



PRACTICA 4 BACKUP

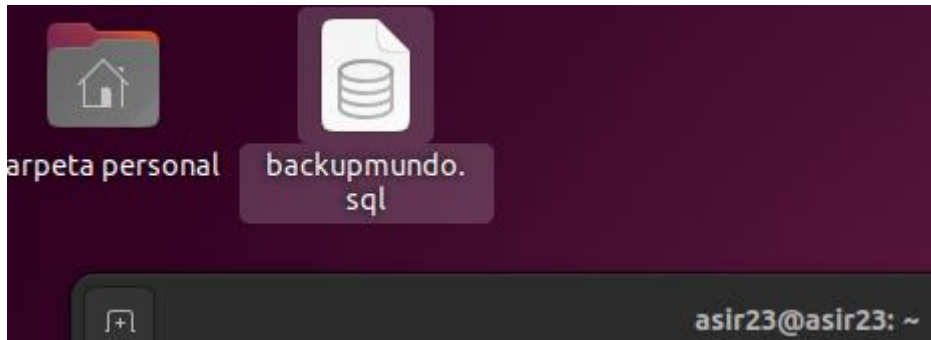
Realizado por: Jesús Padilla Crespo

Índice

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.	2
2. Copia de seguridad Incremental:	4
a. Activa el registro binario de tu servidor MySQL.....	4
b. Realiza una copia de seguridad completa.....	5
c. Haz varias operaciones de modificación de datos en la Base de datos.	5
d. Elimina ahora todos los datos	5
e. Restaura todos los datos de una manera adecuada.	5
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.....	7
4. Repite los 3 ejercicios usando phpmyadmin. Explica las diferencias.....	10
Exportar.....	10
Exportar : Rápido.....	10
Exportar : Personalizado	12
Importar	14

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.

```
asir23@asir23:~$ mysqldump -u root -p world > ./Escritorio/backupmundo.sql
Enter password:
```



Ahora debemos detener el servicio para poder borrar los datos.

Para detener el servicio usaremos el comando `sudo service mysql stop`

```
asir23@asir23:~$ sudo service mysql stop
[sudo] contraseña para asir23:
asir23@asir23:~$ sudo service mysql status
○mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:
   Active: inactive (dead) since Mon 2022-11-07 18:28:19 CET; 18s ago
   Process: 802 ExecStart=/usr/sbin/mysqld (code=exited, status=0/SUCCESS)
   Main PID: 802 (code=exited, status=0/SUCCESS)
   Status: "Server shutdown complete"
   CPU: 7.373s
```

Ahora debemos localizar la ruta donde se encuentren los datos. En mi caso la ruta es `/var/lib/mysql/world` (world es la bbdd que usaremos para ésta práctica).

```
asir23@asir23:/etc/mysql$ sudo ls /var/lib/mysql
asir23.pid      binlog.index    ib_buffer_pool  private_key.pem
auto.cnf        ca-key.pem      ibdata1         public_key.pem
binlog.000001   ca.pem          ibtmp1          server-cert.pem
binlog.000002   client-cert.pem '#innodb_redo'  server-key.pem
binlog.000003   client-key.pem  '#innodb_temp'  sys
binlog.000004   debian-5.7.flag mysql            undo_001
binlog.000005   '#ib_16384_0.dblwr' mysql.ibd        undo_002
binlog.000006   '#ib_16384_1.dblwr' performance_schema world
```

Usaremos el comando `sudo rm -r /var/lib/mysql/world`

```
asir23@asir23:/etc/mysql$ sudo rm -r /var/lib/mysql/world
asir23@asir23:/etc/mysql$ sudo ls /var/lib/mysql
asir23.pid      binlog.index    ib_buffer_pool  private_key.pem
auto.cnf        ca-key.pem      ibdata1         public_key.pem
binlog.000001   ca.pem          ibtmp1          server-cert.pem
binlog.000002   client-cert.pem '#innodb_redo'  server-key.pem
binlog.000003   client-key.pem  '#innodb_temp'  sys
binlog.000004   debian-5.7.flag mysql            undo_001
binlog.000005   '#ib_16384_0.dblwr' mysql.ibd        undo_002
binlog.000006   '#ib 16384 1.dblwr' performance schema
```

Una vez comprobado que hemos borrado los datos deberemos reiniciar el servicio, para ello ejecutaremos la orden `sudo service mysql restart`. Ahora

```
mysql> use world;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from country;
ERROR 1812 (HY000): Tablespace is missing for table `world`.`country`.

asir23@asir23:~$ mysql -u root -p world < ./Escritorio/backupmundo.sql
Enter password:
```

Ya podemos volver a hacer selects ya que hay datos.

```
mysql> select name from country limit 5;
+-----+
| name          |
+-----+
| Aruba          |
| Afghanistan   |
| Angola        |
| Anguilla      |
| Albania       |
+-----+
5 rows in set (0,00 sec)
```

2. Copia de seguridad Incremental:

a. Activa el registro binario de tu servidor MySQL.

Para activar las copias de seguridad incrementales debemos detener el servicio de nuevo.

Ahora accederemos al fichero de configuración con la orden `sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf`

```
asir23@asir23:~$ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

GNU nano 6.2 /etc/mysql/mysql.conf.d/mysqld.cnf
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html
#
# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user                = mysql
# pid-file           = /var/run/mysqld/mysqld.pid
# socket             = /var/run/mysqld/mysqld.sock
# port               = 3306
# datadir            = /var/lib/mysql
```

Una vez dentro del archivo debemos des comentar la siguiente línea “#log bin...”

```
# server-id          = 1
# log_bin            = /var/log/mysql/mysql-bin.log
# binlog_expire_logs_seconds = 2592000

# other settings you may need to change.
# server-id          = 1
log_bin              = /var/log/mysql/mysql-bin.log
# binlog_expire_logs_seconds = 2592000
```

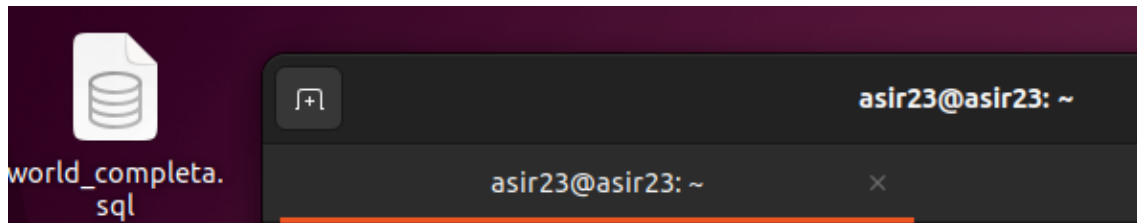
Ahora añadimos la siguiente línea

```
# 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
expire_logs_days     = 10
# binlog_expire_logs_seconds = 2592000
max_binlog_size       = 100M
# binlog_do_db        = include_database_name
```

Una vez acabada la configuración guardamos y reiniciamos el servicio.

b. Realiza una copia de seguridad completa

Para realizar la copia de seguridad completa usaremos el comando `mysqldump -u root -p --flush-logs world > world_completa.sql`



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

Hemos realizado una copia sin cambios(base de datos original), probemos ahora a realizar unos cambios para después guardarlos.

```
mysql> insert into city values(4080,"Albuquerque","USA","Nuevo México",998877);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from city where name like "albuq%";
+-----+-----+-----+-----+-----+
| ID    | Name          | CountryCode | District    | Population |
+-----+-----+-----+-----+-----+
| 3827  | Albuquerque   | USA         | New Mexico  | 448607     |
| 4080  | Albuquerque   | USA         | Nuevo México | 998877     |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

Una vez acabada la edición de la tabla dentro del servidor ejecutamos la orden `flush logs` esto le indicará al sistema un punto de guardado. Posteriormente borraremos la BBDD world y restauraremos la copia de seguridad completa del apartado anterior.

```
mysql> flush logs;
Query OK, 0 rows affected (0.02 sec)
```

d. Elimina ahora todos los datos

```
mysql> drop database world;
Query OK, 3 rows affected (0.08 sec)
```

e. Restaura todos los datos de una manera adecuada.

```
asir23@asir23:~$ mysql -u root -p world < ./Escritorio/world_completa.sql
Enter password:
```

Ahora veremos como haciendo la consulta anterior los cambios no se guardaron en esta copia.

```
mysql> select * from city where name like "albuq%";
+-----+-----+-----+-----+-----+
| ID    | Name          | CountryCode | District    | Population |
+-----+-----+-----+-----+-----+
| 3827  | Albuquerque   | USA         | New Mexico  | 448607     |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Para aplicar los cambios aplicados anteriormente debemos restaurarlos desde los bin de respaldo.

Para ello usaremos el comando `sudo mysqlbinlog /var/log/mysql/mysql-bin.000005 | sudo mysql -u root -p world` para que aplique los cambios en la tabla deseada.

```
asir23@asir23:~$ sudo mysqlbinlog /var/log/mysql/mysql-bin.000005 | sudo mysql -u root -p world
[sudo] contraseña para asir23:
Enter password:
```

Una vez restaurado los cambios comprobemos que todo se ha restaurado correctamente.

```
mysql> select * from city where name like "albuq%";
+-----+-----+-----+-----+-----+
| ID    | Name          | CountryCode | District    | Population |
+-----+-----+-----+-----+-----+
| 3827  | Albuquerque   | USA         | New Mexico  | 448607     |
+-----+-----+-----+-----+-----+
1 row in set (0,01 sec)

mysql> select * from city where name like "albuq%";
+-----+-----+-----+-----+-----+
| ID    | Name          | CountryCode | District    | Population |
+-----+-----+-----+-----+-----+
| 3827  | Albuquerque   | USA         | New Mexico  | 448607     |
| 4080  | Albuquerque   | USA         | Nuevo México | 998877     |
+-----+-----+-----+-----+-----+
2 rows in set (0,00 sec)
```

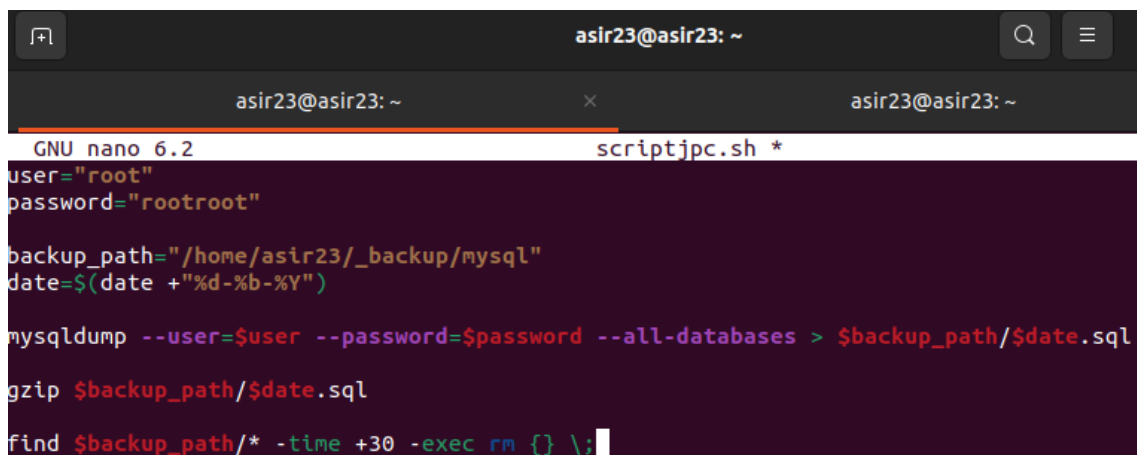
Se han restaurado los cambios correctamente.

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. Repite los 3 ejercicios usando phpmyadmin. Explica las diferencias.

El primer paso será crear nuestro script indicando parámetros como la ruta de guardado, días , hora a la que se ejecute el script etc.

Sudo nano scriptjpc.sh



```
GNU nano 6.2 scriptjpc.sh *
user="root"
password="rootroot"

backup_path="/home/asir23/_backup/mysql"
date=$(date +"%d-%b-%Y")

mysqldump --user=$user --password=$password --all-databases > $backup_path/$date.sql
gzip $backup_path/$date.sql
find $backup_path/* -time +30 -exec rm {} \;
```

user="root"

password="rootroot"

backup_path="/home/asir23/_backup/mysql"

date=\$(date +"%d-%b-%Y")

mysqldump --user=\$user --password=\$password --all-databases > \$backup_path/\$date.sql

gzip \$backup_path/\$date.sql

find \$backup_path/* -time +30 -exec rm {} \;

Ahora debemos darles permiso de ejecución al archivo para ello aplicamos el comando **sudo chmod 777 scriptjpc.sh**

asir23@asir23:~\$ sudo chmod 777 scriptjpc.sh

-rwxrwxrwx 1 asir23 asir23 260 nov 8 09:03 scriptjpc.sh

Debemos instalar Crontab , ejecutaremos la orden `sudo crontab -e`, posteriormente nos pedirá que seleccionemos un editor, en mi caso elegiré la opción 1 que es nano.

```
asir23@asir23:~$ crontab -e
no crontab for asir23 - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed

Choose 1-3 [1]: 1
```

Al pulsar Enter se nos abrirá una ventana para editar la tarea que deseamos programar.

```
GNU nano 6.2 /tmp/crontab.061es7/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 mon dow  command
```

Añadiremos al final del documento(ya que nos da pistas de como escribir la orden correctamente) `0 0 5 * * sh /home/asir23/scriptjpc.sh` →Estoy indicando que a las 5 a.m quiero que ejecute el scriptjpc.sh

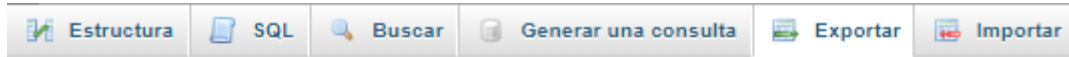
```
GNU nano 6.2 /tmp/crontab.061es7/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide 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 mon dow  command
0 0 5 * * sh /home/asir23/scriptjpc.sh
```

Sólo debemos guardar y salir del editor, ya tendremos una tarea programada.

4. Repite los 3 ejercicios usando phpmyadmin. Explica las diferencias

En PHPMyAdmin elegiremos la base de datos y una vez dentro seleccionar en la barra superior las opciones de importar y exportar.

Exportar



Exportando tablas de la base de datos "world"

Para exportar tenemos 2 opciones: Rápido y Personalizado

Exportar : Rápido

Exportando tablas de la base de datos "world"

Método de exportación:

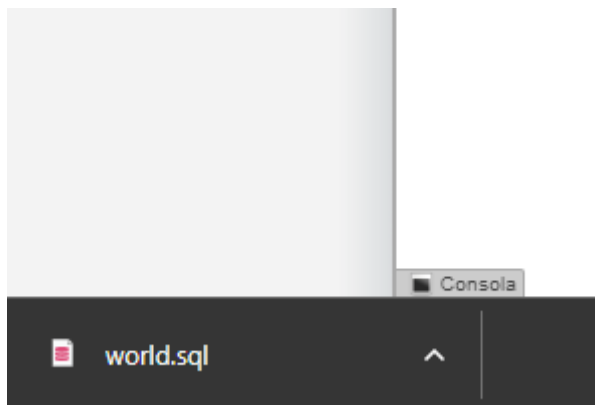
- ☒ Rápido - mostrar sólo el mínimo de opciones de configuración
- ☐ Personalizado - mostrar todas las opciones de configuración posibles

Formato:

SQL

Continuar

La opción Rápido nos sacará un backup de toda la base de datos con instrucciones básicas, create inserts etc...



world.sql X

C: > Users > 2asir-23 > Downloads > world.sql

▷ Run on active connection | ≡ Select block

```
1  -- phpMyAdmin SQL Dump
2  -- version 4.9.1
3  -- https://www.phpmyadmin.net/
4  --
5  -- Servidor: localhost
6  -- Tiempo de generación: 08-11-2022 a las 08:23:52
7  -- Versión del servidor: 8.0.17
8  -- Versión 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
```

Exportar : Personalizado

Al seleccionar la opción de personalizado se nos deja elegir que tablas deseamos exportar con diferentes opciones como puede ser renombrar tablas exportadas, por ejemplo. Algo que me ha llamado la atención es que podemos elegir la opción de creación de datos y objetos ya que nos permite elegir de que manera o estilo de comando deseamos que se guarde en el Backup para restaurar.

☒ Personalizado - mostrar todas las opciones de configuración posibles

Formato:

SQL ▼

Tablas:

	Tablas	Estructura	Datos
	Seleccionar todo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	city	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	country	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	country/language	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Salida:

☐ Renombrar las bases de datos/tablas/columnas exportadas

☐ Usar sentencia LOCK TABLES

☒ Guardar salida a un archivo

Plantilla del nombre del archivo: @DATABASE@ ☒ usar esto para exportaciones futuras

Conjunto de caracteres del archivo: utf-8 ▼

Compresión: Ninguna ▼

☐ Exportar tablas como archivos separados

☐ Ver salida como texto

Omitir tablas mayores a MiB

Opciones específicas al formato:

☒ Mostrar comentarios (incluye información como marca temporal de exportación, versión PHP y versión de servidor)

Comentario de cabecera personalizado adicional (¡n divide líneas):

☐ Incluye una marca temporal de cuando la base de datos fue creada, modificada por última vez y revisada por última vez

☒ Incluir lo exportado en una transacción

☐ Deshabilitar la revisión de las claves foráneas

☐ Exportar vistas como tablas

☐ Exportar metadatos

Sistema de base de datos o versión antigua de servidor MySQL con la que maximizar la compatibilidad de la salida:

NONE ▼

■ Consola

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*)

Opciones de creación de datos

- ☐ Truncar tablas antes de insertar
 - ☐ Sentencias INSERT DELAYED
 - ☐ Sentencias INSERT IGNORE

Función a utilizar al volcar datos: INSERT ▼

Sintaxis a utilizar al insertar datos:

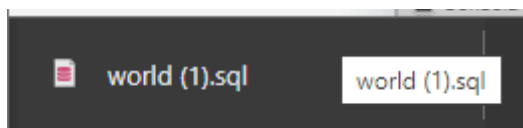
- ☐ incluir nombres de columna en toda sentencia INSERT
Ejemplo: INSERT INTO nombre_tabla (columna_A,columna_B,columna_C) VALUES (1,2,3)
- ☐ incluir múltiples filas en cada sentencia INSERT
Ejemplo: INSERT INTO nombre_tabla VALUES (1,2,3), (4,5,6), (7,8,9)
- ☒ los dos anteriores
Ejemplo: INSERT INTO nombre_tabla (columna_A,columna_B,columna_C) VALUES (1,2,3), (4,5,6), (7,8,9)
- ☐ ninguno de los anteriores
Ejemplo: INSERT INTO nombre_tabla VALUES (1,2,3)

Longitud máxima de la consulta creada

- ☒ Volcar columnas binarias en notación hexadecimal (*por ejemplo "abc" sería 0x616263*)
- ☒ Volcar columnas TIMESTAMP en UTC (*habilita que las columnas TIMESTAMP sean volcadas y cargadas por servidores en diferentes zonas horarias*)

[Continuar](#)

Después de haber configurado los parámetros a nuestro gusto le damos a continuar y nos descargará otro archivo .sql .



Importar



Importando en la base de datos "world"

Haremos una restauración de la base de datos original, para ello hacemos el backup, eliminamos la base de datos y en phpmyadmin nos dirigimos a bases de datos>Importar> seleccionamos el archivo de respaldo y aceptamos.

```
C:\Users\2asir-23>mysqldump -u root -p world > ./Desktop/backupworld.sql
Enter password: *****
```

```
mysql> drop database world;
Query OK, 8 rows affected (0.14 sec)
```

Importando en la base de datos "world"

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: backupworld.sql (Máximo: 2,048KB)

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo:

Importación parcial:

☒ Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite

Omitir esta cantidad de consultas (en SQL) desde la primera:

Otras opciones:

☒ Habilite la revisión de las claves foráneas

Formato:



Había pensado en empaquetar y comprimir los bins que hemos activado en el ejercicio anterior y exportarlos en un .zip ya que PHPMyAdmin aceptaría el formato del paquete pero si la práctica nos pide que sólo usemos PHPMyAdmin esta idea no se puede llevar a la práctica ya que sólo hay una forma de importar y es la que he mostrado más arriba además no he encontrado manera de activar los logs en ésta plataforma.