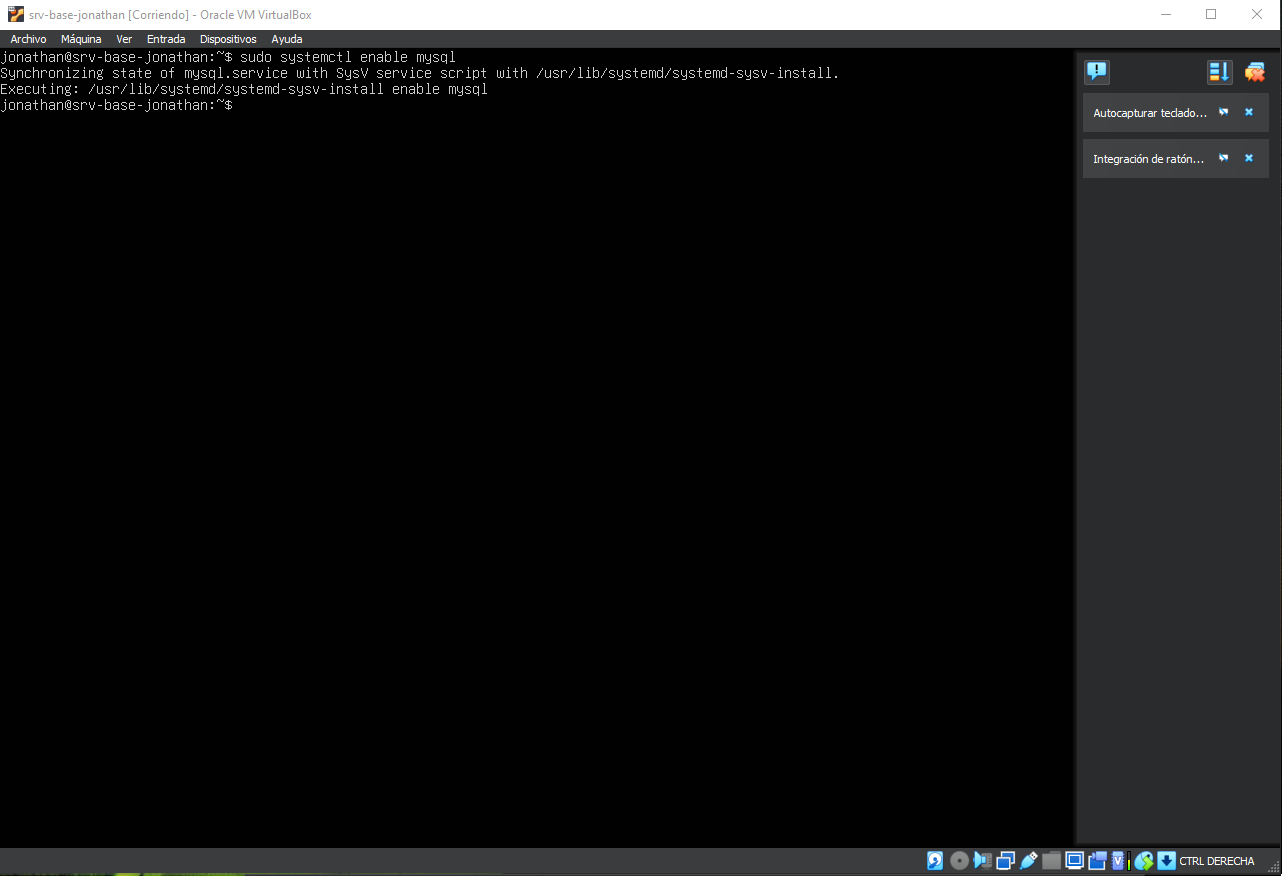
Comprobamos que el servicio este activo despues de la instalación:

sudo systemctl status mysql



Realizamos una configuracion para que mysql se inicie automáticamente al iniciar nuestro servidor:

sudo systemctl enable mysql

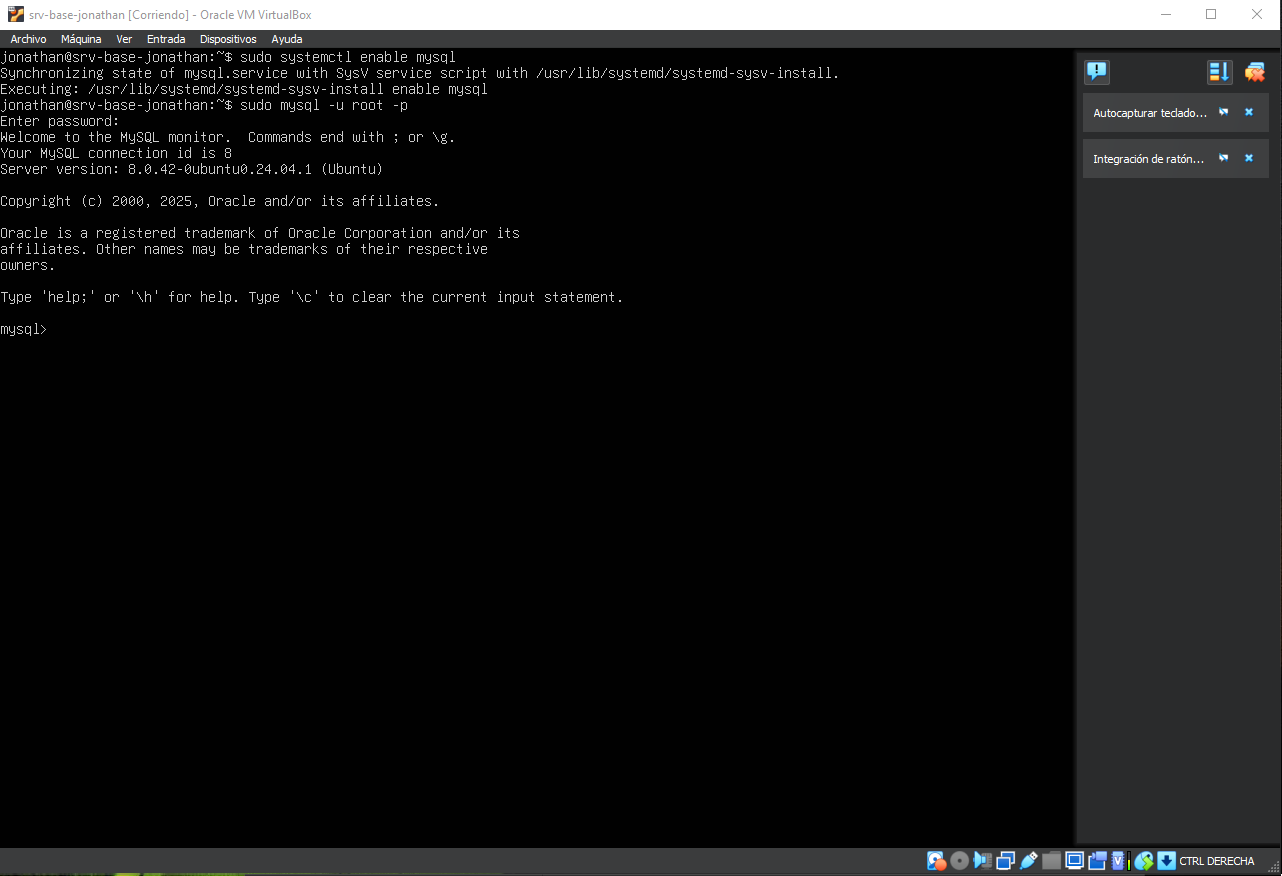


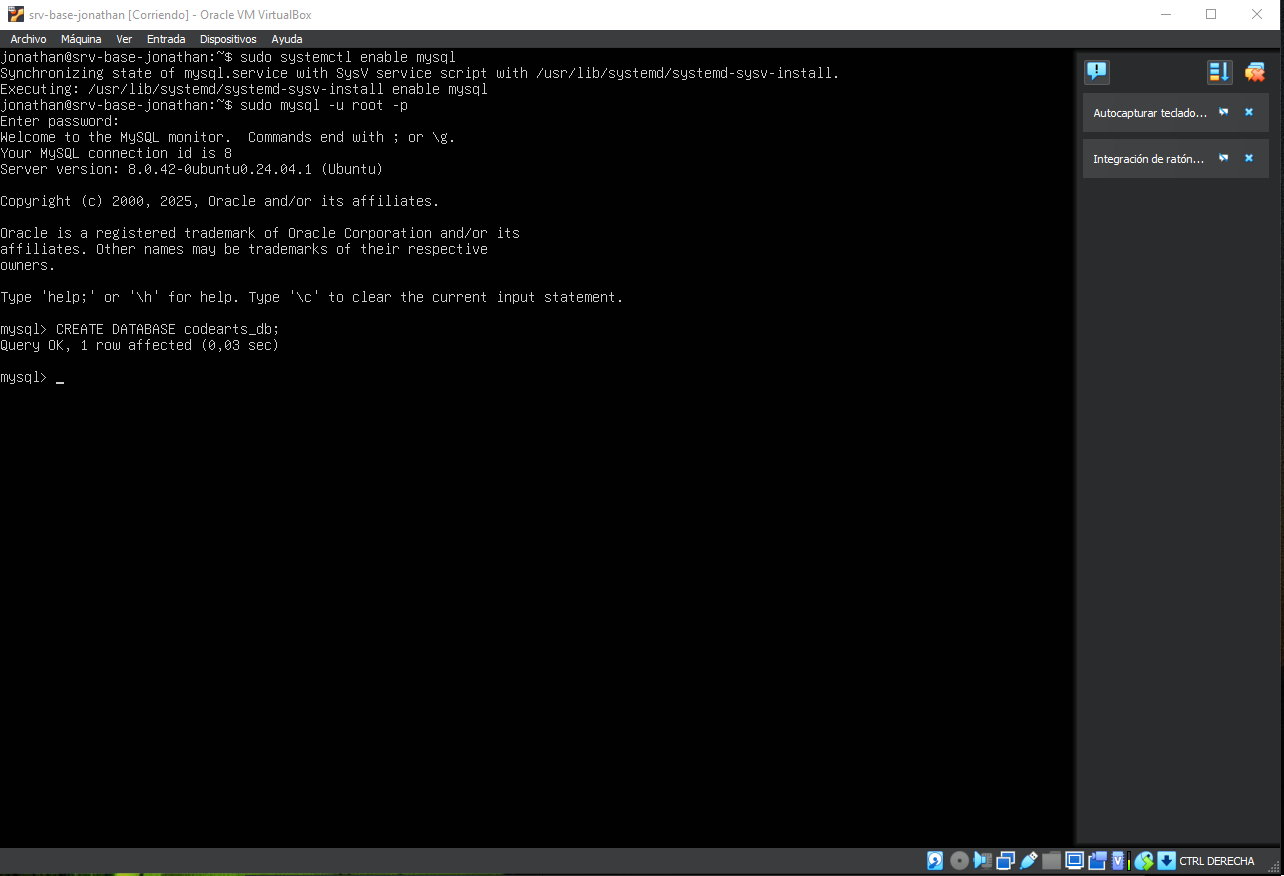
# Fase 2: Creación de base de datos y usuarios

Crearemos una base de datos llamada codearts\_db, accedemos a mysql para poder crear nuestra base de datos:

sudo mysql -u root -p

CREATE DATABASE codearts\_db





Crearemos unos usuarios para la gestión de esta base de datos con diferentes permisos, admin\_db, dev\_user, intern\_user:

CREATE USER 'admin\_db'@'%' IDENTIFIED BY 'Admin123';

GRANT ALL PRIVILEGES ON codearts\_db.\* TO 'admin\_db'@'%';

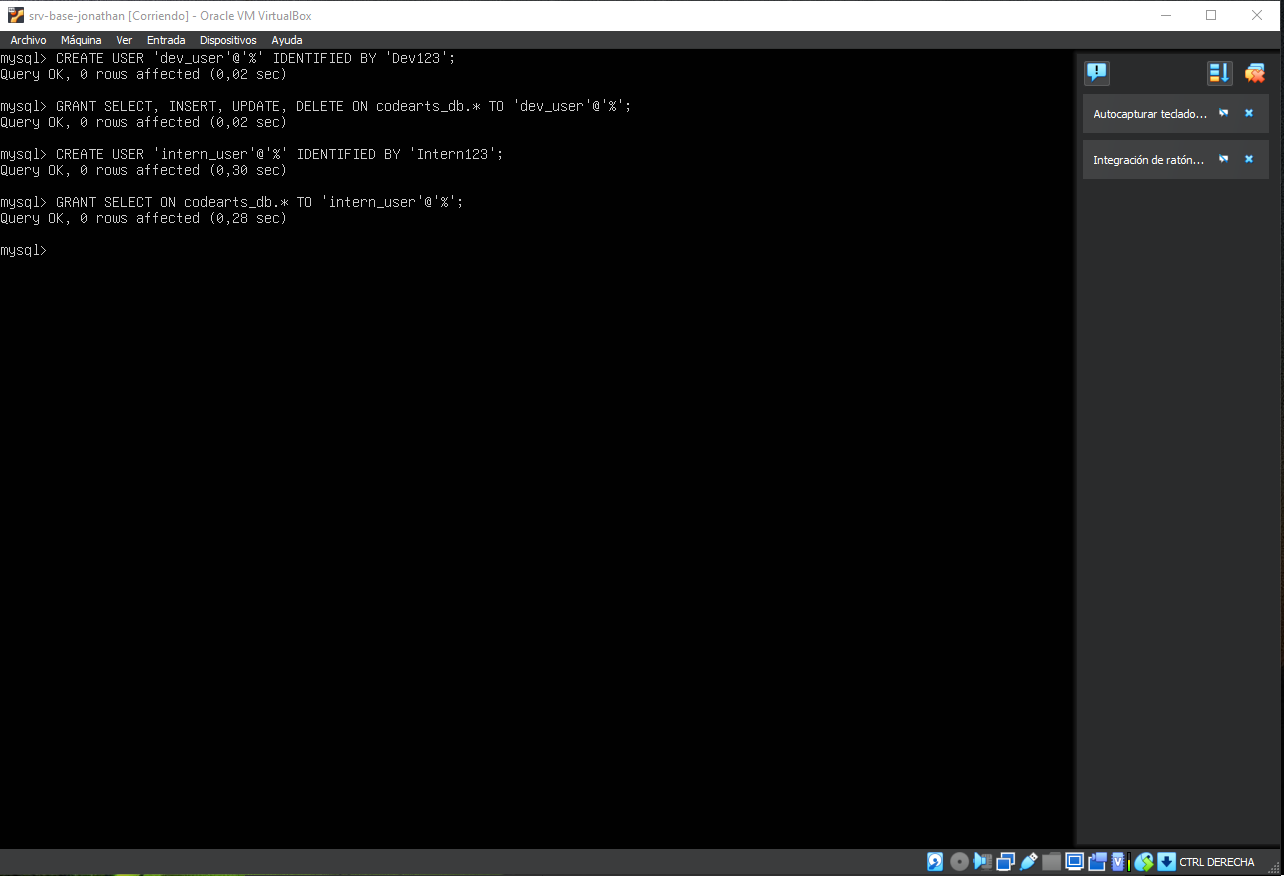
CREATE USER 'dev\_user'@'%' IDENTIFIED BY 'DevUser123';

GRANT SELECT, INSERT, UPDATE, DELETE ON codearts\_db.\* TO 'dev\_user'@'%';

CREATE USER 'intern\_user'@'%' IDENTIFIED BY 'InternUser123';

GRANT SELECT ON codearts\_db.\* TO 'intern\_user'@'%';

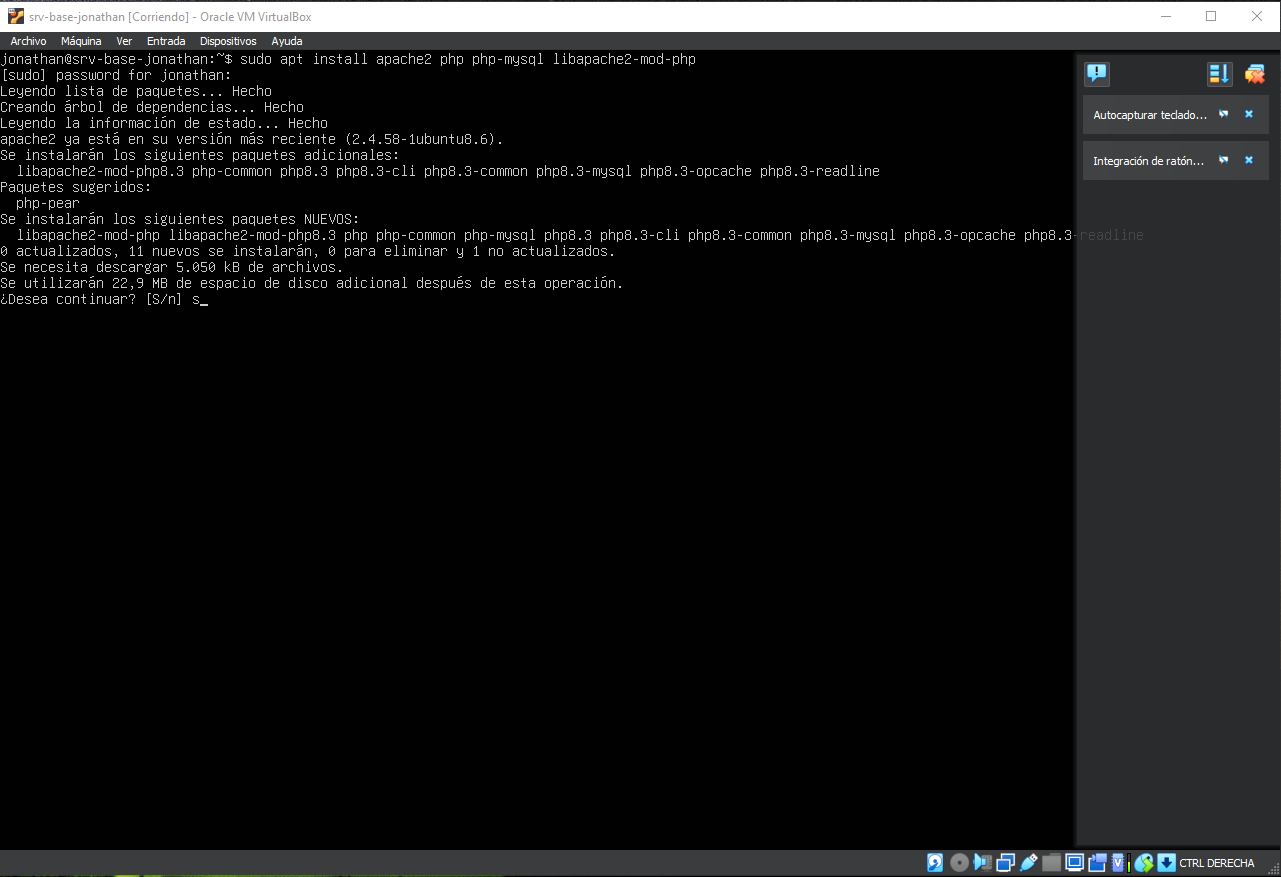
FLUSH PRIVILEGES;



# Fase 3: Conexión con la aplicación web

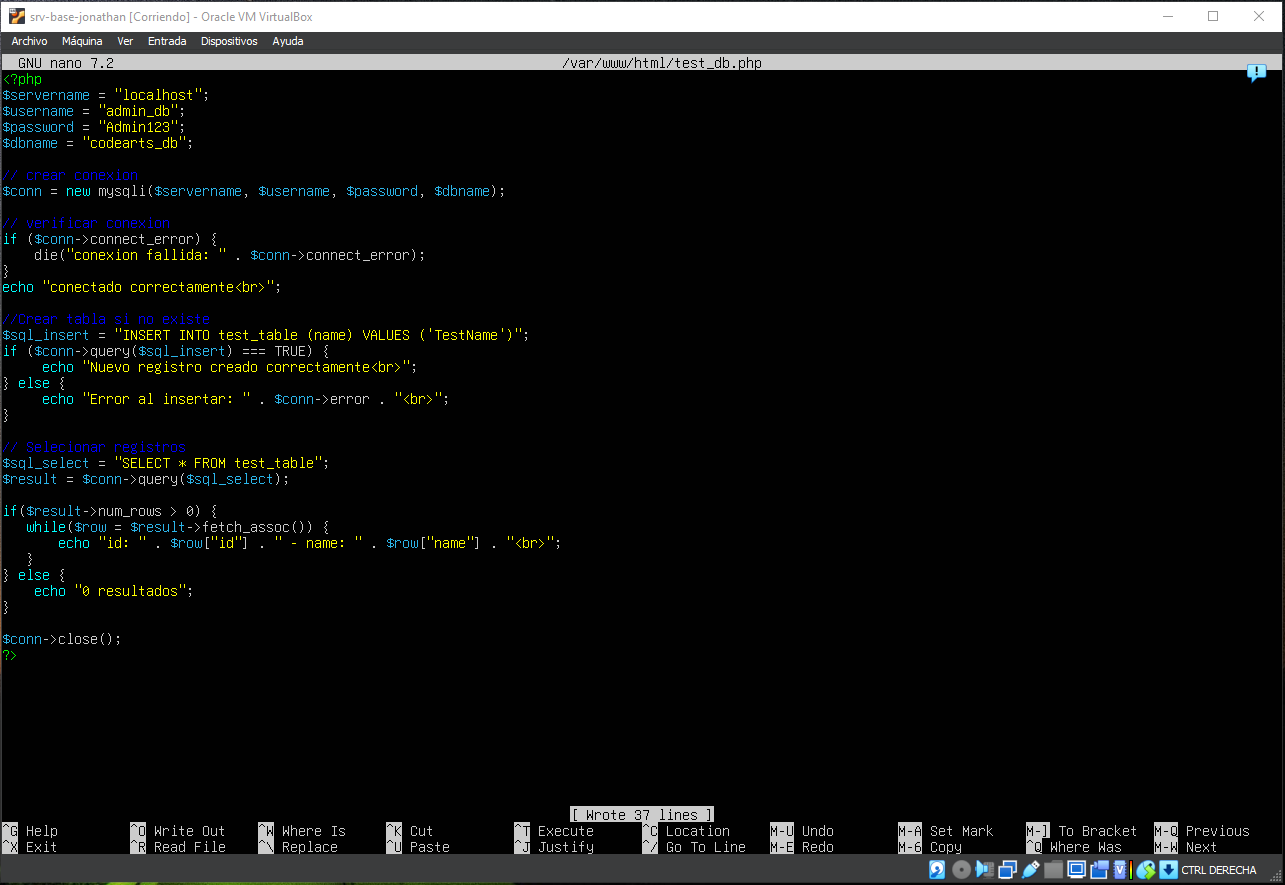
Instalaremos un servidor web con Apache/Ngnix y PHP/Python, en nuestro caso hemos elegido Apache con PHP:

sudo apt install apache2 php php-mysql libapache2-mod-php



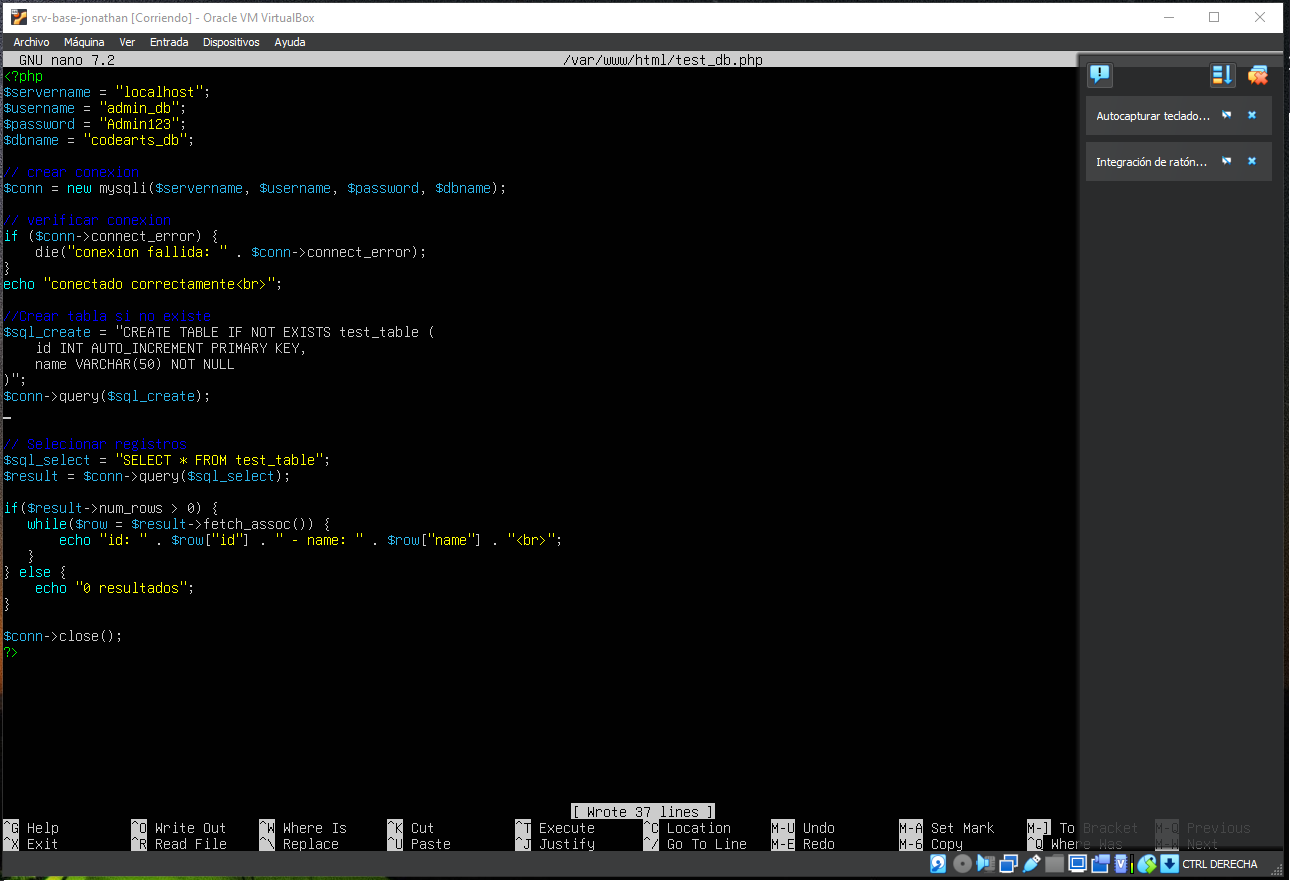
Creamos un script de conexión que permita leer y escribir datos en la base de datos que hemos creado:

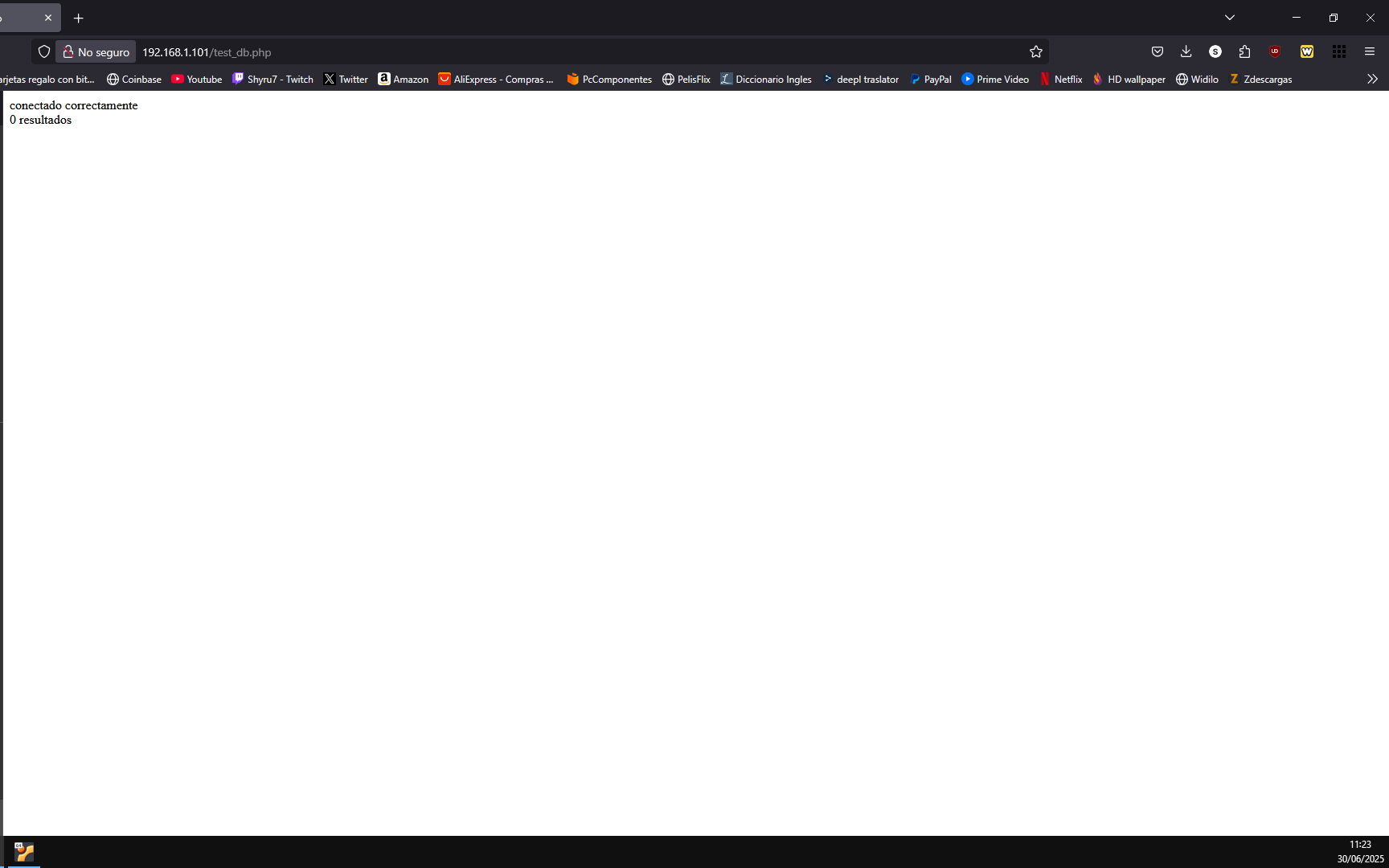
sudo nano /var/www/html/test\_db.php



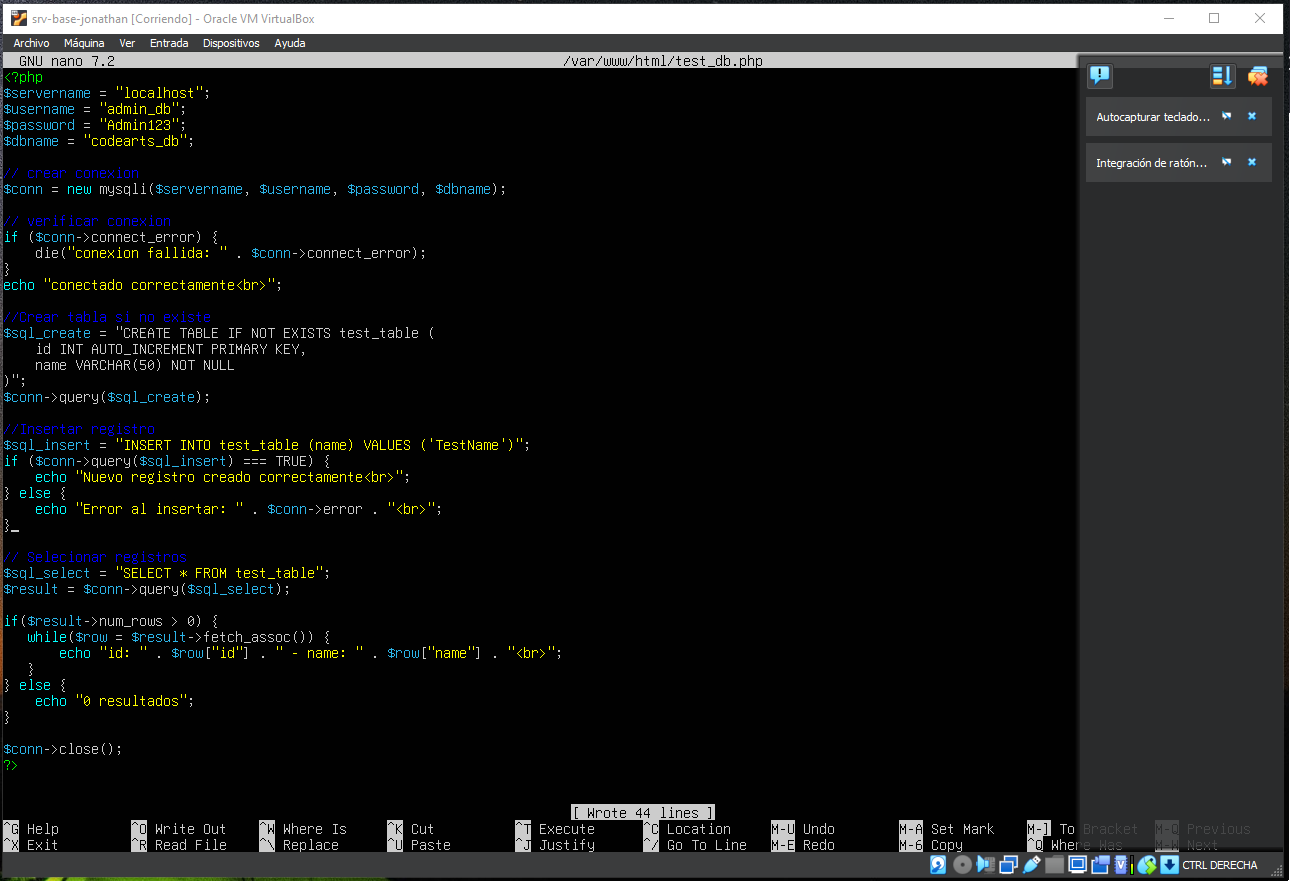
Realizamos unas pruebas con consultas SELECT, INSERT, UPDATE y DELETE

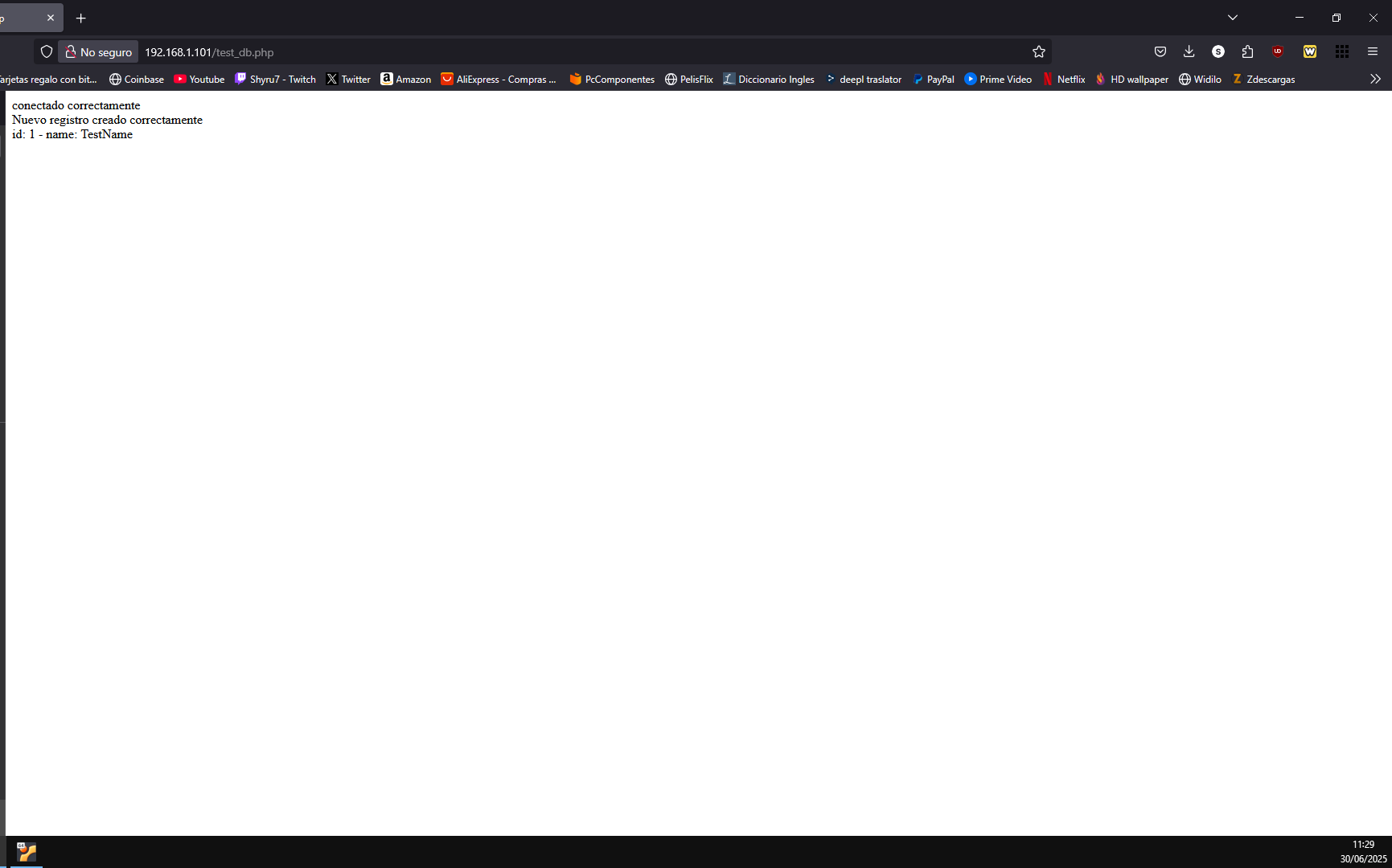
Prueba de tabla:



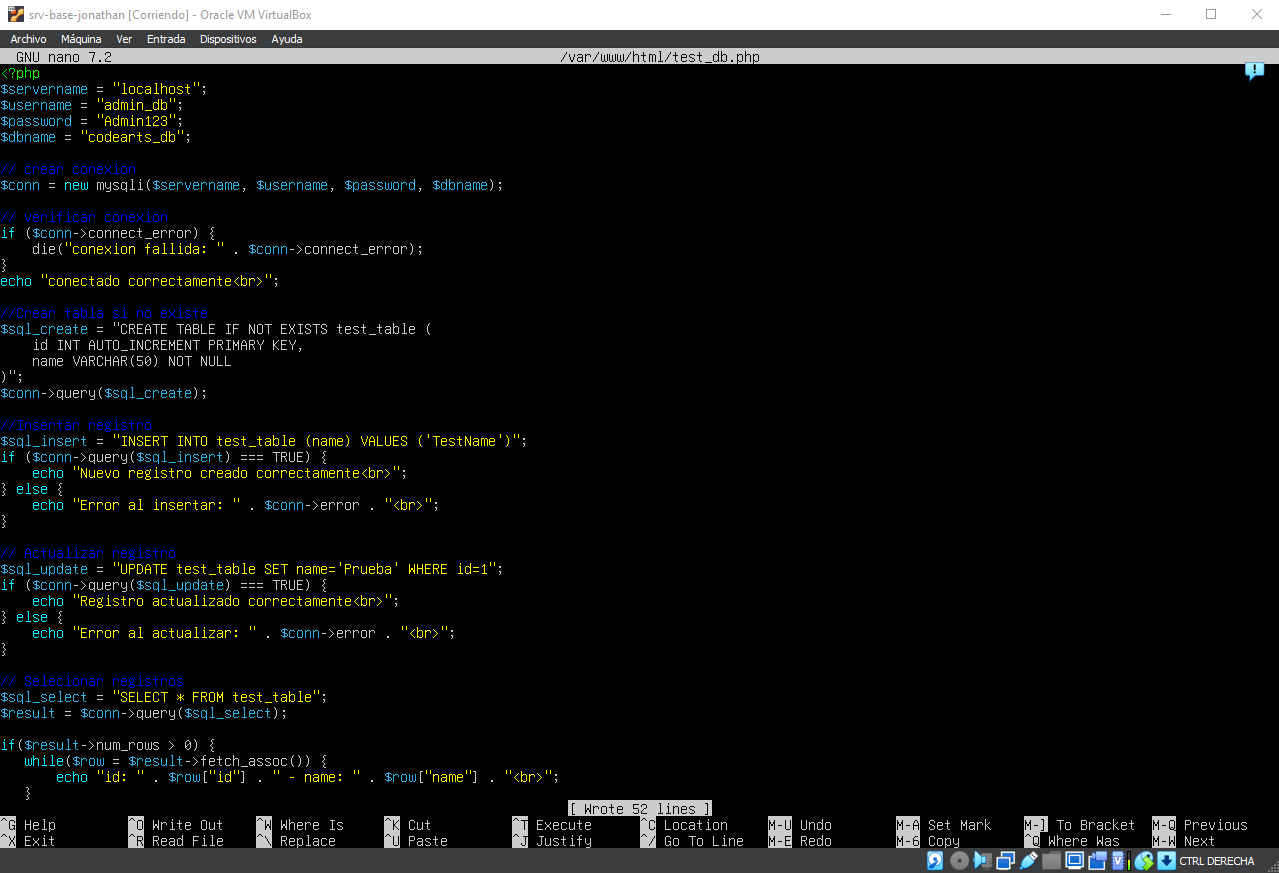


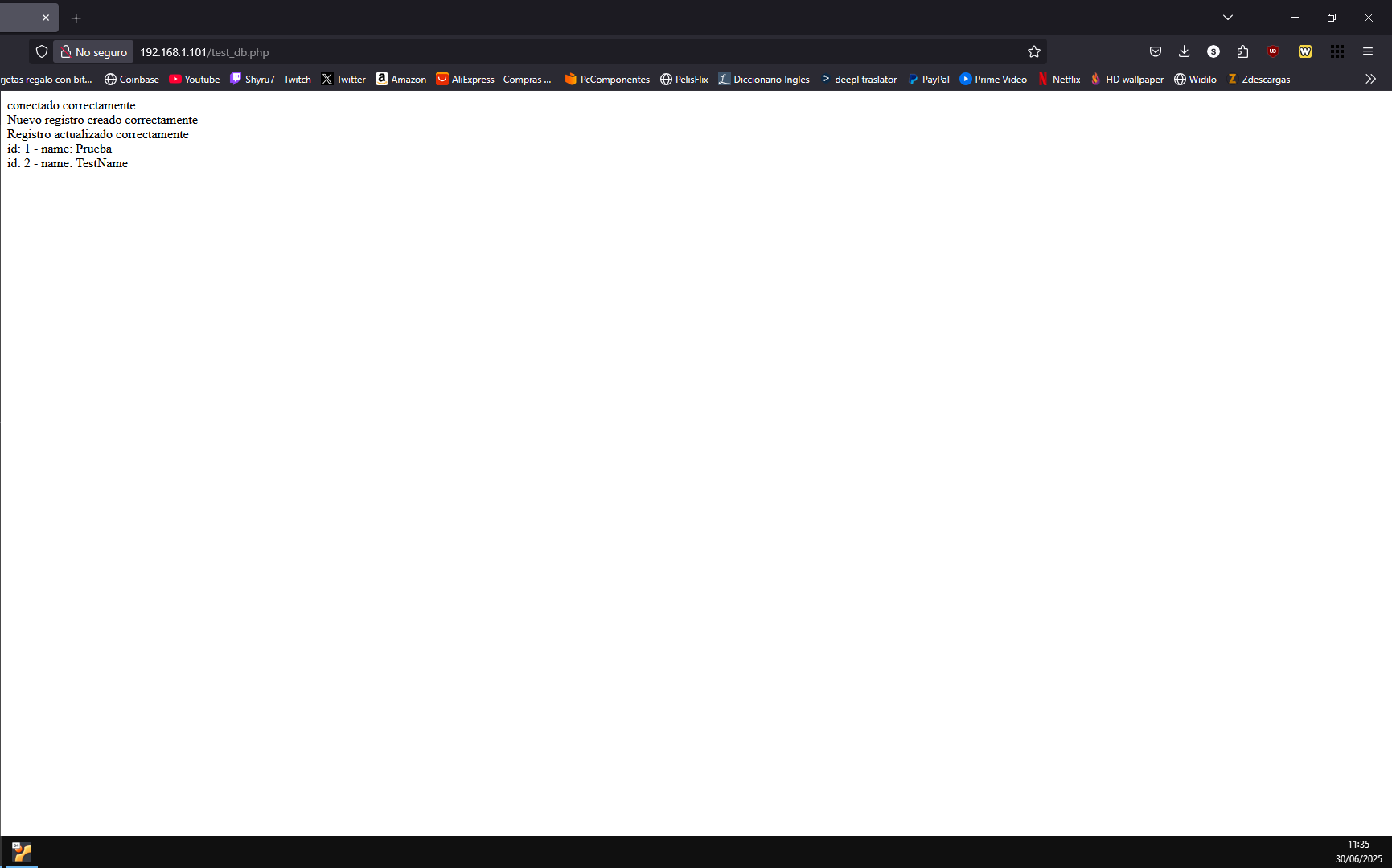
Insertando un registro:



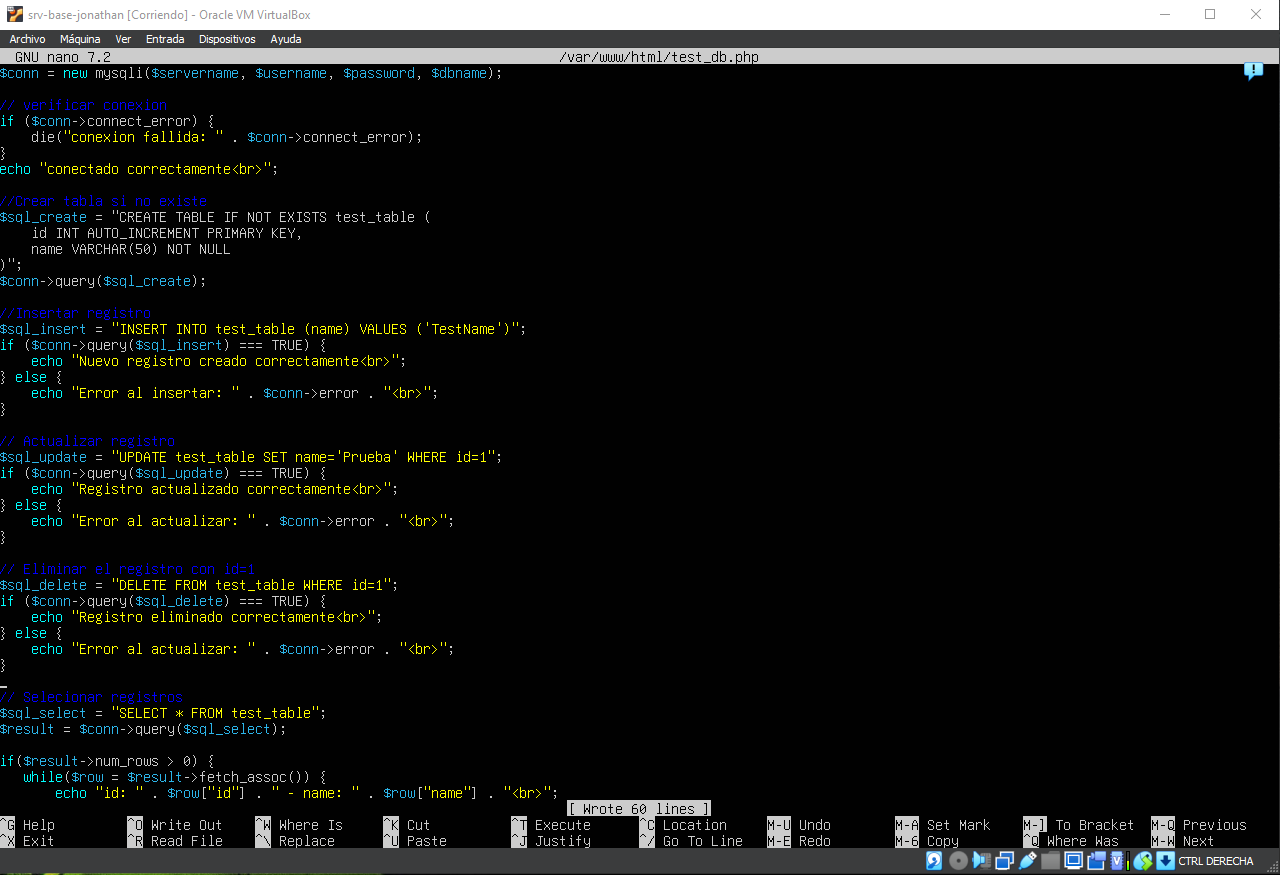


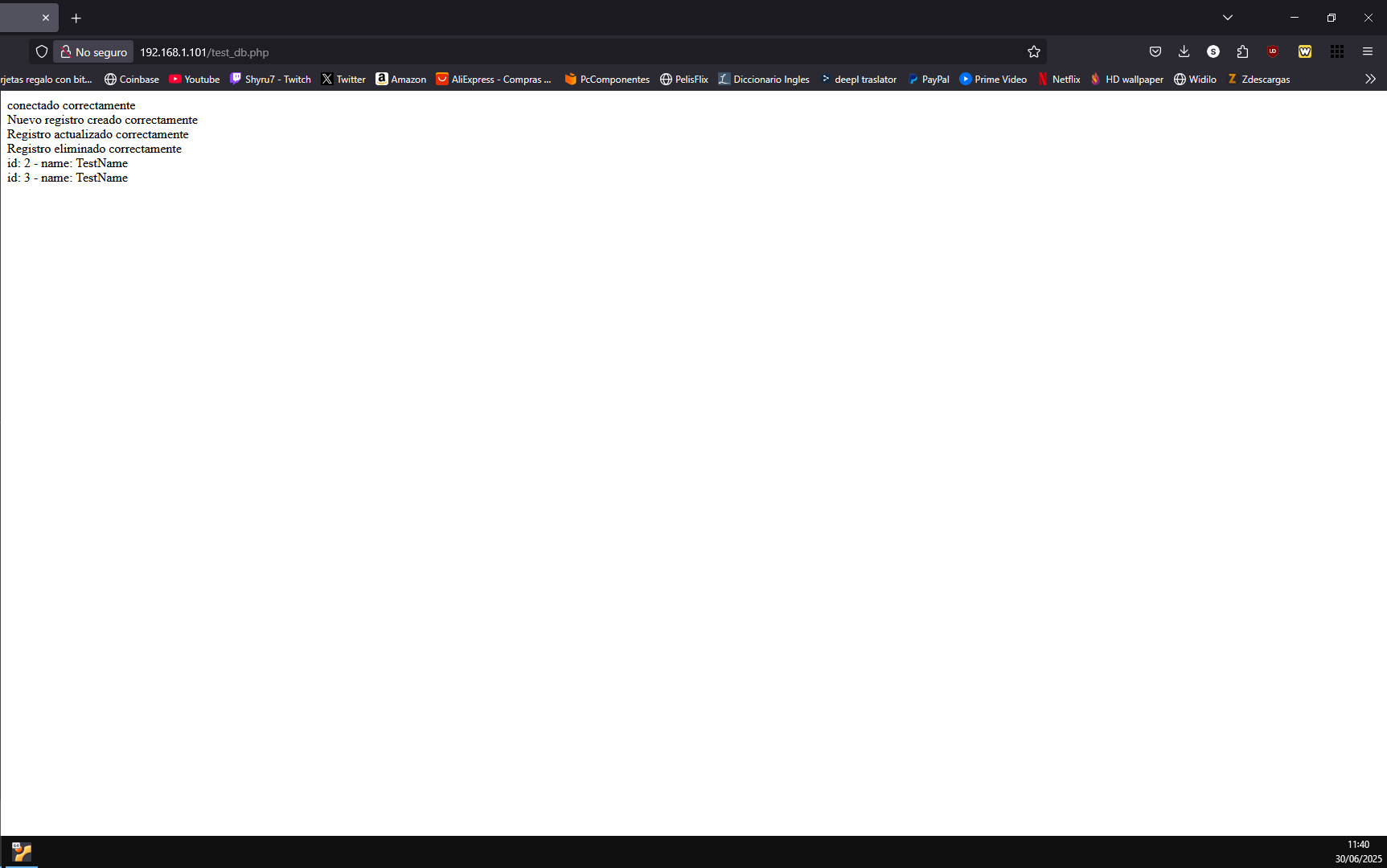
Actualizando un registro:





Eliminando un registro:

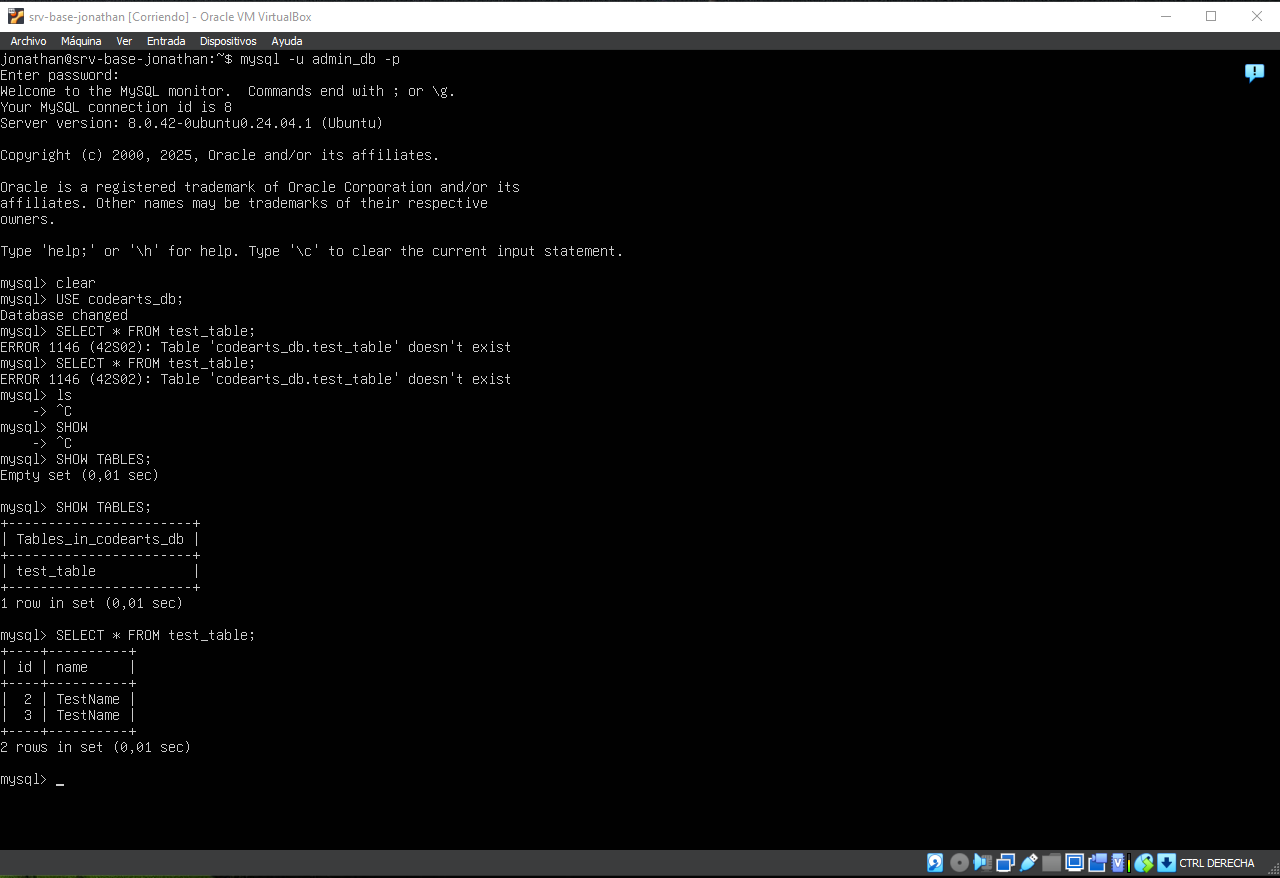




También podemos observar todos los cambios en mysql:

USE codearts\_db

SHOW TABLES



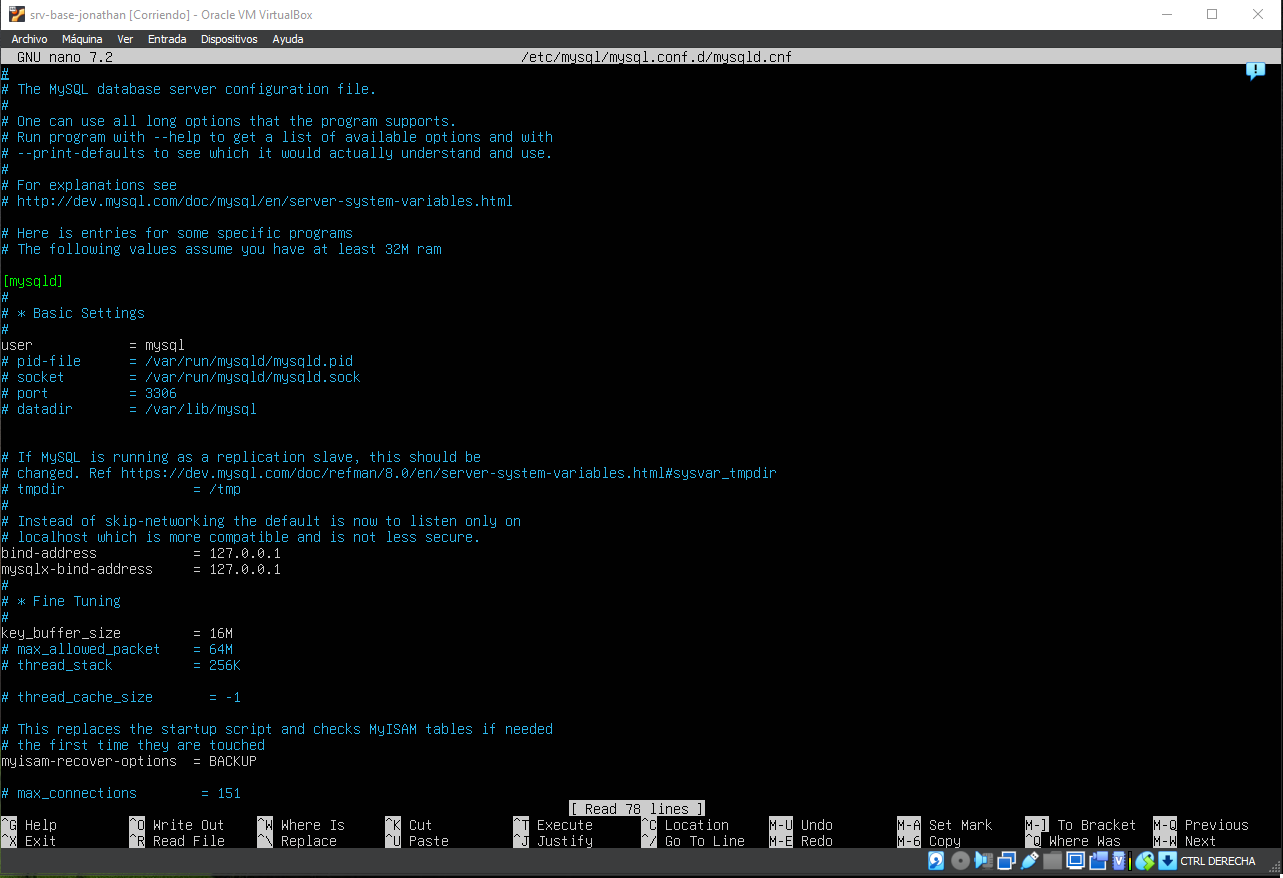
# Fase 4: Seguridad y respaldo

Configuramos un acceso seguro para restringir conexiones remotas (bind-adress):

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

Una vez dentro del documento cambiamos la linea donde nos indica:

bind-address = 127.0.0.1

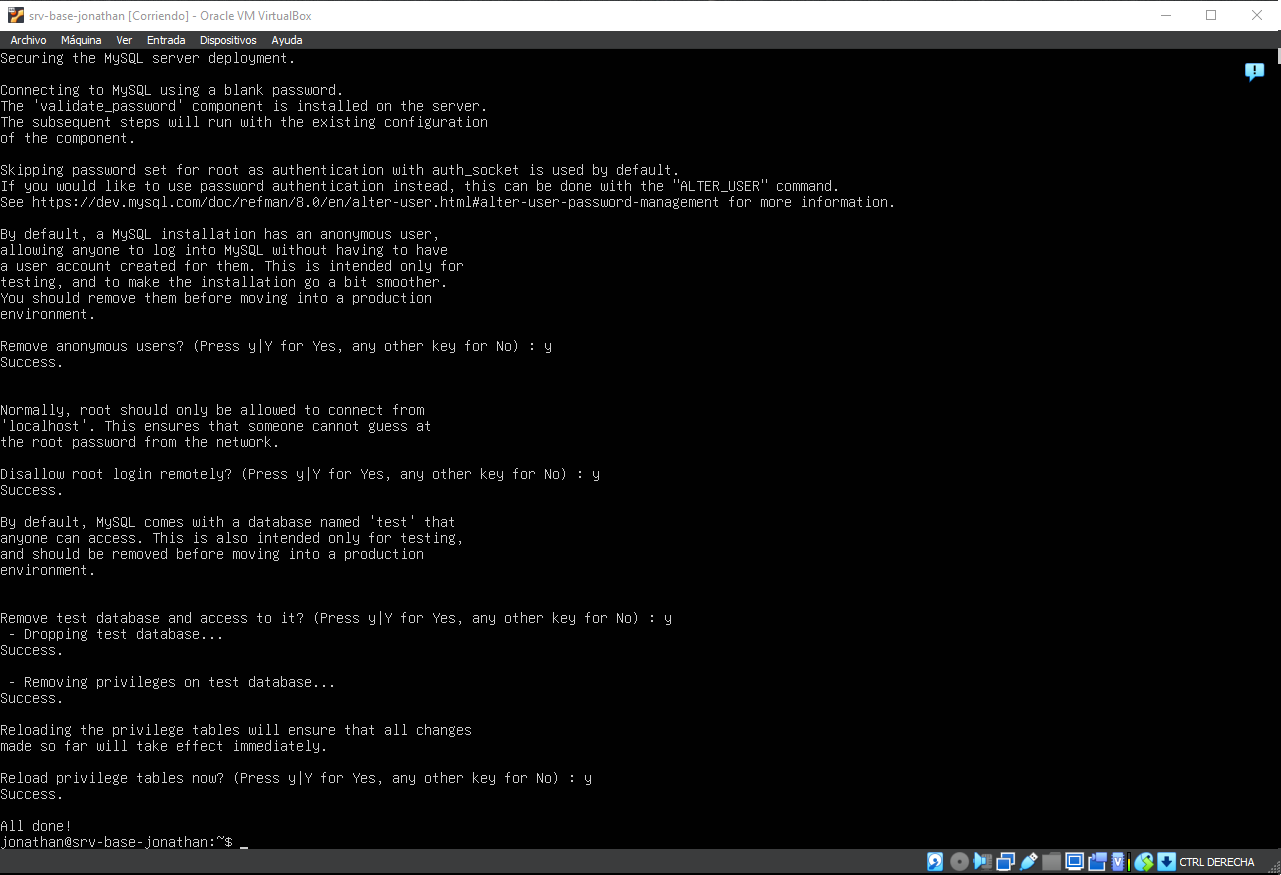


Reiniciamos mysql:

sudo systemctl restart mysql

Implementamos una autenticación segura con políticas de contraseña:

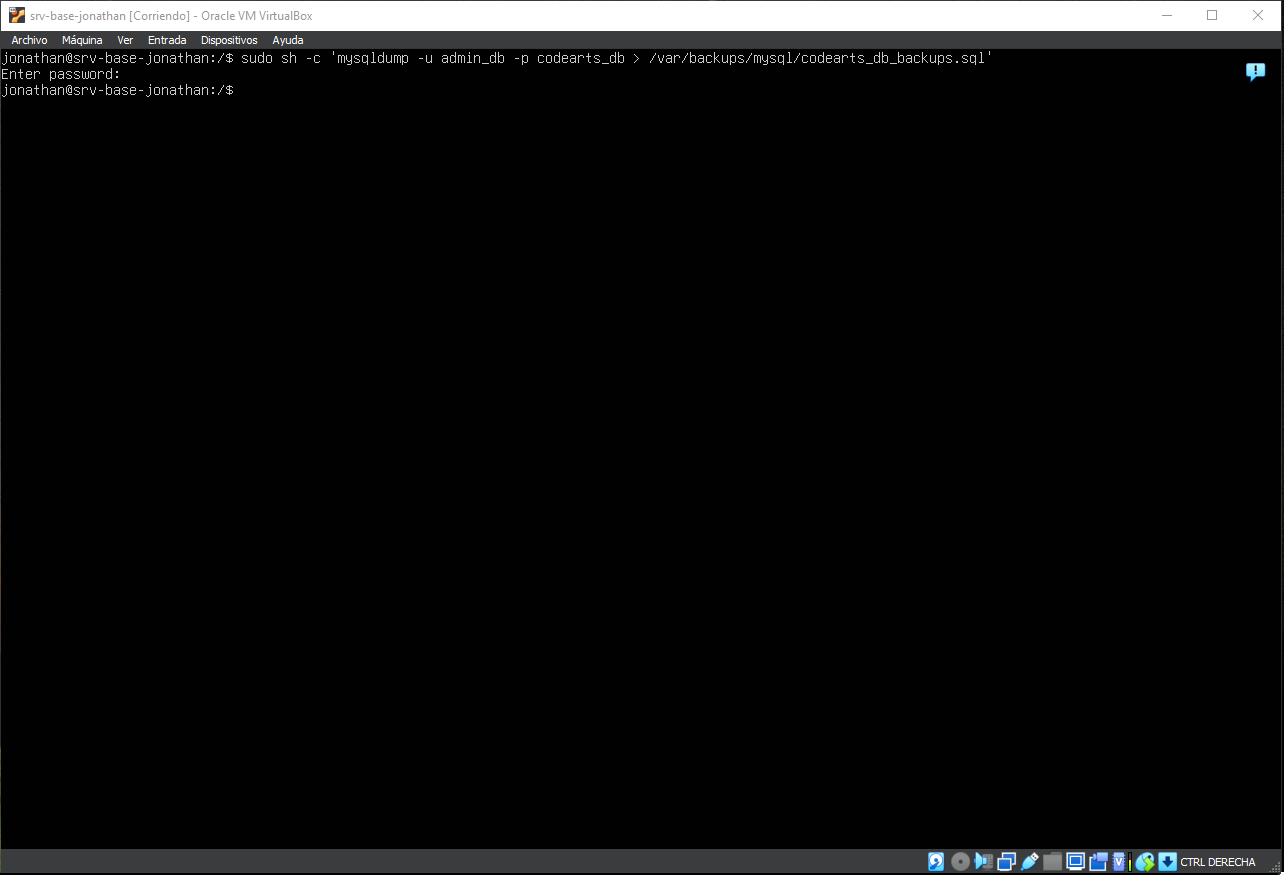
sudo mysql\_secure\_installation



Creamos un sistema de copias de seguridad automaticas con (mysqldump o pg\_dump)

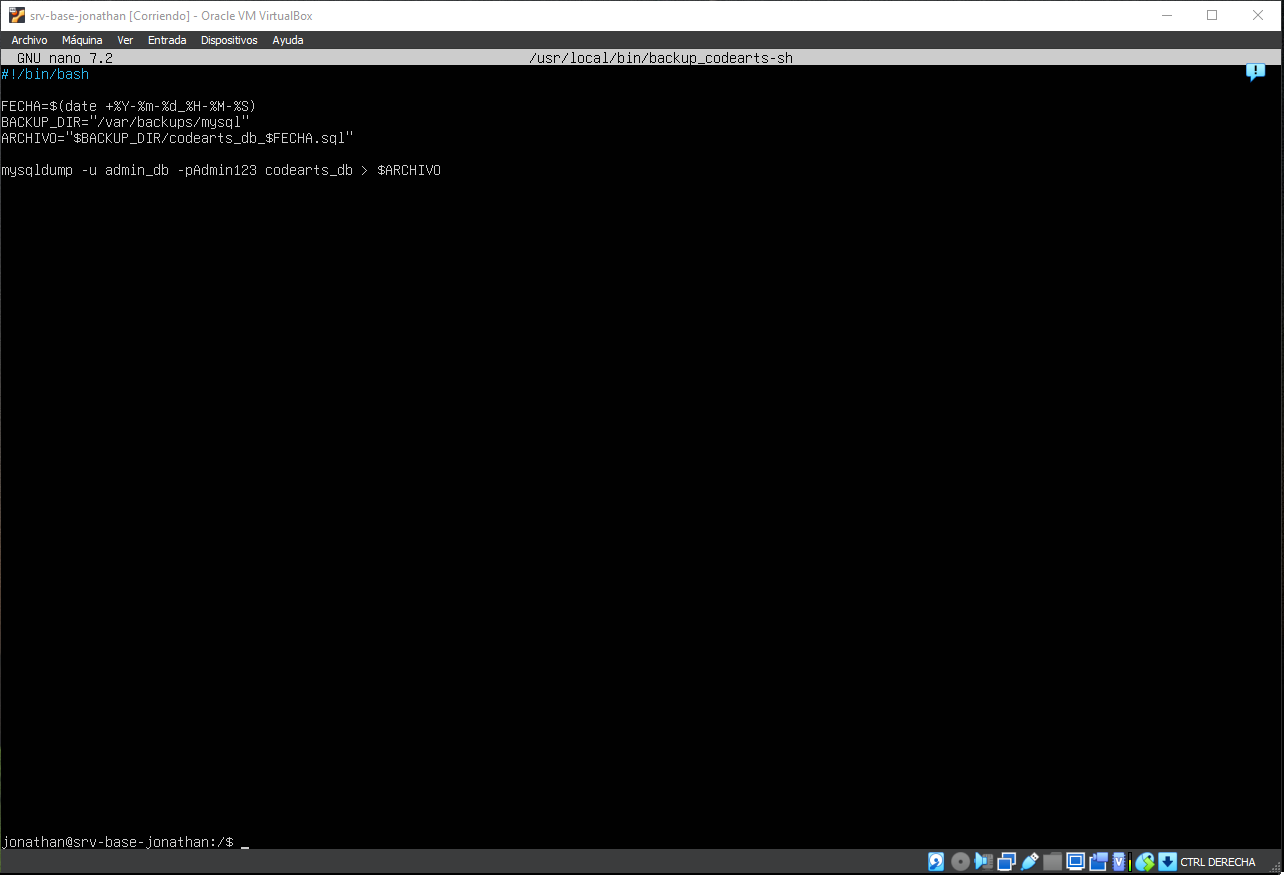
sudo sh -c ‘mysqldump -u admin\_db -p codearts\_db >

/var/backups/mysql/codearts\_db\_backups.sql’

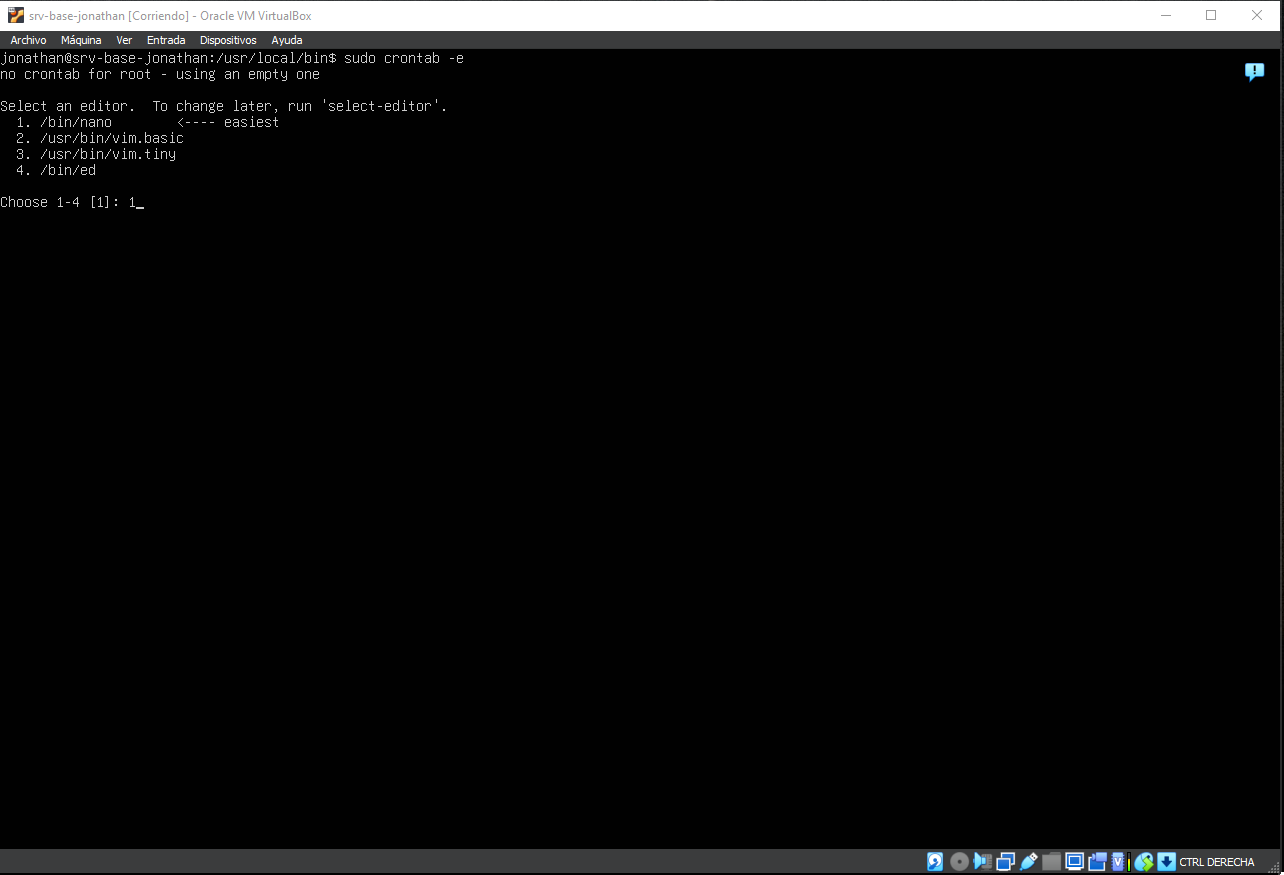


Creamos el script:

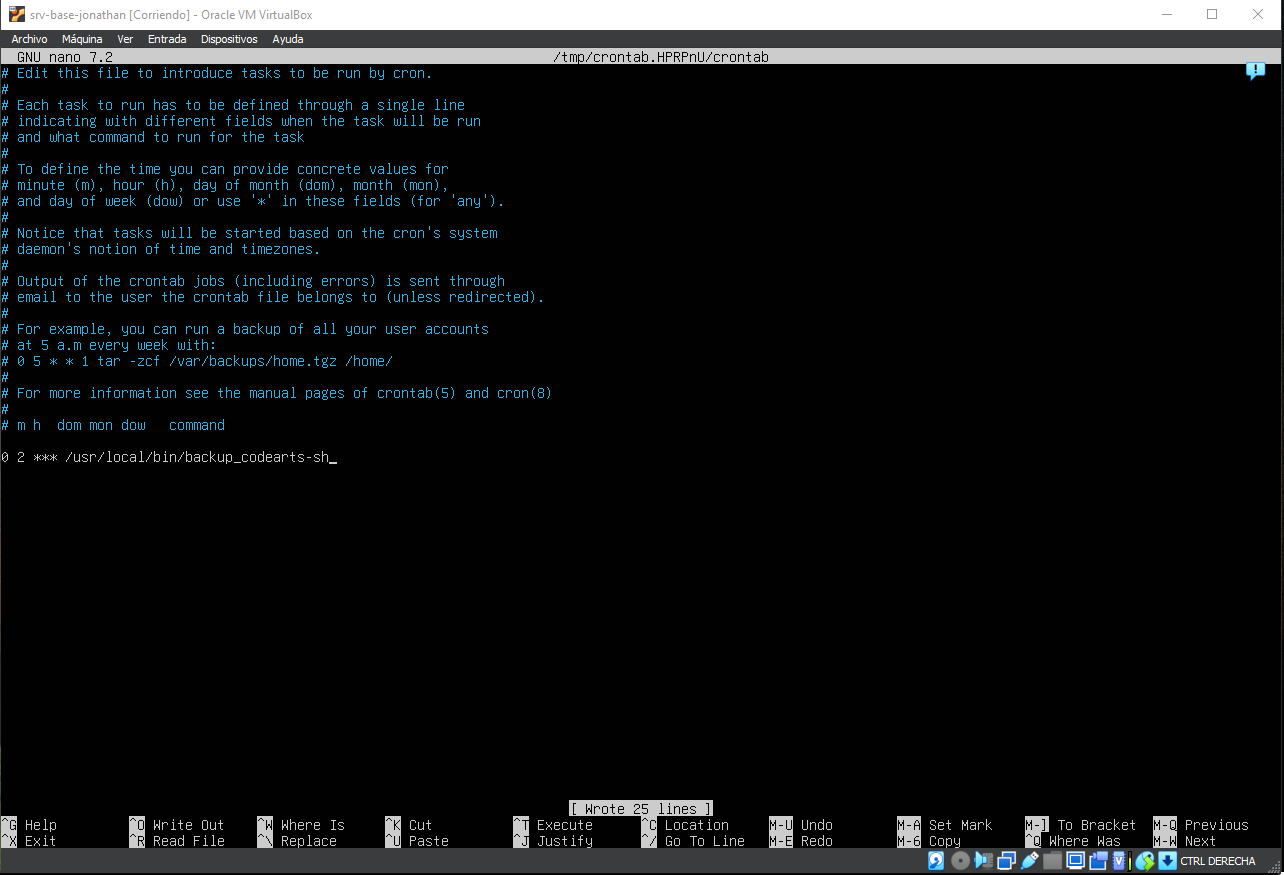
sudo nano /usr/local/bin/backup\_codearts.sh



Programamos con cron una tarea para backup diario:



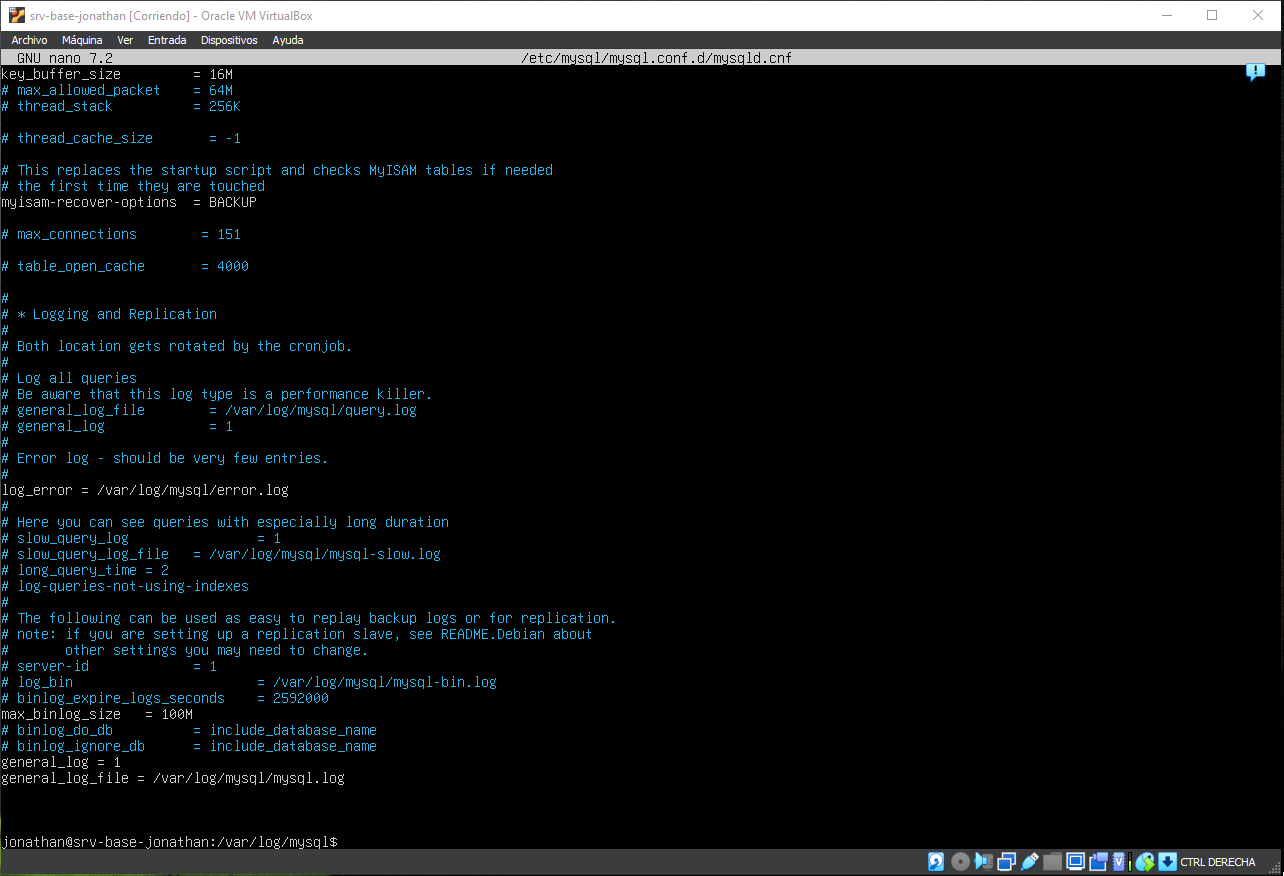
0 2 \* \* \* /usr/local/bin/backup\_codearts.sh



Aquí tuvimos un error de sintaxis y no funcionaba ya que los asteriscos no tenían separaciones.

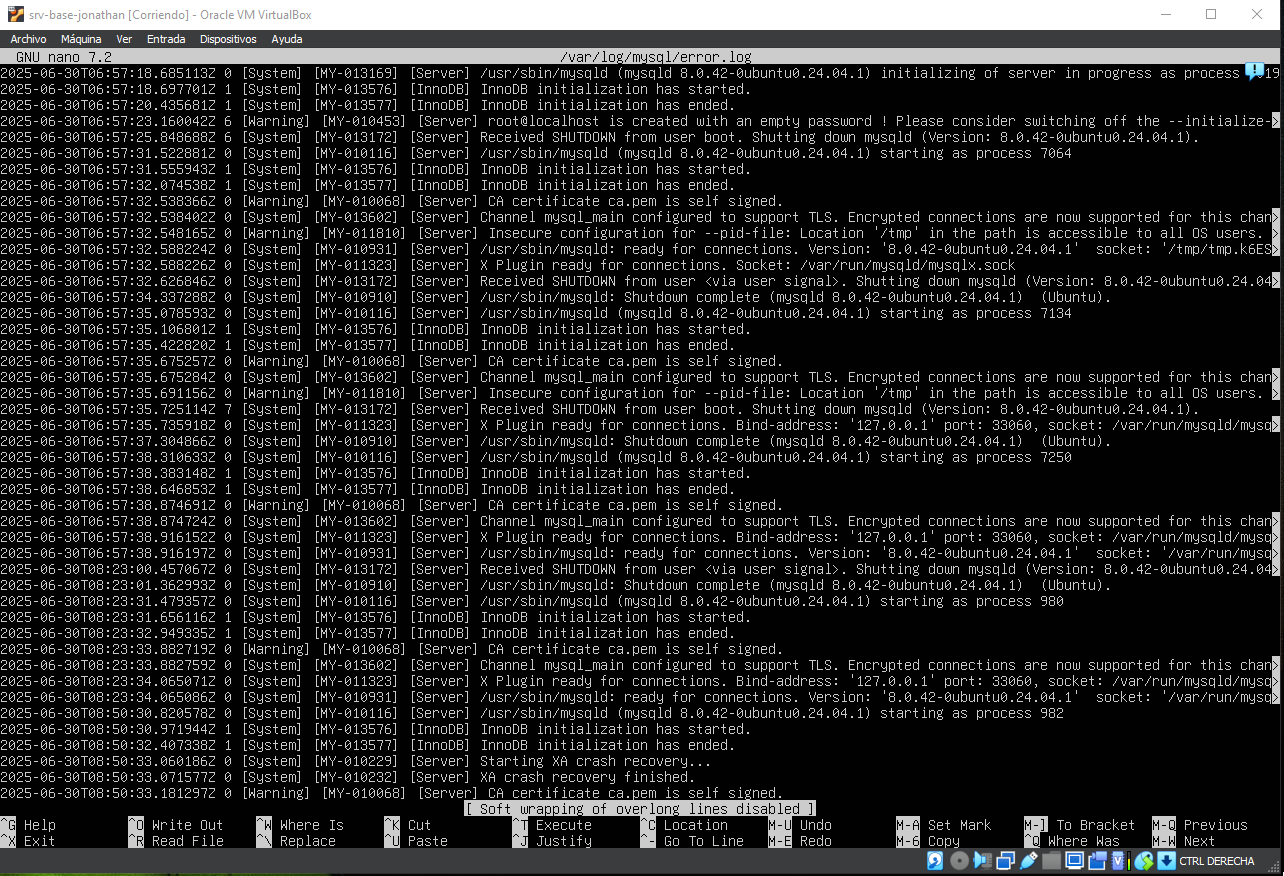
Podemos activar el log general de consultas es algo opcional pero nos dará un mayor volumen de información:

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf



Aquí podemos ver los logs:

sudo tail -f /var/log/mysql/mysql.log



Aqui mediante otra terminal podemos ver intentos fallidos de conexión de login por ejemplo con una contraseña equivocada:

sudo tail -f /var/log/mysql/mysql.log

