



PRACTICA 5 - BACKUP

ADRIÁN ALONSO RIDAO



2º ASIR
1ª EVALUACIÓN
CIBERSEGURIDAD

Contenido

| | |
|--|----|
| Practica 5.- BACKUP Y RECUPERACIÓN..... | 2 |
| 1. Terminal Ubuntu | 2 |
| 2. PhpMyAdmin | 10 |
| Diferencias Terminal Ubuntu y phpMyAdmin | 24 |
| Problemas resueltos | 25 |

Practica 5.- BACKUP Y RECUPERACIÓN.

1. Terminal Ubuntu

El objetivo de este módulo es aplicar los conocimientos dados en teoría poniéndolos en práctica.

1. Realiza una copia de una BBDD con mysqldump. Detén el servidor MySQL y elimina el contenido de la carpeta de datos. Restaura la copia.

Hacemos el backup de la base de datos world:

```
adrian@adrian-VirtualBox:~$ sudo mysqldump -u root -p world > world.sql
Enter password:
adrian@adrian-VirtualBox:~$ ls
Descargas  Escritorio  Música      Público  Videos
Documentos Imágenes    Plantillas  snap     world.sql
adrian@adrian-VirtualBox:~$
```

Al hacer un cat world.sql podemos ver el contenido de nuestra base de datos.

```
,1.6),('VNM','Tho','F',1.8),('VNM','Vietnamese','T',86.8),('VUT','Bislama','T',56.6),('VUT','English','T',28.3),('VUT','French','T',14.2),('WLF','Futuna','F',0.0),('WLF','Wallis','F',0.0),('WSM','English','T',0.6),('WSM','Samoan','T',47.5),('WSM','Samoan-English','F',52.0),('YEM','Arabic','T',99.6),('YEM','Soqutri','F',0.0),('YUG','Albaniana','F',16.5),('YUG','Hungarian','F',3.4),('YUG','Macedonian','F',0.5),('YUG','Romani','F',1.4),('YUG','Serbo-Croatian','T',75.2),('YUG','Slovak','F',0.7),('ZAF','Afrikaans','T',14.3),('ZAF','English','T',8.5),('ZAF','Ndebele','F',1.5),('ZAF','Northsotho','F',9.1),('ZAF','Southsotho','F',7.6),('ZAF','Swazi','F',2.5),('ZAF','Tsonga','F',4.3),('ZAF','Tswana','F',8.1),('ZAF','Venda','F',2.2),('ZAF','Xhosa','T',17.7),('ZAF','Zulu','T',22.7),('ZMB','Bemba','F',29.7),('ZMB','Chewa','F',5.7),('ZMB','Lozi','F',6.4),('ZMB','Nsenga','F',4.3),('ZMB','Nyanja','F',7.8),('ZMB','Tongan','F',11.0),('ZWE','English','T',2.2),('ZWE','Ndebele','F',16.2),('ZWE','Nyanja','F',2.2),('ZWE','Shona','F',72.1);
/*!40000 ALTER TABLE `countrylanguage` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

Paramos nuestro servicio de mysql:

```
adrian@adrian-VirtualBox:~$ sudo systemctl stop mysql
Warning: The unit file, source configuration file or drop-ins of mysql.service changed
on disk. Run 'systemctl daemon-reload' to reload units.
adrian@adrian-VirtualBox:~$ sudo systemctl status mysql
Warning: The unit file, source configuration file or drop-ins of mysql.service changed>
○ mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: inactive (dead) since Fri 2025-10-24 12:58:52 CEST; 13s ago
     Duration: 9min 33.469s
  Invocation: 10f47765d8314d3f9ec671dc2270558c
    Process: 4604 ExecStart=/usr/sbin/mysqld (code=exited, status=0/SUCCESS)
   Main PID: 4604 (code=exited, status=0/SUCCESS)
    Status: "Server shutdown complete (with return value = 0)"
  Mem peak: 449.9M
    CPU: 11.546s

oct 24 12:49:14 adrian-VirtualBox systemd[1]: Starting mysql.service - MySQL Community>
oct 24 12:49:18 adrian-VirtualBox systemd[1]: Started mysql.service - MySQL Community >
oct 24 12:58:51 adrian-VirtualBox systemd[1]: Stopping mysql.service - MySQL Community>
oct 24 12:58:52 adrian-VirtualBox systemd[1]: mysql.service: Deactivated successfully.
oct 24 12:58:52 adrian-VirtualBox systemd[1]: Stopped mysql.service - MySQL Community >
oct 24 12:58:52 adrian-VirtualBox systemd[1]: mysql.service: Consumed 11.546s CPU time>
[lines 1-18/18 (END)]
```

Borramos la carpeta world de la ruta /var/lib/mysql:

```
adrian@adrian-VirtualBox:~$ sudo rm -r /var/lib/mysql/world
adrian@adrian-VirtualBox:~$ sudo ls /var/lib/mysql
auto.cnf          debian-5.7.flag  mysql            server-key.pem
binlog.000001     '#ib_16384_0.dblwr'  mysql.ibd        sys
binlog.index      '#ib_16384_1.dblwr'  mysql_upgrade_history  undo_001
ca-key.pem        ib_buffer_pool     performance_schema  undo_002
ca.pem            ibdata1            private_key.pem
client-cert.pem   '#innodb_redo'      public_key.pem
client-key.pem    '#innodb_temp'      server-cert.pem
adrian@adrian-VirtualBox:~$
```

Activamos el estado de mysql:

```
adrian@adrian-VirtualBox:~$ sudo systemctl start mysql
Warning: The unit file, source configuration file or drop-ins of mysql.service changed
on disk. Run 'systemctl daemon-reload' to reload units.
^[[A^[[A^[[A^[[A
adrian@adrian-VirtualBox:~$ sudo systemctl status mysql
Warning: The unit file, source configuration file or drop-ins of mysql.service changed>
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-24 13:06:42 CEST; 6s ago
     Invocation: d0717b0c489245b69a9323daa96aa40f
    Process: 5749 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, >
   Main PID: 5758 (mysqld)
    Status: "Server is operational"
     Tasks: 35 (limit: 3866)
    Memory: 422.5M (peak: 436.5M)
       CPU: 9.606s
    CGroup: /system.slice/mysql.service
            └─5758 /usr/sbin/mysqld

oct 24 13:06:36 adrian-VirtualBox systemd[1]: Starting mysql.service - MySQL Community>
oct 24 13:06:42 adrian-VirtualBox systemd[1]: Started mysql.service - MySQL Community >
```

E importamos nuestro backup de world que hemos realizado antes:

```
adrian@adrian-VirtualBox:~$ sudo mysql -u root -p world < world.sql
Enter password:
adrian@adrian-VirtualBox:~$
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| world |
+-----+
5 rows in set (0,01 sec)
```

2. Copia de seguridad Incremental:

a. Activa el registro binario de tu servidor MySQL.

Paramos el servicio MySQL:

```
adrian@adrian-VirtualBox:~$ sudo systemctl stop mysql
Warning: The unit file, source configuration file or drop-ins of mysql.service changed
on disk. Run 'systemctl daemon-reload' to reload units.
adrian@adrian-VirtualBox:~$ sudo systemctl status mysql
Warning: The unit file, source configuration file or drop-ins of mysql.service changed
○ mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: inactive (dead) since Fri 2025-10-24 13:21:50 CEST; 20s ago
     Duration: 15min 7.145s
  Invocation: d0717b0c489245b69a9323daa96aa40f
    Process: 5749 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, >
    Process: 5758 ExecStart=/usr/sbin/mysqld (code=exited, status=0/SUCCESS)
   Main PID: 5758 (code=exited, status=0/SUCCESS)
    Status: "Server shutdown complete (with return value = 0)"
  Mem peak: 446.9M
    CPU: 19.895s

oct 24 13:06:36 adrian-VirtualBox systemd[1]: Starting mysql.service - MySQL Community>
oct 24 13:06:42 adrian-VirtualBox systemd[1]: Started mysql.service - MySQL Community >
oct 24 13:21:50 adrian-VirtualBox systemd[1]: Stopping mysql.service - MySQL Community>
oct 24 13:21:50 adrian-VirtualBox systemd[1]: mysql.service: Deactivated successfully.
oct 24 13:21:50 adrian-VirtualBox systemd[1]: Stopped mysql.service - MySQL Community >
oct 24 13:21:50 adrian-VirtualBox systemd[1]: mysql.service: Consumed 19.895s CPU time>
```

Descomentamos la línea del archivo que encontramos en

“/etc/mysql/mysql.conf.d/mysqld.cnf”:

```
adrian@adrian-VirtualBox: ~
GNU nano 8.3 /etc/mysql/mysql.conf.d/mysqld.cnf *

# max_connections      = 151
# table_open_cache     = 4000
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
#
# Log all queries
# Be aware that this log type is a performance killer.
# general_log_file      = /var/log/mysql/query.log
# general_log           = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
# slow_query_log        = 1
# slow_query_log_file   = /var/log/mysql/mysql-slow.log
# long_query_time       = 2
# log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
# other settings you may need to change.
# server-id             = 1
log_bin                = /var/log/mysql/mysql-bin.log
# binlog_expire_logs_seconds = 2592000
max_binlog_size        = 100M
# binlog_do_db          = include_database_name
# binlog_ignore_db      = include_database_name

^G Ayuda      ^O Guardar    ^F Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar      ^J Justificar ^/ Ir a línea
```

b. Realiza una copia de seguridad completa

Volvemos a iniciar nuestro estado de MySQL:

```
adrian@adrian-VirtualBox:~$ sudo systemctl start mysql
Warning: The unit file, source configuration file or drop-ins of mysql.service changed
on disk. Run 'systemctl daemon-reload' to reload units.
^[[A^[[Aadrian@adrian-VirtualBox:~$ sudo systemctl status mysql
Warning: The unit file, source configuration file or drop-ins of mysql.service changed>
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-24 13:28:39 CEST; 7s ago
 Invocation: 1a19502fba4d46ed9ec98bec222aedc3
   Process: 6031 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, >
  Main PID: 6040 (mysqld)
    Status: "Server is operational"
     Tasks: 35 (limit: 3866)
  Memory: 422.7M (peak: 435.8M)
     CPU: 7.279s
    CGroup: /system.slice/mysql.service
            └─6040 /usr/sbin/mysqld
```

Y hacemos una copia de world:

```
adrian@adrian-VirtualBox:~$ sudo mysqldump -u root -p world > ./backup.sql
Enter password:
```

c. Haz varias operaciones de modificación de datos en la Base de datos.

He creado una nueva tabla llamada 'Cultura':

```
mysql> CREATE TABLE cultura(
->     Fiesta_Nacional VARCHAR(20) PRIMARY KEY,
->     Dia int (2),
->     Mes VARCHAR (20),
->     DescripciOn VARCHAR(200),
->     Country_Code CHAR(3) REFERENCES country (code)
-> );
Query OK, 0 rows affected, 1 warning (0,06 sec)

mysql> show tables
-> ;
+-----+
| Tables_in_world |
+-----+
| city             |
| country          |
| countrylanguage |
| cultura          |
+-----+
4 rows in set (0,00 sec)
```

d. Elimina ahora todos los datos

Eliminamos los datos de la base de datos world:

```
mysql> drop table city;
Query OK, 0 rows affected (0,05 sec)

mysql> drop table country;
ERROR 3730 (HY000): Cannot drop table 'country' referenced by a foreign key constraint
'countryLanguage_ibfk_1' on table 'countrylanguage'.
mysql> drop table countrylanguage;
Query OK, 0 rows affected (0,05 sec)

mysql> drop table cultura;
Query OK, 0 rows affected (0,03 sec)

mysql> drop table country;
Query OK, 0 rows affected (0,04 sec)

mysql> show tables;
Empty set (0,00 sec)
```

e. Restaura todos los datos de una manera adecuada.

Restauramos el backup en world, la 000002 nos restaurará cultura:

```
adrian@adrian-VirtualBox:~$ sudo mysqlbinlog /var/log/mysql/mysql-bin.000002 | sudo mysql -u root -p world
```

```
Database changed
mysql> show tables;
+-----+
| Tables_in_world |
+-----+
| cultura          |
+-----+
1 row in set (0,00 sec)

mysql> █
```


Y la 000004 nos restaura el resto de bases de datos:

```
Database changed
mysql> show tables;
+-----+
| Tables_in_world |
+-----+
| city             |
| country          |
| countrylanguage  |
| cultura          |
+-----+
4 rows in set (0,00 sec)
```

3. **Copia de Seguridad periódica. Configura en Ubuntu (con Cron) o Windows (tarea programada o directiva de tarea) para realizar copias de seguridad periódicas cada mes. Puedes hacer un script para guardar de alguna manera un backup lo puedes realizar en Ubuntu o Windows. Te dejo un ejemplo de Linux.**

Creamos un archivo de configuración para root dado que root ejecutará el script.

```
adrian@adrian-VirtualBox:~$ sudo nano /root/.my.cnf
```

Configuramos el archivo para que quede del siguiente modo:

```
GNU nano 8.3
[mysqldump]
user=root
password=rootroot
```

Protegemos el archivo para que solo **root** pueda leerlo:

```
adrian@adrian-VirtualBox:~$ sudo chmod 600 /root/.my.cnf
```

Hemos realizado estas operaciones para que cron pueda ejecutar mysqldump sin que nadie escriba la contraseña de modo seguro.

Vamos a crear el script que hará el trabajo. Creamos un directorio para guardar los backups:

```
adrian@adrian-VirtualBox:~$ sudo mkdir -p /var/backups/mysql
```

Creamos el archivo del script:

```
adrian@adrian-VirtualBox:~$ sudo nano /usr/local/bin/backup_mensual_world.sh

GNU nano 8.3 /usr/local/bin/backup_mensual_world.sh
#!/bin/bash

DBDIR="/var/backups/mysql"
DB="world"
FILENAME="world_backup"
DIA=$(date +%Y-%m-%d)

echo "Iniciando backup de $DB..."
mysqldump --opt $DB > $DBDIR/$DB'_'$DIA.sql
cd $DBDIR
tar -cjvf $FILENAME'_'$DIA'.tar.bz2' $DB'_'$DIA'.sql' 2> /tmp/world_backup.log > /dev/null
mv $DBDIR/$DB'_'$DIA'.sql'
echo "Backup de $DB completado y comprimido en $DBDIR/$FILENAME'_'$DIA'.tar.bz2"
```

Hacemos que nuestro script sea ejecutable:

```
adrian@adrian-VirtualBox:~$ sudo chmod +x /usr/local/bin/backup_mensual_world.sh
```

Vamos a probar nuestro script, lo ejecutamos con:

```
adrian@adrian-VirtualBox:~$ sudo /usr/local/bin/backup_mensual_world.sh
Iniciando backup de world...
Backup de world completado y comprimido en /var/backups/mysql/world_backup'_'2025-11-03.tar.bz2
```

Si vemos los mensajes de “Iniciando backup...” y “Backup... completado” es que se está ejecutando correctamente.

Comprobamos el resultado:

```
adrian@adrian-VirtualBox:~$ ls -lh /var/backups/mysql
total 76K
-rw-r--r-- 1 root root 73K nov  3 17:38 world_backup_2025-11-03.tar.bz2
```

Vemos que se ha creado el archivo nuevo comprimido.

Ahora vamos a decirle a cron que ejecute el script cada mes, para ello abrimos el editor de cron para el usuario root:

```
adrian@adrian-VirtualBox:~$ sudo crontab -e
```

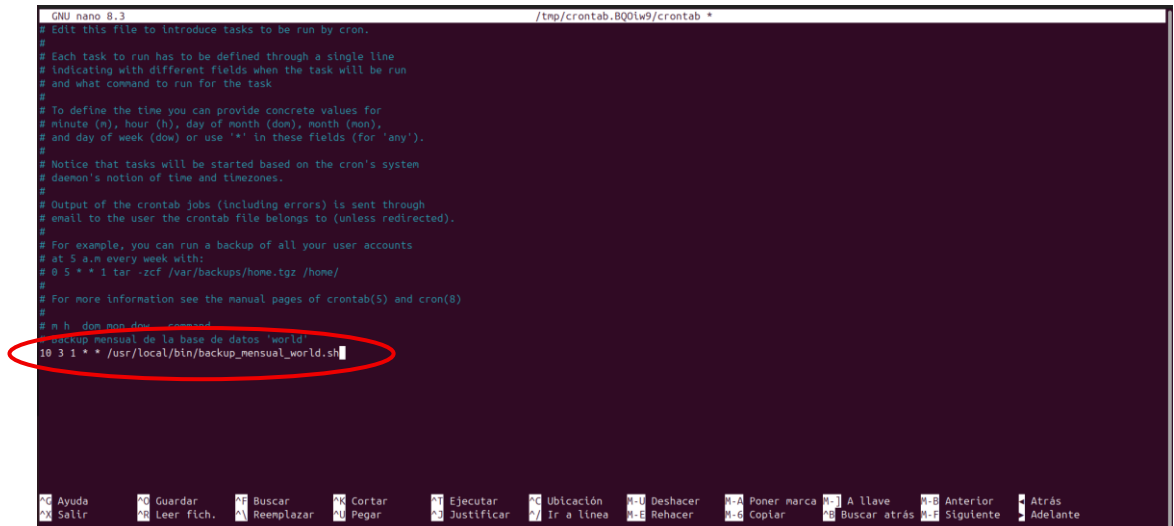
Puede que nos habrá con el antiguo editor Vim, para usar nano usaremos el siguiente comando:

```
adrian@adrian-VirtualBox:~$ sudo EDITOR=nano crontab -e
```

Añadimos en la ultima línea # **Backup mensual de la base de datos 'world'**

10 3 1 * * /usr/local/bin/backup_mensual_world.sh

Esto significa que en el minuto 10 a las 3 de la mañana el día 1 de cada mes de cualquier mes de cualquier día de la semana ejecute el comando /usr/local/bin/backup_mensual_world.sh



```
GNU nano 8.3 /tmp/crontab.BQ0Lw9/crontab *
# 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 dow dow command
# backup mensual de la base de datos 'world'
10 3 1 * * /usr/local/bin/backup_mensual_world.sh
```

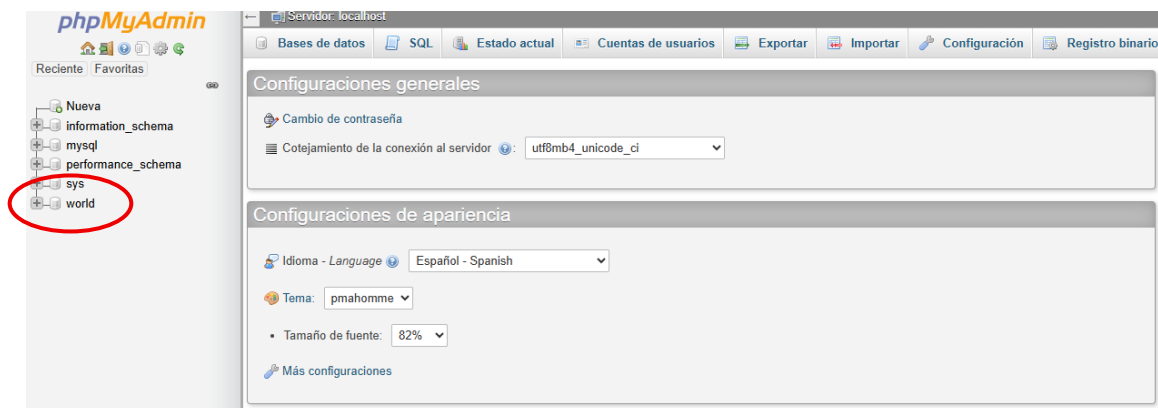
Y con esto crearíamos un script para que ejecute de forma segura y automática cada mes una copia de seguridad.

2. PhpMyAdmin

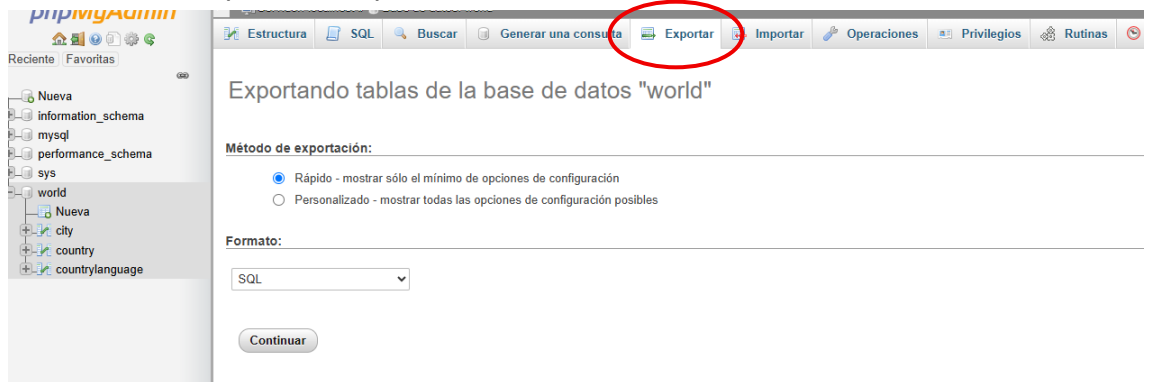
Vamos a realizar esta parte de la práctica en phpMyAdmin en un sistema de Windows 11.

1. **Realiza una copia de una BBDD con mysqldump. Detén el servidor MySQL y elimina el contenido de la carpeta de datos. Restaura la copia.**

Al entrar en <http://localhost/phpmyadmin> en el apartado de nuestra izquierda vemos las bases de datos que tenemos en MySQL, seleccionamos world.



Nos vamos a la pestaña exportar.



Elegimos el método personalizado y marcamos las casillas **Agregar sentencia DROP TABLE / VIEW / PROCEDURE / FUNCTION / EVENT / TRIGGER** y **Agregar sentencia CREATE DATABASE / USE**.

Opciones de creación de objetos

Agregar sentencias.

- ☒ Agregar sentencia CREATE DATABASE / USE
- ☒ Agregar sentencia DROP TABLE / VIEW / PROCEDURE / FUNCTION / EVENT / TRIGGER
- ☐ Agregar sentencia CREATE TABLE

☐ IF NOT EXISTS (menos eficiente porque los índices se generarán durante la creación de la tabla)

☒ Valor AUTO_INCREMENT


☒ Agregar sentencia CREATE VIEW

☒ Agregar sentencia CREATE PROCEDURE / FUNCTION / EVENT

☒ Agregar sentencia CREATE TRIGGER

☒ Encerrar nombres de tablas y campos con comillas invertidas (*protege los nombre de tablas y campos compuestos con caracteres especiales o palabras clave*)

Y pulsamos en continuar, esto nos creara un archivo llamado **world.sql**.

 **world.sql**
724 KB • Hecho

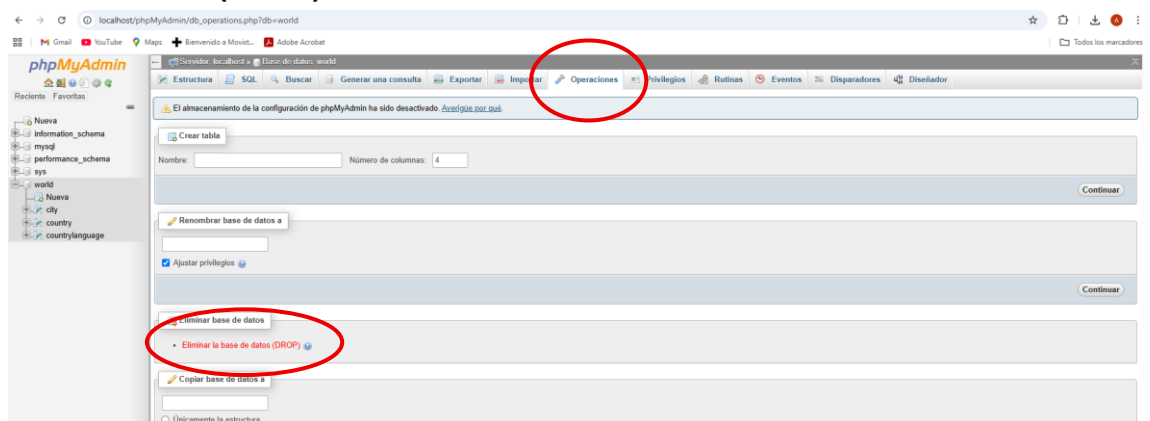
```

C: > Users > Usuario > Downloads > world.sql

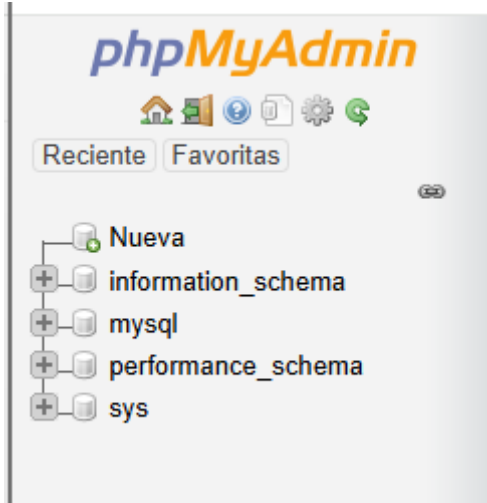
1  -- phpMyAdmin SQL Dump
2  -- version 4.9.1
3  -- https://www.phpmyadmin.net/
4  --
5  -- Servidor: localhost
6  -- Tiempo de generacion: 07-11-2025 a las 10:23:24
7  -- Version del servidor: 8.0.17
8  -- Version de PHP: 7.3.10
9
10 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 SET AUTOCOMMIT = 0;
12 START TRANSACTION;
13 SET time_zone = "+00:00";
14
15
16 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
17 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
18 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
19 /*!40101 SET NAMES utf8mb4 */;
20
21 --
22 -- Base de datos: `world`
23 --
24 CREATE DATABASE IF NOT EXISTS `world` DEFAULT CHARACTER SET utf8mb4 COLLA
25 USE `world`;
26
27 -----
28
29 --
30 -- Estructura de tabla para la tabla `city`
31 --
32
33 DROP TABLE IF EXISTS `city`;
34 CREATE TABLE `city` (
35   `ID` int(11) NOT NULL,

```

Vamos ahora a borrar los datos de world, para ello vamos a nuestra base de datos world en phpMyAdmin, vamos a la pestaña operaciones y **Eliminar la base de datos (DROP)**.



Hemos eliminado nuestra base de datos world y ya no aparece en la lista de base de datos.



Ahora vamos a importar nuestra base de datos **world**, vamos a la pestaña Importar:

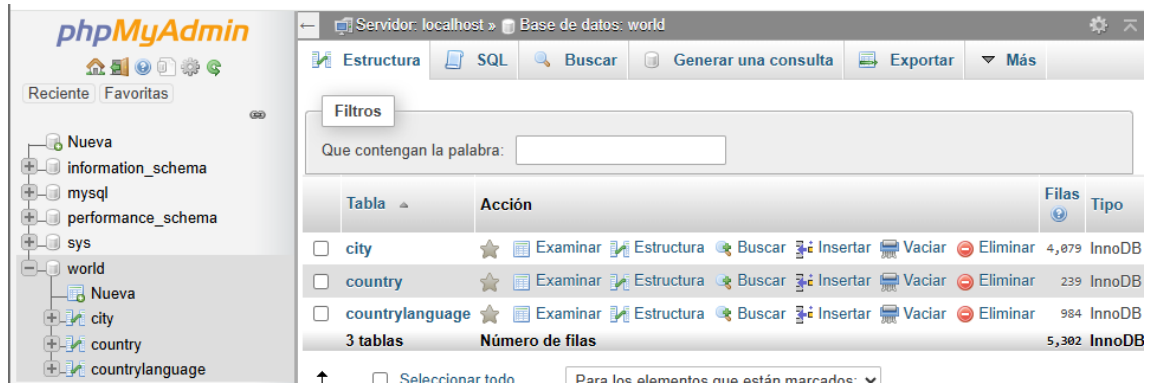


Seleccionamos nuestro archivo creado previamente como **world.sql**:

Archivo a importar:

El archivo puede ser comprimido (gzip, zip) o descomprimido.
Un archivo comprimido tiene que terminar en **[formato].[compresión]**. Por ejemplo: **.sql.zip**
Buscar en su ordenador: **Seleccionar archivo** world.sql (Máximo: 2,048KB)
También puede arrastrar un archivo en cualquier página.
Conjunto de caracteres del archivo: utf-8

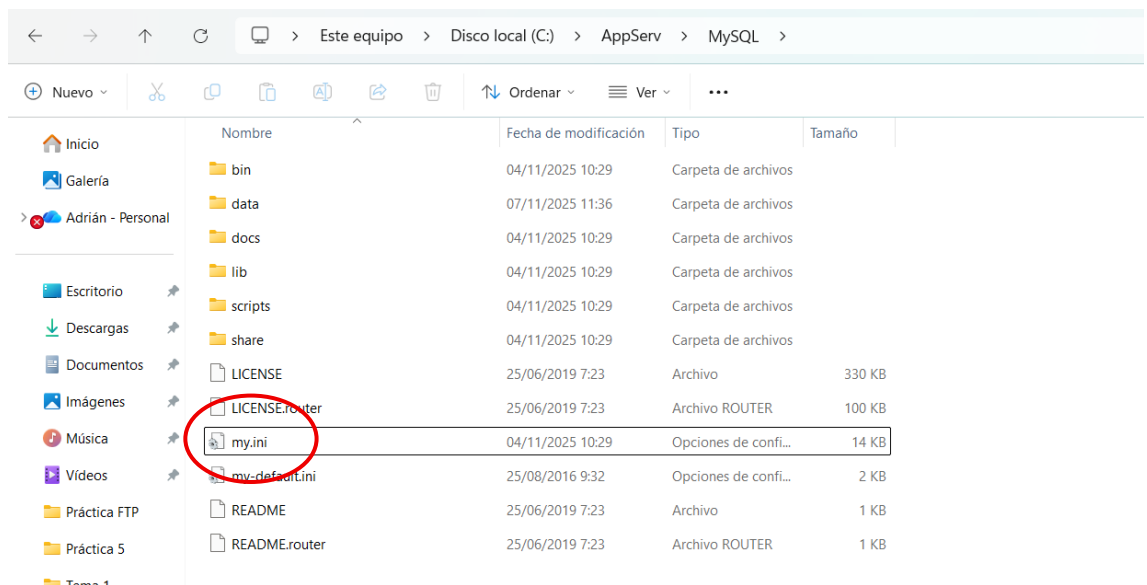
Formato **SQL** y clicamos en Continuar. Entramos dentro de nuestra base de datos recién restaurada y vemos que contiene todas las tablas que tenía:



2. Copia de seguridad Incremental:

a. Activa el registro binario de tu servidor MySQL.

Para activar nuestro registro binario tenemos que irnos al archivo **my.ini** en nuestra carpeta de configuración de AppServ:

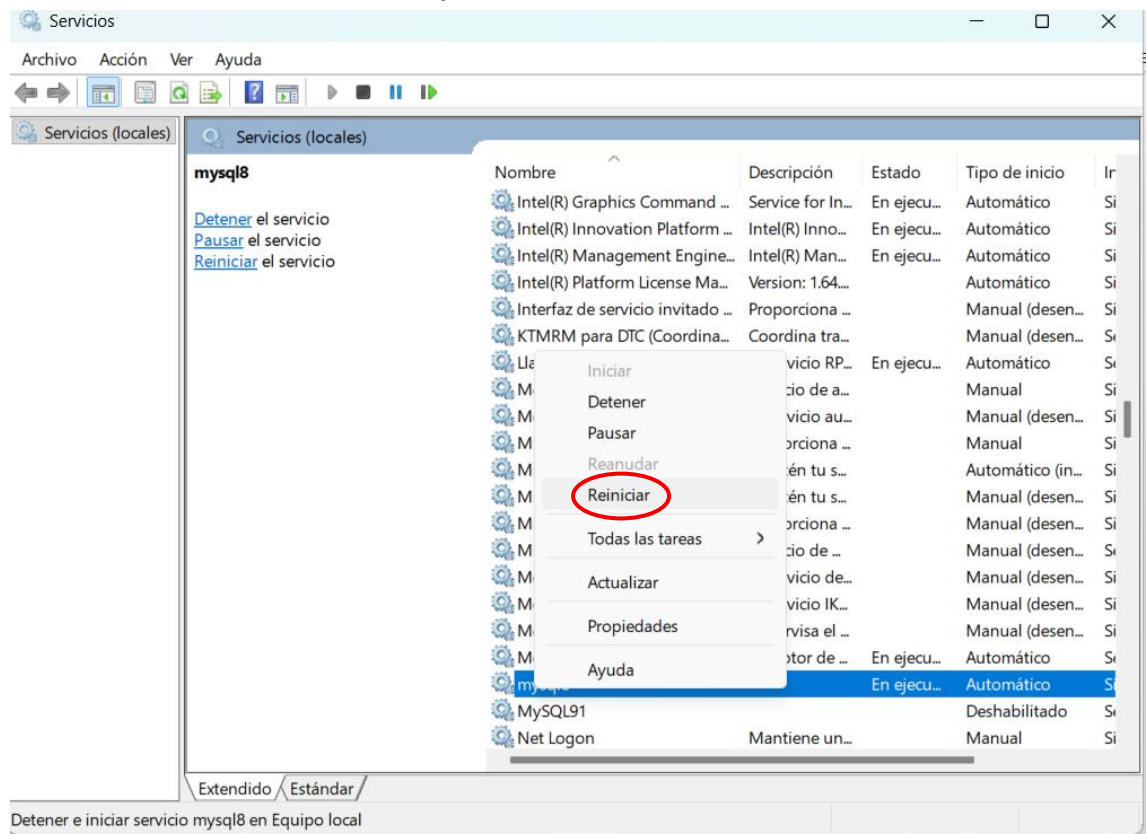


Vamos a modificar este archivo justo debajo de la parte de **[mysql]**:

```
[mysql]
server-id=1
log-bin=mysql-bin
binlog-format=MIXED
expire_logs_days=7
max_binlog_size=100M
```

Así activamos nuestros binlogs y hacemos que borre logs más antiguos de 7 días.

Reiniciamos nuestro servicio MySQL:



Y vamos a comprobar que nuestro binlog este activo ejecutando en nuestro CMD **SHOW VARIABLES LIKE 'log_bin'; SHOW BINARY LOGS;**

```
Current database: world

+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_bin       | ON    |
+-----+-----+
1 row in set, 1 warning (0.04 sec)

+-----+-----+-----+
| Log_name      | File_size | Encrypted |
+-----+-----+-----+
| binlog.000004 | 178       | No        |
| binlog.000005 | 178       | No        |
| binlog.000006 | 723171    | No        |
| binlog.000007 | 724039    | No        |
| binlog.000008 | 155       | No        |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Vemos que está activo y una lista de archivos binlogs.

b. Realiza una copia de seguridad completa.

Antes de continuar, los archivos **mysql-bin.00000X** no son archivos SQL sino archivos binarios, por lo tanto phpMyAdmin no puede restaurarlos, dado que solo puede restaurar archivos .sql.

Para continuar con la práctica vamos a realizar por consola de Windows todos los comandos que phpMyAdmin no nos permite realizar.

Primero en nuestro archivo **my.ini** comentamos la siguiente línea debajo de **[client]**:

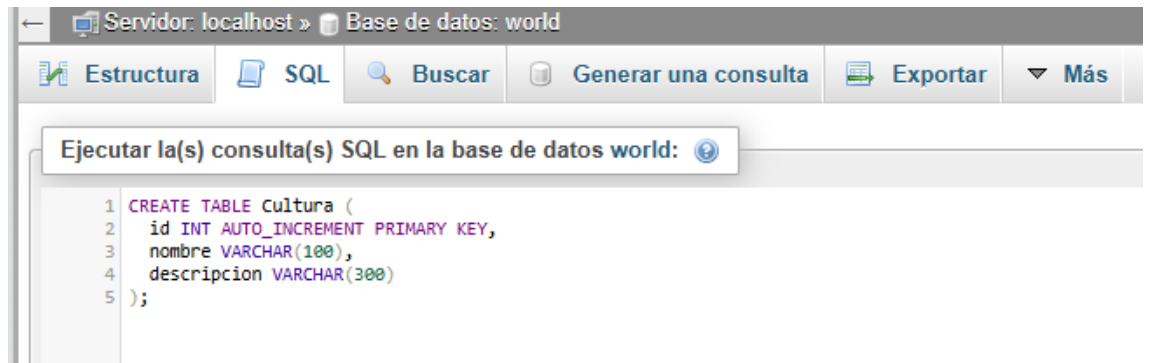
```
"
[client]
#no-beep
```

Después, creamos una carpeta llamada **backups** en nuestro disco local C: y ejecutamos un **mysqldump** con la base de datos **world**:

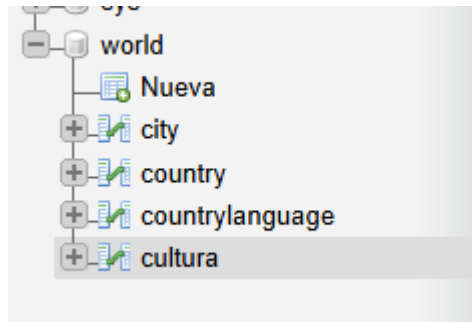
```
C:\Users\Usuario>"C:\AppServ\MySQL\bin\mysqldump.exe" -u root -p --databases world > C:\backups\world_full.sql
Enter password: *****
```

c. Haz varias operaciones de modificación de datos en la Base de datos.

Vamos a crear nuestra tabla **cultura** como hemos hecho en Ubuntu pero esta vez en phpMyAdmin, para ello vamos a la pantalla **SQL** y ejecutamos:



Clicamos en Continuar y crearemos nuestra tabla cultura:



d. Eliminamos todos los datos.

Vamos a ir tabla por tabla en nuestra base de datos **world**, nos iremos a la pestaña **Operaciones** y seleccionaremos la opción **Borrar la tabla (DROP)**:



e. Restaurar todos los datos de una manera adecuada.

Vamos a ver la listade binlogs creados:

```
mysql> SHOW BINARY LOGS;
```

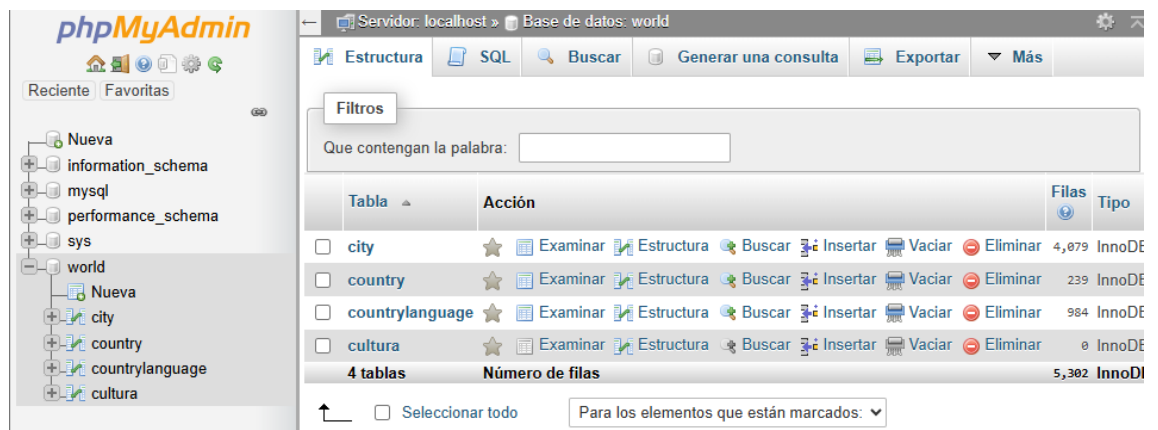
| Log_name | File_size | Encrypted |
|---------------|-----------|-----------|
| binlog.000004 | 178 | No |
| binlog.000005 | 178 | No |
| binlog.000006 | 723171 | No |
| binlog.000007 | 724039 | No |
| binlog.000008 | 1288 | No |

5 rows in set (0.00 sec)

Vamos a ejecutar nuestro **binlog.000007**, que probando los binlogs hemos llegado a la conclusión de que esa tiene nuestra base de datos enteras incluida la tabla cultura:

```
C:\Users\Usuario>"C:\AppServ\MySQL\bin\mysqlbinlog.exe" "C:\AppServ\MySQL\data\binlog.000007" | "C:\AppServ\MySQL\bin\mysql.exe" -u root -p
Enter password: *****
```

Si refrescamos phpMyAdmin podemos ver que nuestra base de datos world ha vuelto a tener todas sus tablas incluido Cultura:



3. Copia de Seguridad periódica. Configura en Ubuntu (con Cron) o Windows (tarea programada o directiva de tarea) para realizar copias de seguridad periódicas cada mes.

Puedes hacer un script para guardar de alguna manera un backup lo puedes realizar en Ubuntu o Windows.

Vamos a crear previamente una carpeta llamada **mysql** dentro de nuestra carpeta creada anteriormente llamada backups en nuestro disco C:.

Creemos nuestro script en un bloc de notas:

```
@echo off
:: =====
:: Script de copia de seguridad MySQL (AppServ)
:: =====

:: Fecha en formato YYYY-MM-DD
for /f "tokens=1-3 delims=/" %%a in ("%date%") do (
    set d=%%c-%%b-%%a
)

:: Carpeta destino
set DEST=C:\backups\mysql

:: Nombre del archivo
set FILE=%DEST%\world_%d%.sql

:: Crear carpeta si no existe
if not exist "%DEST%" mkdir "%DEST%"

echo Realizando copia de seguridad de la base de datos WORLD...
"C:\AppServ\MySQL\bin\mysqldump.exe" --no-defaults -u root -pTU_CONTRASEÑA --databases
world > "%FILE%"

echo Copia completada: %FILE%
```

En la parte que pone **TU_CONTRASEÑA** ponemos la contraseña de root.

Guardamos el archivo como **backup_world.bat**.

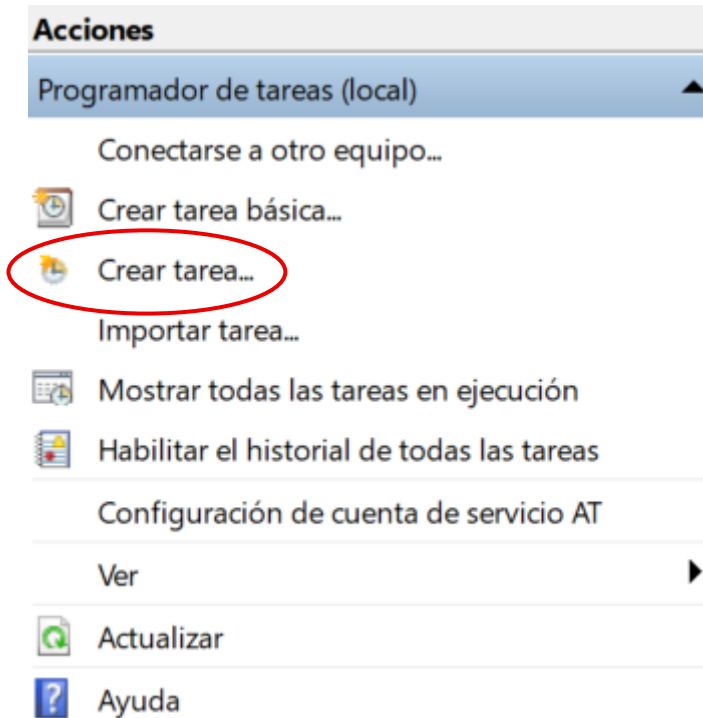
| | | | |
|--|------------------|-----------------------|------|
|  backup_world.bat | 07/11/2025 12:45 | Archivo por lotes ... | 1 KB |
|--|------------------|-----------------------|------|

Para probar si nuestro script funciona lo abrimos como administración y podemos ver que dentro de nuestra carpeta mysql aparece un archivo nuevo parecido a este:

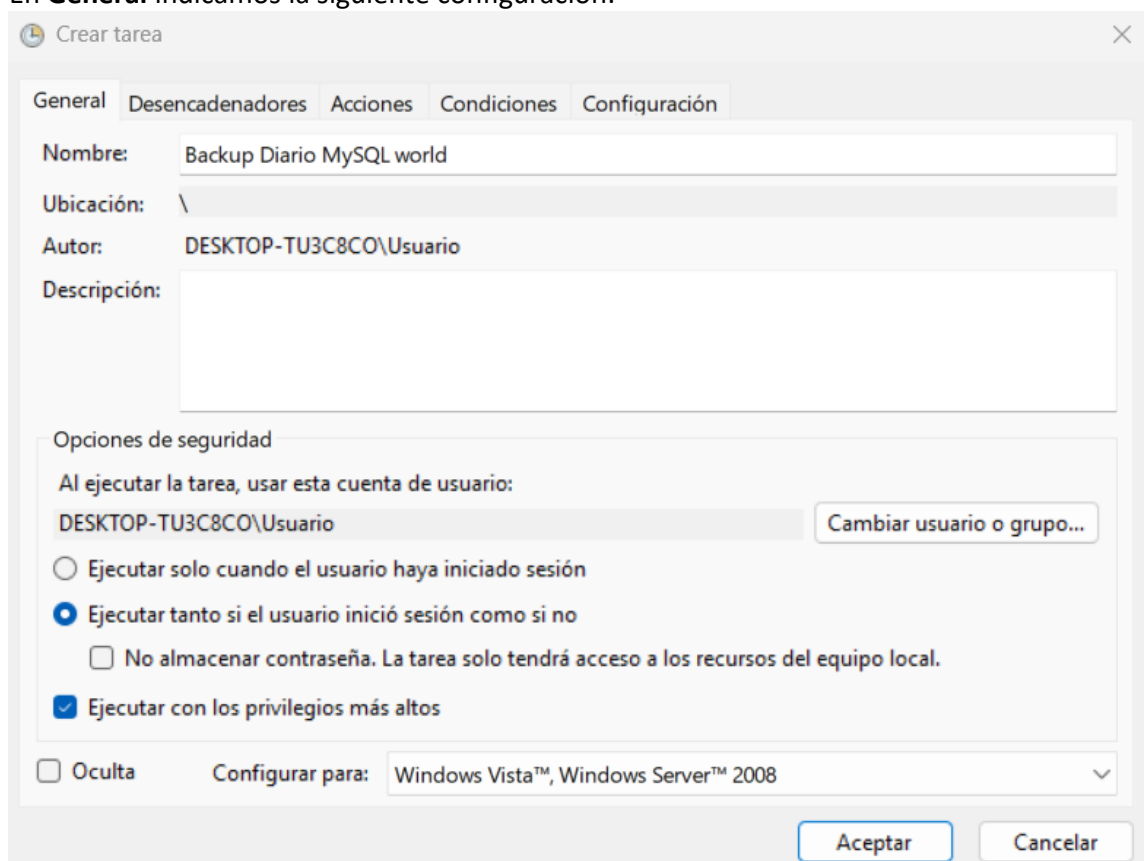
| | | | |
|--|------------------|-----------------------|--------|
|  world_2025-11-07.sql | 07/11/2025 12:46 | Archivo de origen ... | 239 KB |
|--|------------------|-----------------------|--------|

Ahora vamos a programar la tarea automática con el **Programador de tareas**.

Una vez estemos en el hacemos clic en **Crear tarea**:



En **General** indicamos la siguiente configuración:



En **Desencadenadores** clicamos en **Nuevo** y ponemos la siguiente configuración:

Nuevo desencadenador ✕

Iniciar la tarea: Según una programación ▼



Configuración

☐ Una vez

☒ Diariamente

☐ Semanalmente

☐ Mensualmente

Inicio: 07/11/2025  3:00:00  ☐ Sincronizar zonas

Repetir cada: 1 días



Configuración avanzada

☐ Retraso máx. (retraso aleatorio): 1 hora ▼

☐ Repetir cada: 1 hora ▼ durante: 1 día ▼

☐ Detener todas las tareas en ejecución al final de la duración de repetición

☐ Detener la tarea si se ejecuta durante más de: 3 días ▼

☐ Expiración: 07/11/2026  12:51:56  ☐ Sincronizar zonas horaria

☒ Habilitado

Aceptar Cancelar

En la pestaña **Acciones** clicamos en **Nueva** y agregamos nuestro script:

Nueva acción ✕

Especifique la acción que debe realizar esta tarea.

Acción: Iniciar un programa ▼

Configuración

Programa o script:
C:\backups\backup_world.bat Examinar...

Agregar argumentos (opcional):

Iniciar en (opcional):

Aceptar Cancelar

En la pestaña **Condiciones** dejamos la siguiente configuración:

The screenshot shows the 'Crear tarea' (Create task) dialog box with the 'Condiciones' (Conditions) tab selected. The dialog has a title bar with a close button and a subtitle 'Crear tarea'. Below the title bar are five tabs: 'General', 'Desencadenadores', 'Acciones', 'Condiciones', and 'Configuración'. The 'Condiciones' tab is active, displaying the following settings:

- General instruction:** Especifique las condiciones que, junto a los desencadenadores, determinarán si se debe ejecutar la tarea. Si alguna de las condiciones especificadas no se cumple, no se ejecutará la tarea.
- Inactivo:**
 - ☐ Iniciar la tarea solo si el equipo está inactivo durante: 10 minutos (dropdown)
 - Esperar a que esté inactivo durante: 1 hora (dropdown)
 - ☒ Detener si el equipo deja de estar inactivo
 - ☐ Reiniciar si el estado de inactividad se reanuda
- Energía:**
 - ☐ Iniciar la tarea solo si el equipo está conectado a la corriente alterna
 - ☒ Detener si el equipo empieza a usar la batería
 - ☐ Activar el equipo para ejecutar esta tarea
- Red:**
 - ☐ Iniciar solo si la siguiente conexión de red está disponible: Cualquier conexión (dropdown)

At the bottom right, there are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel).

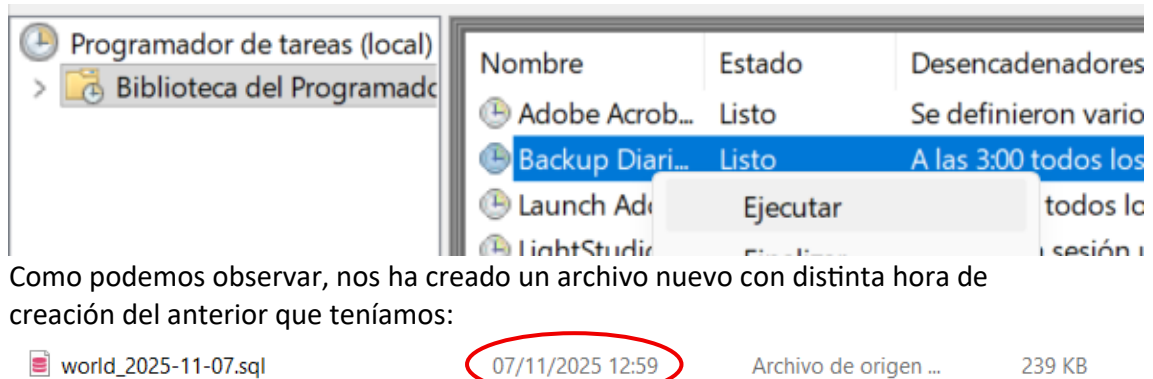
Y en la pestaña **Configuración**:

The screenshot shows the 'Crear tarea' (Create task) dialog box with the 'Configuración' (Configuration) tab selected. The dialog has the same title bar and tabs as the previous screenshot. The 'Configuración' tab is active, displaying the following settings:

- General instruction:** Especifique la configuración adicional que afecta al comportamiento de la tarea.
- ☒ Permitir que la tarea se ejecute a petición
- ☐ Ejecutar tarea lo antes posible si no hubo inicio programado
- ☐ Si la tarea no se ejecuta, reiniciarla cada: 1 minuto (dropdown)
- Intentar el reinicio un máximo de: 3 veces
- ☒ Detener la tarea si se ejecuta durante más de: 3 días (dropdown)
- ☒ Detener tarea en ejecución si no finaliza cuando se solicite
- ☐ Eliminar tareas no reprogramadas después de: 30 días (dropdown)
- Aplicar la siguiente regla si la tarea ya está en ejecución:**
 - No iniciar una instancia nueva (dropdown)

At the bottom right, there are two buttons: 'Aceptar' (Accept) and 'Cancelar' (Cancel).

Guardamos nuestra tarea y vamos a localizarla y ejecutarla con clic derecho para ver si funciona:



Diferencias Terminal Ubuntu y phpMyAdmin

En los ejercicios de **Ubuntu**, las copias de seguridad de MySQL se realizan desde la **consola** mediante comandos como mysqldump, mysql y mysqlbinlog. Este método ofrece **mayor control y precisión técnica**, aunque requiere conocimientos de línea de comandos.

En cambio, en **phpMyAdmin (Windows con AppServ)**, las mismas operaciones se hacen desde una **interfaz gráfica web**, más cómoda e intuitiva.

En el **Ejercicio 1**, la copia completa se realiza en Ubuntu con mysqldump y en phpMyAdmin con la opción **Exportar**, pudiendo restaurarse desde consola o desde la pestaña **Importar**.

En el **Ejercicio 2**, las copias **incrementales** se gestionan con binlogs (mysql-bin.00000x). En Ubuntu se aplican directamente desde terminal, mientras que en phpMyAdmin se deben manejar mediante **CMD con mysqlbinlog**, ya que phpMyAdmin no soporta archivos binarios.

En el **Ejercicio 3**, la copia **automática** se programa con **cron** en Ubuntu y con el **Programador de tareas de Windows** mediante un script .bat con mysqldump.

En resumen, Ubuntu ofrece un enfoque más profesional y técnico, mientras que phpMyAdmin simplifica el proceso para usuarios que prefieren trabajar en entorno gráfico, logrando en ambos casos el mismo objetivo: **proteger y restaurar la base de datos MySQL** de forma fiable.

Problemas resueltos

Problema detectado

Al intentar acceder a **phpMyAdmin** tras instalar **AppServ**, no se podía iniciar sesión con el usuario root.

El error se debía a una **incompatibilidad entre dos versiones de MySQL** instaladas en el sistema:

- una versión anterior (**MySQL 9.1**) ya instalada en Windows,
- y la versión incluida en AppServ (**MySQL 8.0**).

phpMyAdmin se conectaba por defecto al servicio MySQL 9.1, que usaba un **método de autenticación más moderno (caching_sha2_password)** no reconocido por la versión de PHP incluida en AppServ.

Solución aplicada

1. **Identificamos el conflicto** revisando los puertos y rutas de cada servidor MySQL.
2. **Desactivamos la versión 9.1**, dejando únicamente en ejecución la de AppServ.
3. **Reconfiguramos el usuario root** en el MySQL de AppServ para que usara un método de autenticación compatible:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password  
BY 'TuContraseña';
```

```
FLUSH PRIVILEGES;
```

4. Reiniciamos el servicio MySQL de AppServ y probamos el acceso desde phpMyAdmin.
5. Finalmente, se pudo acceder correctamente con root y la nueva contraseña.

*phpMyAdmin quedó conectado al **servidor MySQL 8.0 de AppServ**, funcionando correctamente.*

El conflicto de autenticación y de versiones se resolvió garantizando compatibilidad total entre MySQL y PHP.