
TEMA 8(I). COPIAS DE SEGURIDAD.

[Necesidad de realizar copias de seguridad.](#)

[Tipología de las copias de seguridad.](#)

[Planes de copias de seguridad.](#)

[Herramientas para hacer copias de seguridad y restaurar Bases de datos.](#)

[Ejemplos.](#)

[Hacer una copia de seguridad de una BBDD MySQL utilizando el comando Mysqldump desde la consola.](#)

[Copia de seguridad y restauración de una bd con MySQLWorbench.](#)

1. Necesidad de realizar copias de seguridad.

Una copia de seguridad (en general) puede definirse como **una copia de los datos originales que se realiza con el fin de disponer de un medio para recuperarlos en caso de su pérdida.**

Las copias de seguridad son útiles ante distintos eventos y usos: recuperar los sistemas informáticos y los datos de una catástrofe informática, natural o ataque; restaurar una pequeña cantidad de archivos que pueden haberse eliminado accidentalmente, corrompido, infectado por un virus informático u otras causas; guardar información histórica de forma más económica que los discos duros y además permitiendo el traslado a ubicaciones distintas de la de los datos originales; etc.

El proceso de copia de seguridad se complementa con otro conocido como restauración de los datos que es la acción de leer y grabar en la ubicación original u otra alternativa los datos que fueron almacenados en una copia de seguridad.

Algunos factores o sucesos que pueden ocasionar pérdidas en la información son los siguientes:

- **Fallos en el hardware:** averías en discos, que en ocasiones pueden ser recuperados fácilmente o a través de empresas especializadas en estos servicios, pero a un coste que habitualmente es mayor al daño sufrido; en algunas ocasiones puede hacerse imposible recuperar la información.
- **Errores humanos:** se puede llegar a eliminar o modificar información importante, y necesitar su recuperación.
- **Necesidad de recuperar antiguas versiones de datos** que hayan sido modificados y que por el motivo que sea, necesitamos consultar o recuperar.
- **Virus informáticos o cualquier otro tipo de malware** en general.
- **Degradación de un sistema operativo o de la propias base de datos**, simplemente por su uso, o mal uso.
- **Cualquier intrusismo**, por ejemplo de un hacker, sobre cualquier parte del sistema, que pueda provocar la modificación o pérdida de la .
- **Un incendio o un robo.**
- **Catástrofes naturales como inundaciones o terremotos.**

Además de estos factores que pueden producir un borrado o modificación de los datos, una copia de seguridad puede ser necesarias porque **alguna ley lo exija en función del país, la actividad de la empresa u**

otras circunstancias. Por ejemplo, las obligaciones legales, por ejemplo, el Reglamento Europeo de Protección de Datos (RGPD) obliga a cualquier empresa que trate datos de carácter personal, a establecer procedimientos de actuación para la realización de copias de respaldo. A este tipo de copias de seguridad se las suele llamar **COPIAS REGULADAS**.

Otro motivo puede ser la de **archivar datos en un medio de almacenamiento más económico**. Por ejemplo, si se dispone de una cabina de discos redundantes de alto rendimiento que albergue toda la información de la empresa, el precio de este sistema de almacenamiento por Megabyte de información podría resultar demasiado costoso para una eventual ampliación, de forma que podría ser recomendable, bajo un determinado volumen de información, el archivado de proyectos cerrados o de archivos obsoletos a soportes externos, como cintas LTO (Linear Tape Open), (https://es.wikipedia.org/wiki/Linear_Tape_Open) o conjuntos de discos más económicos, cuyo precio por Megabyte sea inferior.

La importancia para una empresa u organización de mantener su información convenientemente respaldada reside en valorar varios aspectos:

- Coste económico para la empresa en caso de que tuviera que reiniciar su actividad interrumpida en caso de pérdida de la información.
- El tiempo invertido, con su coste asociado, en la restauración de los datos y la vuelta a la normalidad.
- El coste derivado de una posible sanción por incumplimiento de alguna obligación relativa a la LOPD.
- Los daños que se pueden ocasionar en la imagen de la empresa, en el caso de que se pierda información de clientes, o que su pérdida incide de alguna forma en sus intereses.

2. Tipología de las copias de seguridad.

Hay varias maneras de clasificar las copias de seguridad, atendiendo a diversos criterios. Vamos a ver algunos de ellos:

1º CRITERIO: SEGÚN NECESIDADES:

- **Copias para recuperación ante desastres:** tienen como objeto disponer de una copia que subsane la pérdida potencial de datos valiosos para el usuario.
- **Copias operacionales:** Se hacen para disponer de una instantánea de los datos del sistema en un momento determinado, con la intención de poder regresar a esa situación (sin necesidad de que haya un desastre).
- **Copias reguladas:** Se realizan para cumplir con normativas legales que exigen el almacenado de datos históricos durante un periodo de tiempo. (LOPD en España).

2º CRITERIO: POR GRANULARIDAD: – “Cuándo y de qué se hace copia”. Se determinan diferentes tipos de backups de acuerdo a cuáles son los ficheros copiados.

- **Una copia de seguridad en espejo**, también conocida como RAID 1, crea una copia exacta de los datos en tiempo real. Es decir, mientras trabajas con la información creas una copia espejo en una ubicación alternativa. Sus principales ventajas son las siguientes:
 - La copia se realiza en tiempo real.
 - La recuperación de los datos es un proceso muy ágil.
 - No se almacenan archivos antiguos o en desuso. Al tratarse de una copia en espejo, si borramos un archivo accidentalmente también lo estaremos borrando de la copia de seguridad, por lo tanto, su mayor desventaja reside en la posible pérdida de archivos.
- **BACKUP COMPLETO:** Se trata del tipo de copia de seguridad más básica y probablemente la más realizada. Consiste, como su propio nombre indica, en hacer una copia de todos los datos de nuestro sistema en otro soporte. La ventaja principal de este tipo de copia es que proporciona una fácil restauración de los datos, ya que todos los datos han sido copiados. Sin embargo, existen varios inconvenientes en este tipo de copias:
 - Tienen una mayor necesidad de espacio de almacenamiento frente a los otros tipos de copias, puesto que se copian todos los ficheros de nuestro sistema cada vez que se realiza la copia de seguridad, lo que implica tener información redundante, ya que almacenaremos múltiples veces todos los ficheros, incluso los que no han sufrido ninguna modificación.
 - No es recomendable realizar una copia completa en horario laboral por la carga que conlleva sobre el servidor y los sistemas. Como consecuencia podría ralentizar los equipos o recursos que están siendo utilizados (ordenadores, acceso al servidor, etc.)
 - Coste elevado, debido a que se necesita mucho espacio de almacenamiento
- **BACKUP DIFERENCIAL:** Una copia de seguridad diferencial es similar a una copia incremental en la primera vez que se lleva a cabo, ya que se copiarán todos los datos que hayan cambiado desde el respaldo anterior. Sin embargo, cada vez que se vuelva a lanzar, no solo se copiarán los datos que se hayan modificado desde la última copia, si no todos los que se hayan modificado desde la última copia completa realizada. Esto significa que, con el tiempo, estos tipos de copia se van haciendo más grandes, hasta que se vuelve a realizar la copia completa. Las principales ventajas son:
 - No requieren tanto espacio de almacenamiento como una copia completa.

- A la hora de recuperar un fichero, solo habrá que comprobar su existencia en dos copias de respaldo: la última copia diferencial realizada y la última copia completa realizada.

Entre sus desventajas podemos destacar:

- No es la solución más optimizada en cuanto a espacio. En este sentido la copia es considerable (sin ser tan alta como la de la copia completa).
- **BACKUP INCREMENTAL: Una copia de seguridad incremental sólo copia los datos que han variado desde la última copia de respaldo realizada** (sea completa o incremental). Los ficheros que se han modificado después del último backup se guardan. Las aplicaciones de backup registran la fecha y hora de una copia de seguridad, de manera que, cuando se realiza una copia incremental, dicha aplicación busca la fecha de la última copia y solo almacena los archivos que han sido modificados en el sistema desde esa fecha registrada hasta el momento actual. Como este tipo de respaldo no almacena todos los ficheros, sino solo los ficheros modificados desde la anterior copia, las ventajas principales son:
 - El espacio necesario es mucho menor que el que requiere una copia completa.
 - El tiempo de realización de la copia de seguridad es mucho más corto.

El inconveniente de este tipo de copias reside en la recuperación de los datos. Si, por ejemplo, queremos recuperar un directorio completo cuyos ficheros se han ido modificando poco a poco debemos recuperar los diferentes ficheros de las distintas copias incrementales, ralentizando así el proceso de recuperación. Para evitar este gran inconveniente existen aplicaciones de copias de datos que permiten realizar copias por versiones de archivos, facilitando así su recuperación.

3º CRITERIO: POR OPERATIVIDAD DEL SISTEMA: “En qué estado está el sistema cuando se realiza la copia”. Se determina si es necesario detener la operativa del sistema (dejar de proporcionar servicio para realizar el backup o continuar con el sistema en activo). Tenemos varios tipos:

- **BACKUP FRÍO (COLD) U OFF-LINE:** La operativa del sistema se detiene. Entre el comienzo de la fase de copia y el final de la misma no se hace ninguna operación sobre los datos. No válido para sistemas 24x7 (por ejemplo, una tienda on-line).
- **BACKUP CALIENTE (HOT) U ON-LINE:** La operativa del sistema no se detiene y la copia se hace con el sistema en producción. Requiere fijar el instante de tiempo de referencia. Gestionar no sólo los datos estables sino las modificaciones (log de operaciones) entre ese instante y el final de la copia).



Diferentes copias de seguridad. Fuente INCIBE, instituto nacional de ciberseguridad.

3. Planes de copias de seguridad.

La política de backups es la definición de los diferentes aspectos de las copias de seguridad: ¿de qué se debe hacer backup? ¿Cada cuánto se realiza la copia de seguridad? ¿Qué retención deben tener? ¿Dónde se guardan las copias? ¿Cuánto tiempo es aceptable que se pueda tardar en recuperar datos?

El plan general de copias de seguridad de cualquier entidad, considerado como un proceso bastante estandarizado, es sin embargo muy dependiente de las peculiaridades de cada empresa o usuario, en función de varios aspectos: presupuestales, de actividad de negocio, volumen de información, ámbito de restauración necesario, tipos de datos, etc.

4. Herramientas para hacer copias de seguridad y restaurar Bases de datos.

Existen muchas herramientas que pueden ser utilizadas para realizar las copias de seguridad, algunas herramientas proporcionadas por el SO, otras proporcionadas por los propios SGBD, y otros paquetes de terceros, tanto comerciales como libres.

Algunas de esas herramientas son:

- **tar, cpio (en entornos UNIX o LINUX)**
- **dump**, es una utilidad del comando “mysqldump” que ya viene instalada en el servidor MySQL.

Ejemplo:

```
mysqldump -u USUARIO -p CONTRASENA -all-databases > respaldo.sql
```

- Desde MySQL Workbench, también podemos hacer copias de seguridad desde la opción “Data Export”.
- Otros programas de administración de servidores de BD:
 - o Para MYSQL: MYSQL Administrator, Navicat SQL, Toad, Heidi SQL
 - o Para ORACLE: Oracle Developer, Navicat for Oracle, Toad for Oracle, Iperius Backup.
 - o Para PostgreSQL: PGAdmin
 - o Para SQL Server: SQL Server Management Studio.

5. Ejemplos.

1. Hacer una copia de seguridad de una BBDD MySQL utilizando el comando Mysqldump.

El comando mysqldump es un programa que incluye por defecto el gestor de base de datos MySQL y que permite la posibilidad de realizar copias de seguridad de los datos almacenados ejecutando la instrucción mediante línea de comando, ya sea en una ventana MS-Dos de Windows o en un terminal Linux.

Este comando permite crear copias de seguridad que pueden ser restaurados en distintos tipos de gestores de bases de datos ya que devuelve un fichero SQL con todas las sentencias necesarias para la restauración de la copia en cualquier otro sitio.

Algunas de las opciones más destacadas que nos ofrece el comando mysqldump son:

-add-drop-table: Añade un “drop table” (borrado de tabla) antes de cada sentencia “create” (creación de tabla).

-all-databases o -A: Realiza un backup de todas las bases de datos que hay en el gestor MySQL.

-databases o -B: Realiza una copia de las bases de datos que se le indican a continuación de la opción. Ejemplo: mysqldump –databases bbdd1 bbdd2 > backup.sql

-force o -f: Continúa realizando el backup aunque se produzca un error durante la creación.

-host o -h: Nos permite indicar el servidor MySQL donde está la base de datos.

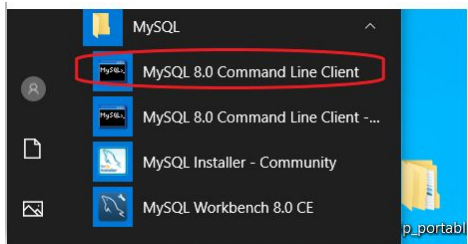
El servidor por defecto es localhost.

-password o -p: Permite indicar la contraseña para conectarse al servidor MySQL.

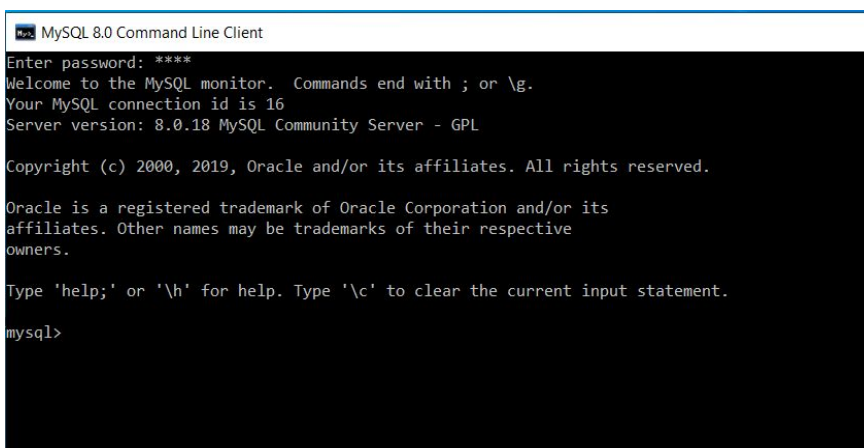
-user o -u: Para indicar el usuario MySQL que se conectará al servidor.

-xml: Nos devuelve un backup en formato XML en vez de SQL.

Vamos a hacer una copia de seguridad utilizando mysqldump. Para ello debemos acceder a la consola de administración de MySQL. Para ello hay que buscar en la carpeta MySQL la opción MySQL Command Line

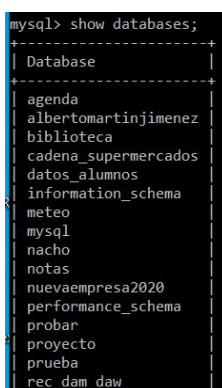


Nos abrirá la consola y lo primero que debemos hacer es conectarnos al servidor, cuando nos pide la password introducir 1234, que es la contraseña del administrador del servidor MySQL.



Ya estamos conectados, ahora veremos las bases de datos que tenemos en nuestro servidor, para ello usamos en comando:

show databases;



Estas son las bases de datos que tengo en mi servidor.

Vamos a crear una base de datos muy pequeña para luego hacer una copia de seguridad:

```
mysql> create database ejemplo_copia;  
Query OK, 1 row affected (0.01 sec)  
  
mysql> use ejemplo_copia;  
Database changed
```

Ahora vamos a crear dos tablas dentro de esa base de datos:

```
mysql> create table t1 (campo1 int primary key, campo2 varchar(24), campo3 varchar(35));  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> create table t2 (campo1 int primary key, campo2 varchar(24), campo3 varchar(35));  
Query OK, 0 rows affected (0.06 sec)
```

Comprobamos que las tablas se han realizado:

```
mysql> show tables;  
+-----+  
| Tables_in_ejemplo_copia |  
+-----+  
| t1                       |  
| t2                       |  
+-----+  
2 rows in set (0.00 sec)
```

Insertamos varios datos en cada tabla:

```
mysql> insert into t1 values (1,'uno','dos'),(2,'dos','tres');  
Query OK, 2 rows affected (0.01 sec)  
Records: 2 Duplicates: 0 Warnings: 0  
  
mysql> insert into t2 values (3,'uno','dos'),(4,'dos','tres');  
Query OK, 2 rows affected (0.01 sec)  
Records: 2 Duplicates: 0 Warnings: 0
```

```
mysql> select * from t1;  
+-----+  
| campo1 | campo2 | campo3 |  
+-----+  
| 1      | uno    | dos    |  
| 2      | dos    | tres   |  
+-----+  
2 rows in set (0.00 sec)
```

```
mysql> select * from t2;  
+-----+  
| campo1 | campo2 | campo3 |  
+-----+  
| 3      | uno    | dos    |  
| 4      | dos    | tres   |  
+-----+  
2 rows in set (0.00 sec)
```

Nos desconectamos de la base de datos y del servidor con el comando quit.

```
mysql> quit
```

 (pulsar ENTER y nos habremos desconectado del servidor).

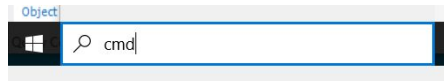
Ahora vamos a hacer la copia de seguridad de la BD ejemplo_copia usando el comando **mysqldump**.

Este comando es uno de los comandos que podemos encontrar dentro de las herramientas de mysql, pero al cual **NO se accede estando conectados al servidor**, sino desde un directorio donde se encuentran todos

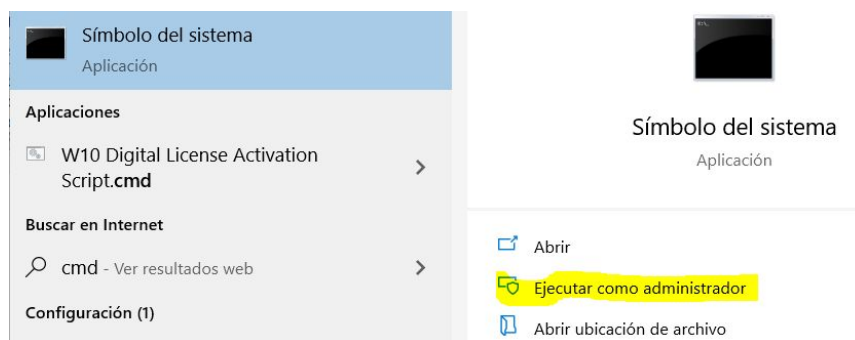
los comandos del servidor. Este directorio se denomina **bin**.

Vamos a situarnos en ese directorio para hacer la copia de seguridad. Estos son los pasos que debes seguir:

1. Abrir un terminal:



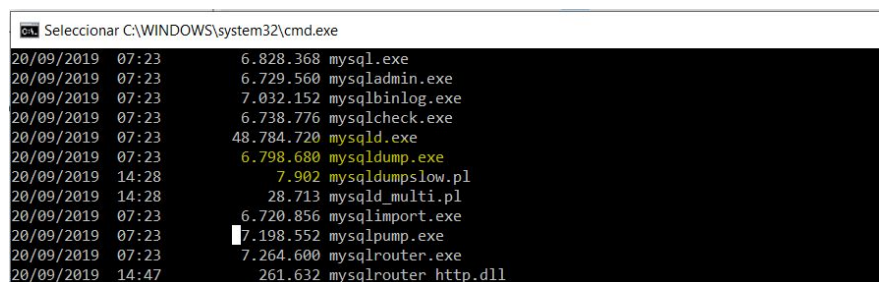
Es conveniente que sea como Administrador, para no tener problemas a la hora de guardar el fichero sql en cualquier directorio.



2. Sitúate en la carpeta correspondiente, la ruta debe ser similar a esta:

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>
```

3. Podemos ver que en ese directorio, está, entre otros muchos, el comando que necesitamos:



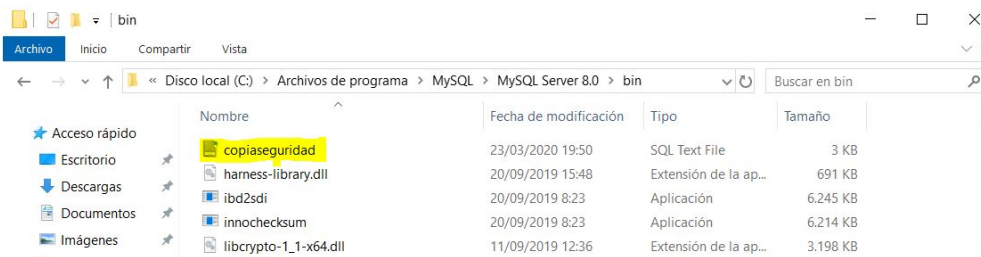
4. escribimos:

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -u root ejemplo_copia > ejemplo_copia.sql
```

(no termina en ; porque no estamos dentro del servidor). Al ejecutarlo, se crea un fichero de texto con extensión .sql que se almacenará en nuestro equipo, y a partir del cual podremos restaurar la base de datos en nuestro servidor o en otro diferente.

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -u root -p ejemplo_copia>copiaseguridad.sql
Enter password: ****
```

El comando se ejecuta pero no devuelve ningún mensaje en caso de que haya funcionado. Ahora vamos a buscar el fichero que se ha creado como copia de seguridad. Si no hemos incluido ninguna ruta específica, se habrá guardado en el directorio donde está el programa.



Este fichero contiene las sentencias sql para crear la bd y las tablas, se puede editar y es apto para poder restaurarlo en mysql o en cualquier otro servidor.

Si editamos el fichero podemos ver las instrucciones de creación y de inserción de datos en las tablas de la BD.

```
copiaseguridad: Bloc de notas
Archivo Edición Formato Ver Ayuda
-- MySQL dump 10.13 Distrib 8.0.18, for Win64 (x86_64)
--
-- Host: localhost    Database: ejemplo_copia
-- Server version    8.0.18
--
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
--
-- Table structure for table `t1`
--
DROP TABLE IF EXISTS `t1`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `t1` (
  `campo1` int(11) NOT NULL,
  `campo2` varchar(24) DEFAULT NULL,
  `campo3` varchar(35) DEFAULT NULL,
  PRIMARY KEY (`campo1`)
```

Ahora vamos a hacer la operación contraria, es decir, vamos a restaurar la BD.

En primer lugar, creo la BD donde voy a restaurar la copia de seguridad. Para ello me conecto al servidor como vimos anteriormente:

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 29
Server version: 8.0.18 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

voy a crear una base de datos vacía, que llamaré restaurar.

```
mysql> create database restaurar;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
```

```

+-----+
| proyecto      |
| prueba        |
| rec_dam_daw   |
| restaurar     |
| sub           |
| sys           |
| teoria        |
+-----+
```

Vuelvo a salir del servidor con quit

```
mysql> quit
Bye
```

Ahora, escribo el siguiente comando: (observa que ahora el comando es mysql ...) Password sigue siendo 1234.

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p restaurar<copiaseguridad.sql
Enter password: ****
```

También puedo utilizar el comando

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysqldump -u root -p restaurar<copiaseguridad.sql
Enter password: ****
```

En ambos casos, la se restaurará la BD restaurar con el contenido del fichero sql copiaseguridad. Ahora voy a comprobar que la Bd se ha restaurado correctamente:

Me conecto como root, y cambio a la BD restaurada:

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: ****
```

```
mysql> use restaurar;
Database changed
mysql> show tables;
+-----+
| Tables_in_restaurar |
+-----+
| t1                   |
| t2                   |
+-----+
2 rows in set (0.00 sec)
```

Contiene las 2 tablas T1 y T2. Comprobamos sus datos:

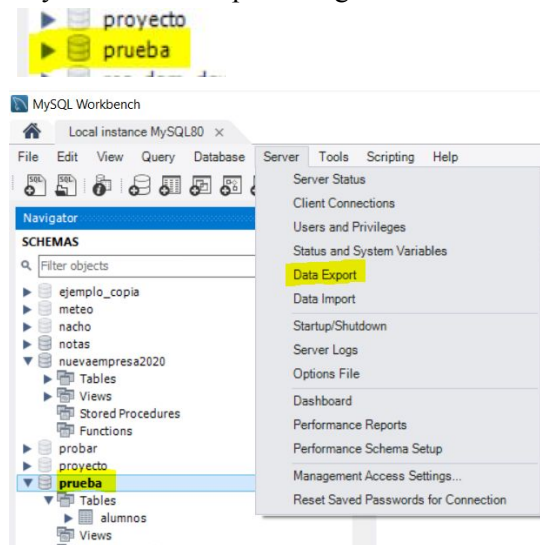
```
mysql> select * from t1;
+-----+-----+-----+
| campo1 | campo2 | campo3 |
+-----+-----+-----+
| 1      | uno    | dos    |
| 2      | dos    | tres   |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

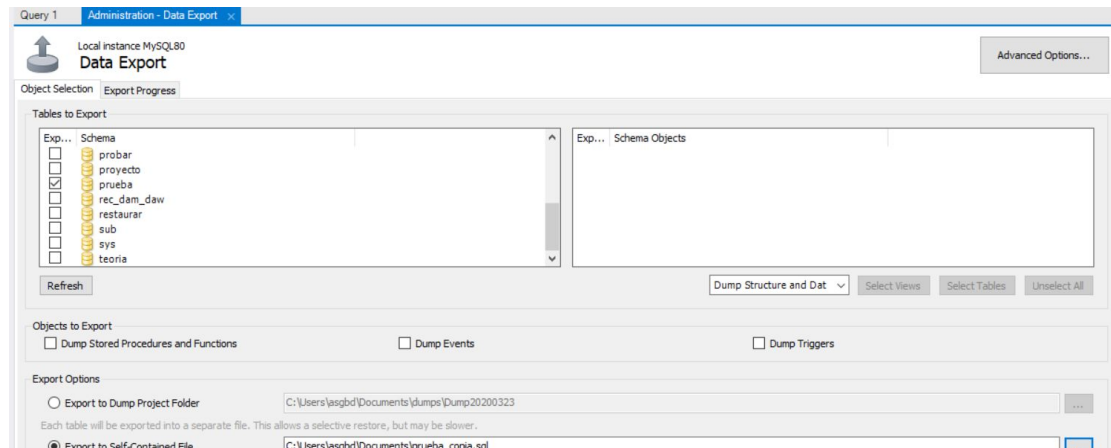
```
mysql> select * from t2;
+-----+-----+-----+
| campo1 | campo2 | campo3 |
+-----+-----+-----+
| 3      | uno    | dos    |
| 4      | dos    | tres   |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

2. Copia de seguridad y restauración de una bd con MySQLWorkbench.

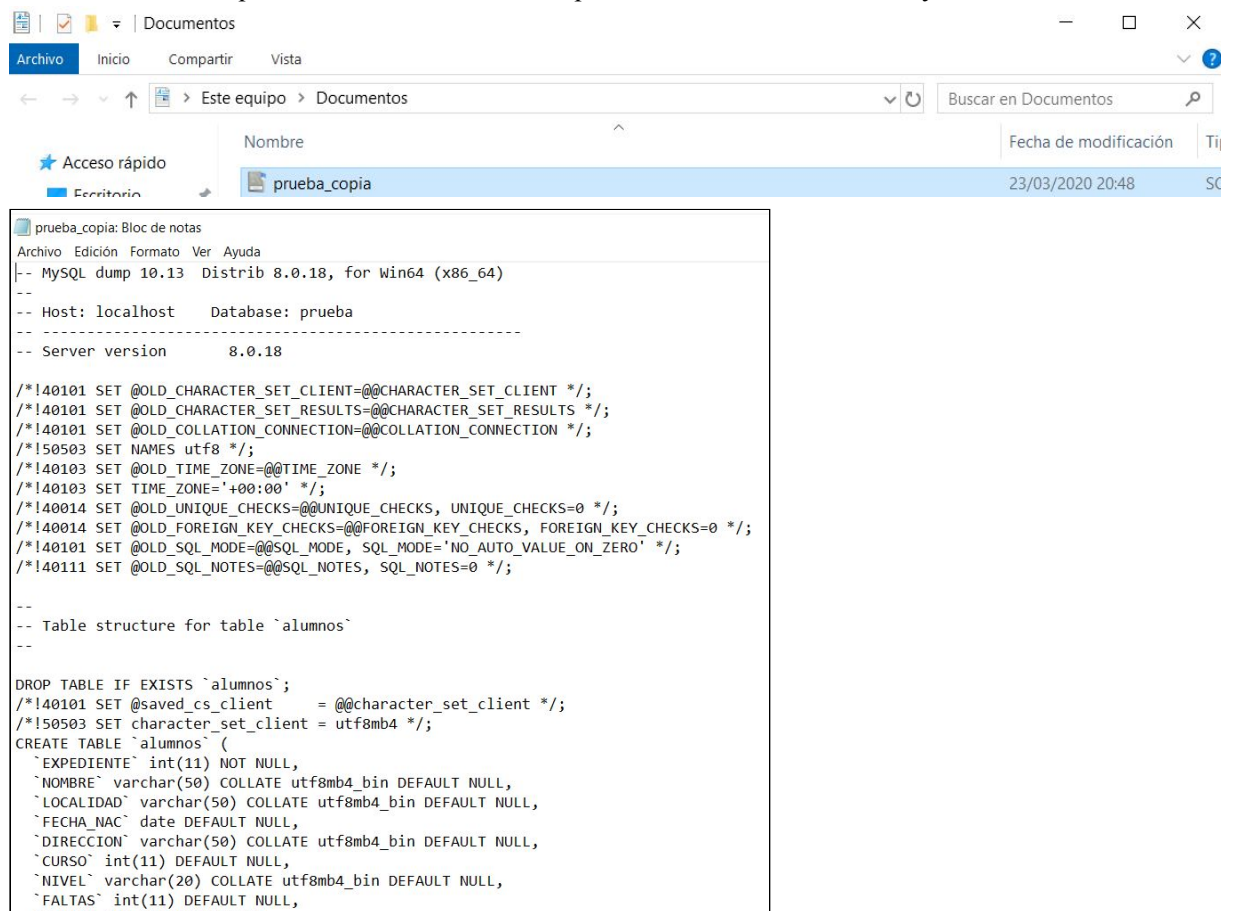
Ahora vamos a hacer una copia de seguridad y restaurar la BD con el MysqlWorkbench y sus herramientas gráficas.

Voy a hacer una copia de seguridad de la BD prueba que tengo en mi servidor.

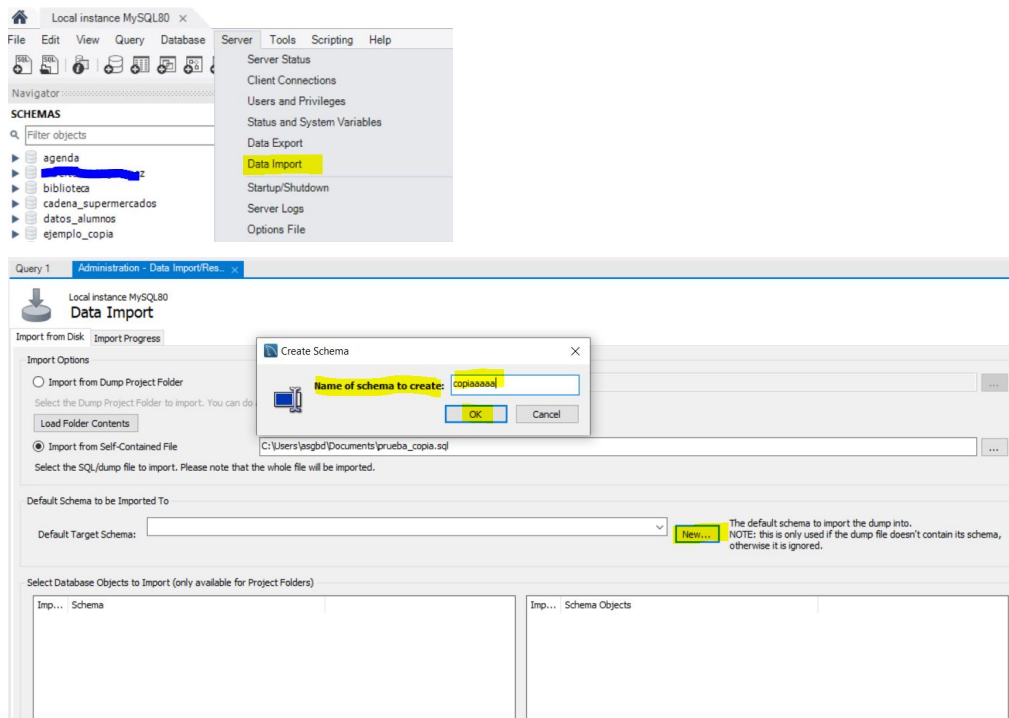




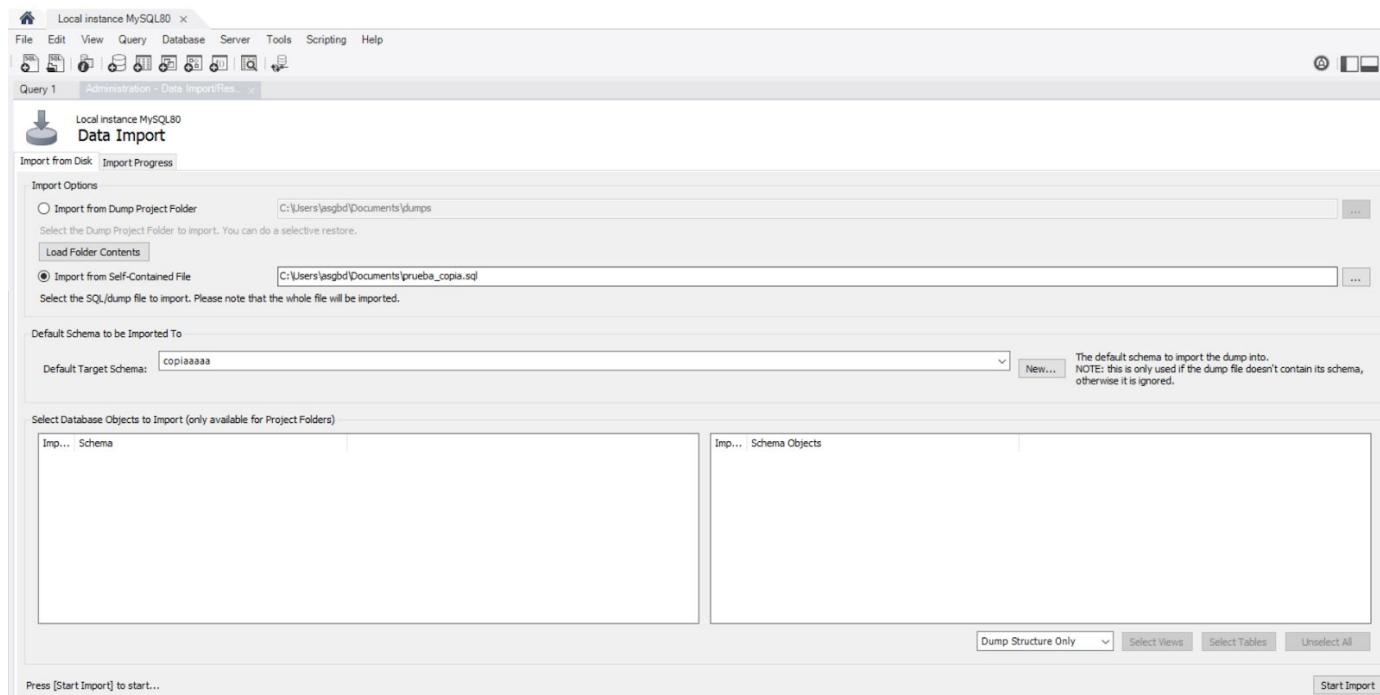
Se crea un archivo que contiene las sentencias sql de creación de la BD, tabla y datos.



La operación contraria es restaurar o importar la base de datos:



El botón **New...** nos permite crear una nueva base de datos para restaurar en ella desde un fichero procedente de una copia de seguridad.



Pulsar en el botón **Start Import** para comenzar la importación de la base de datos.

Import from Disk Import Progress

Import Completed

Status:
1 of 1 imported.

Log:

```

Creating schema copiaaaaa
22:17:07 Restoring C:\Users\asgbd\Documents\prueba_copia.sql
Running: mysql.exe --defaults-file="C:\Users\asgbd\AppData\Local\temp\lmpnl4hl.cnf" --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=copiaaaaa < "C:\Users\asgbd\Documents\prueba_copia.sql"
22:17:07 Import of C:\Users\asgbd\Documents\prueba_copia.sql has finished
  
```

Comprobamos que tanto la BD como las tablas se ha creado correctamente.

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHMAS

- biblioteca
- cadena_supermercados
- copiaaaaa**
 - Tables
 - alumnos**
 - Columns
 - EXPEDIENTE
 - NOMBRE
 - LOCALIDAD
 - FECHA_NAC
 - DIRECCION
 - CURSO
 - NIVEL
 - FALTAS
 - Indexes
 - Foreign Keys
 - Triggers

Administration Schemas

Information

Table: alumnos

Query 1 Administration - Data Import/Res... alumnos x

1 • SELECT * FROM copiaaaaa.alumnos;

Limit to 1000 rows

Result Grid

EXPEDIENTE	NOMBRE	LOCALIDAD	FECHA_NAC	DIRECCION	CURSO	NIVEL	FALTAS
123456	Juan Miquel Soler Bakero	Plasencia	1995-10-10	Gran Vía, 2, 4A	1	CF	15
223322	Raquel Riquelme Rubio	Plasencia	1990-11-23	San Juan, 14, 3B	1	BACHILLERATO	7
334400	Javier Ramanez Rodriguez	Plasencia	1993-05-27	Gran Vía, 4, 3A	3	ESO	0
334455	Pedro Jesus Rodriguez Soler	Plasencia	1994-03-10	Camino de Badel, 4	2	ESO	11
445544	Juan Francisco Cano Riquelme	Caceres	1992-07-01	Plaza de Beluga, 3, 4A	4	ESO	13
554411	Joaquán Hernandez Gonzalez	Plasencia	1991-12-12	Junterones, 4, 5A	2	BACHILLERATO	14
654321	Laura Gomez Fernandez	Caceres	1994-05-10	Junterones, 10, 5B	2	ESO	25
765432	Beatriz Martinez Hernandez	Plasencia	1993-05-05	Plaza Mayor, 6, 3B	3	ESO	5
987654	Diego Martán Llorente	Navalmoral	1990-06-03	Diego de la Cierba, 5, 7A	1	BACHILLERATO	34
993322	Gema Rubio Colero	Caceres	1992-09-09	Plaza Fuensanta, 5, 7A	1	BACHILLERATO	19
998877	Cristina Sanchez Bermejo	Caceres	1995-03-19	Torre de Romo, 7	1	CF	1