



**Argentina
programa
4.0**



Ministerio de Economía
Argentina

Secretaría de
Economía del Conocimiento

***primero
la gente***

Clase 29: Bases de datos Relacionales

Backup y Restauración desde Node.js

Agenda de hoy

- A. Administración de bb.dd. - Backup Y Restore
- B. Backup de bases de datos
 - a. backup mediante Dump Project Folder
 - b. backup mediante Self contained File
- C. Restauración de base de datos
 - a. Mecanismos de restauración
 - b. Dump Project Folder
 - c. Self Contained File
- D. Prácticas

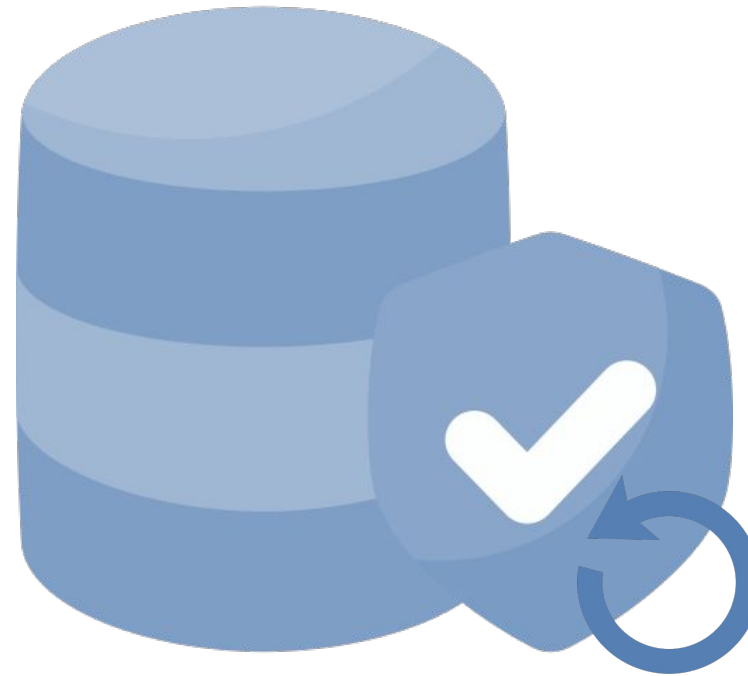


Administración de bases de datos

Administración de bases de datos

Ya hemos hablado anteriormente de la importancia de los datos hoy en día. En principio, la información se almacenaba solo por obligaciones impositivas de mantener un histórico de operaciones.

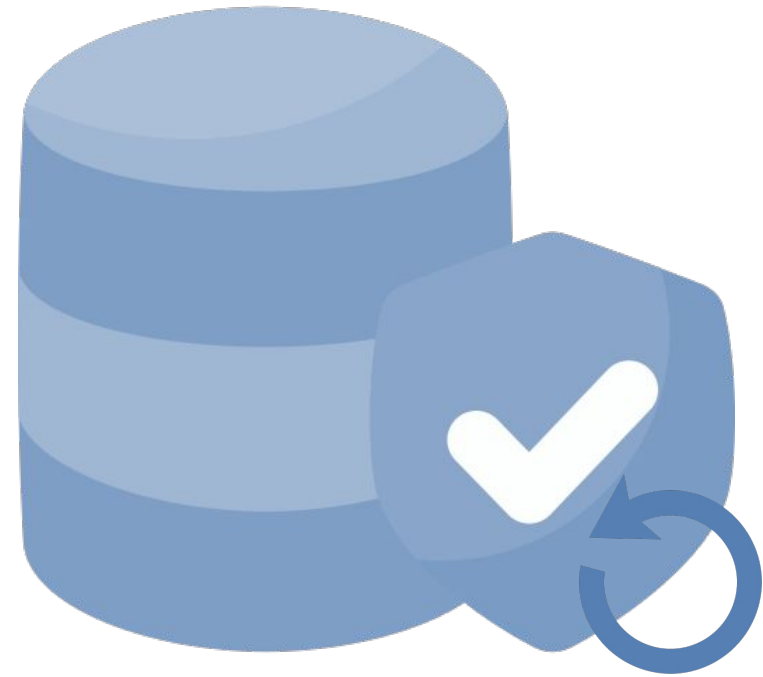
Desde hace algunos años, se descubrió que los datos “*pueden hablar*” y/o representar la información como métricas. Y estas métricas pueden decirnos, anticiparnos, o ayudarnos a tomar decisiones de una forma más asertiva.



Administración de bases de datos

Y más allá de que nuestro rol principal se enfoque en el desarrollo de software, en nuestro día a día vamos a tener la necesidad de interactuar con datos todo el tiempo.

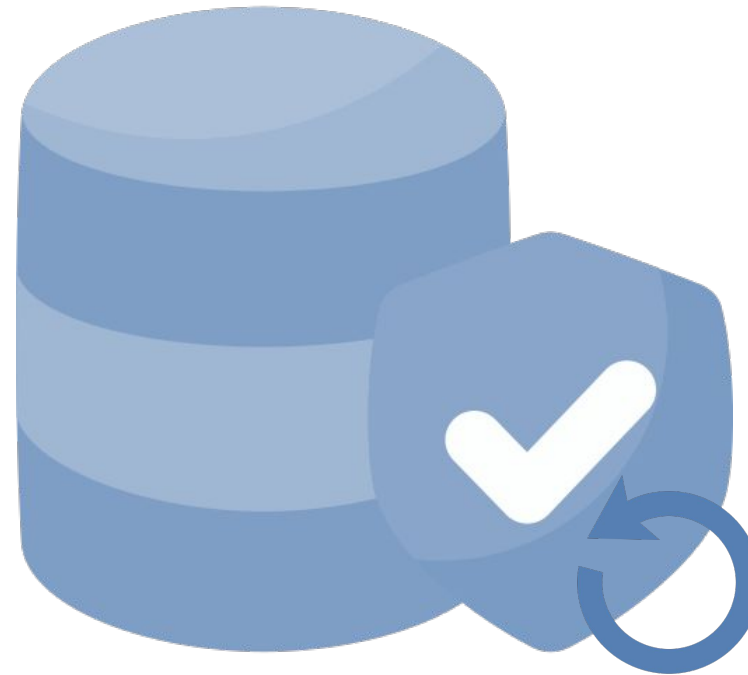
Y como parte de esta tarea, tener en cuenta la forma más apropiada de realizar backups y restauración de la información, es un punto que no podemos pasar por alto.



Administración de bases de datos

Además, la importancia de resguardar los datos también aplica porque toda información digital es uno de los activos más valiosos.

Por ello, hablaremos en este encuentro sobre la importancia y fundamentos de backup y restauración de bases de datos, y así garantizar la seguridad y disponibilidad continua de la información generada.

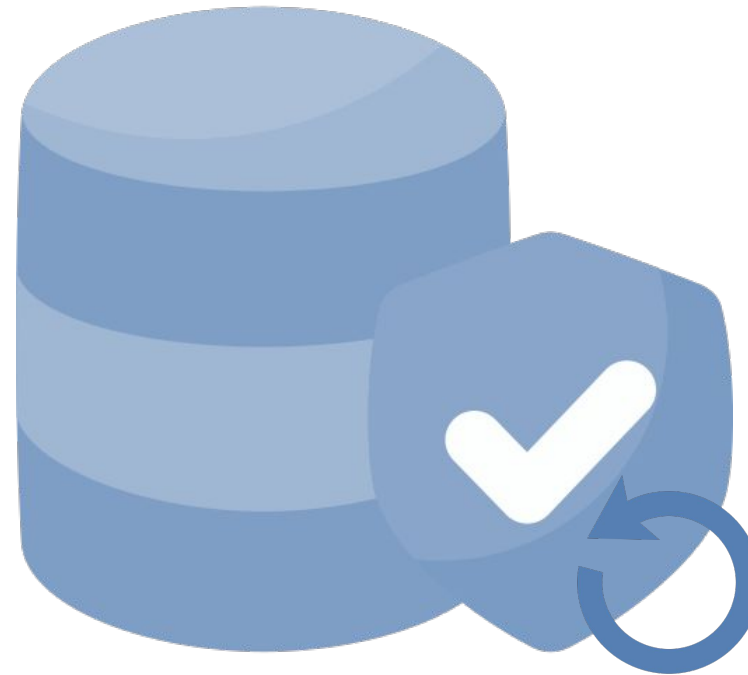


Backup y Restauración

Administración de bases de datos

La importancia del **backup y restauración** radica en su capacidad para **prevenir la pérdida de datos** debido a **errores humanos**, fallas de **hardware**, **ataques** cibernéticos o **desastres naturales**.

Contar con una estrategia sólida de backup y restauración nos **ayuda a minimizar el impacto de estos imprevistos**.

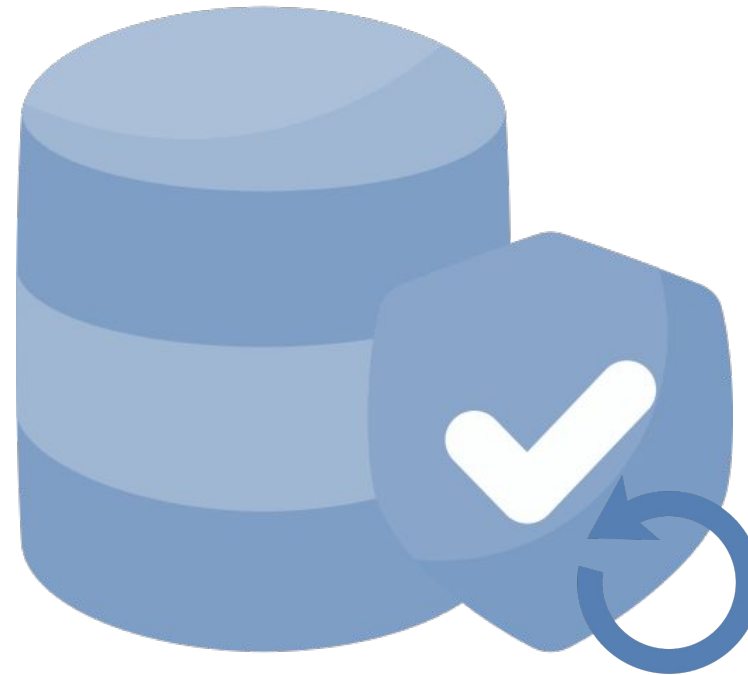


Administración de bases de datos

Dentro de las ventajas destacables en el terreno de backup y restauración de la información, podemos destacar:

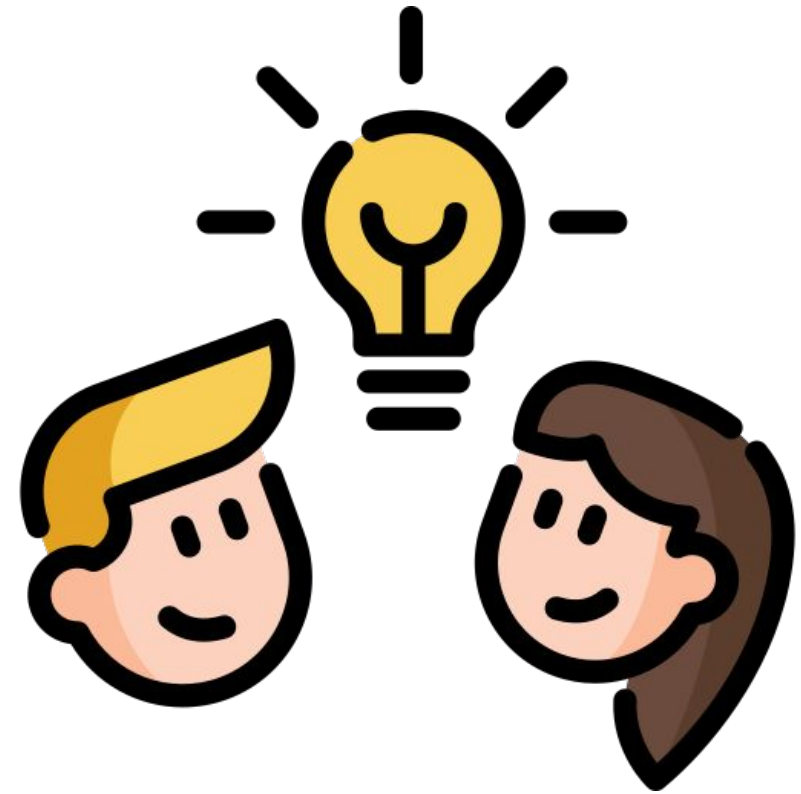
- Disponibilidad y continuidad del negocio
- Facilitar recuperación rápida y eficiente
- Brindar tranquilidad con datos protegidos

Entre otros tantos beneficios.



Administración de bases de datos

Además, en la actualidad, las nubes de datos ayudan mucho en este terreno, facilitando copias continuas de la información generada en estos datacenters, y almacenando la misma de forma espejada (*varías copias en diferentes lugares*).

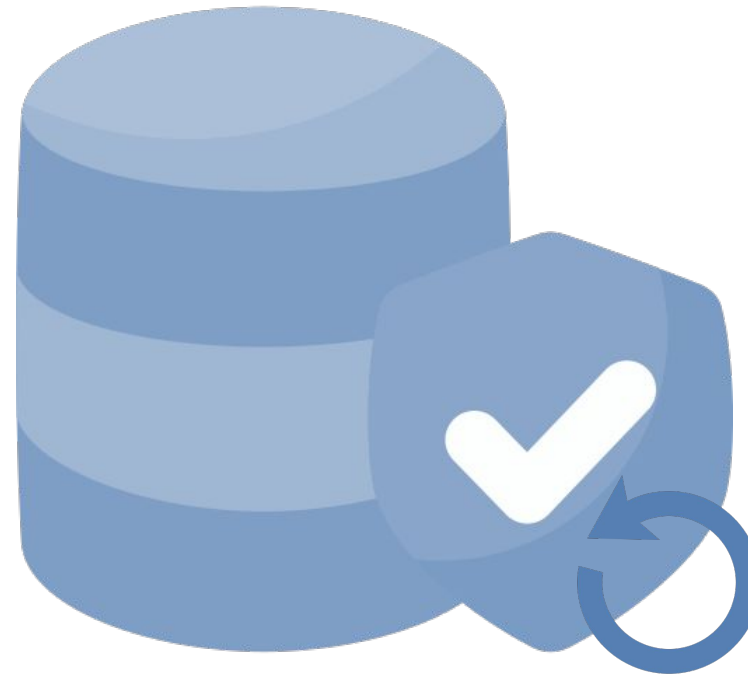


Análisis previo

Administración de bases de datos

Para realizar una copia de seguridad de la información que utilizamos de manera frecuente, lo primero que debemos considerar son los siguientes factores:

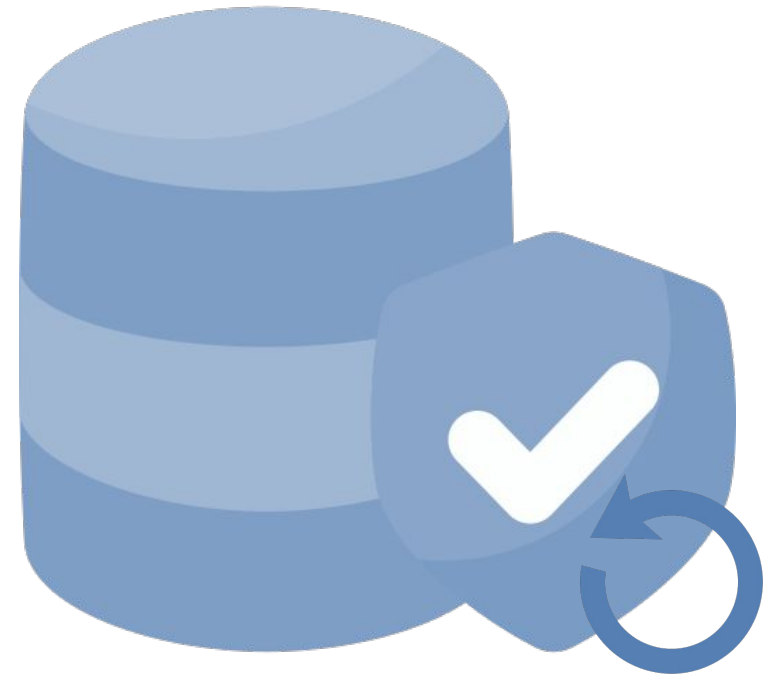
- La frecuencia y programación de los backups
- El resguardo físico de las copias de seguridad
- Verificar regularmente la integridad de los backups
- Documentar claramente los procedimientos de copia de seguridad y restauración



Administración de bases de datos

Resumiendo lo visto hasta aquí, podemos decir que el backup y restauración de bases de datos MySQL es una práctica esencial para garantizar la protección y disponibilidad de nuestros valiosos datos. Contar con un plan sólido y seguir buenas prácticas, nos ayuda a minimizar riesgos y asegurar la continuidad de nuestras operaciones.

Exploremos a continuación las estrategias de backup y restauración, herramientas, y mejores prácticas recomendadas para lograr una gestión efectiva de las bases de datos MySQL.



**Migrar un modelo
nosql hacia un
modelo relacional**

Data Manipulation Language

Pero cuando se trata de datos en cantidades notables donde estos crecerán de forma continua todo el tiempo, y que mucha de esta información se repetirá constantemente, allí el modelo de datos SQL es el modelo de información más asertivo para administrar todo este cúmulo de información de la manera más apropiada.



Data Manipulation Language

Y si nos posicionamos en algún ejemplo de datos específico, donde el volumen de la información seguramente se repita de forma frecuente, allí podremos entender que, para estos tipos de datos repetitivos, el modelo relacional será el que mejor se adapte para minimizar el crecimiento de esta información a futuro.

```
JSON
{
  "id": 17,
  "poster": "./posters/17.jpg",
  "titulo": "Halt and Catch Fire",
  "categoria": "Serie",
  "genero": "Drama",
  "tags": "Ficción, Drama, Tecnología",
  "busqueda": "Halt and Catch Fire, Ficción, Drama, Tecnología, Lee Pace, Scoot McNairy, Mackenzie Davis, Kerry Bishé, Toby Huss, Alana Cavanaugh",
  "temporadas": "4",
  "reparto": "Lee Pace, Scoot McNairy, Mackenzie Davis, Kerry Bishé, Toby Huss, Alana Cavanaugh",
  "trailer": "https://www.youtube.com/embed/pWrIoRji60A"
}
```



Migrar un modelo nosql hacia un modelo relacional

Como vimos con el ejemplo anterior, migrar un modelo de datos NoSQL hacia un modelo de datos relacional (SQL) puede ofrecer varias ventajas significativas.

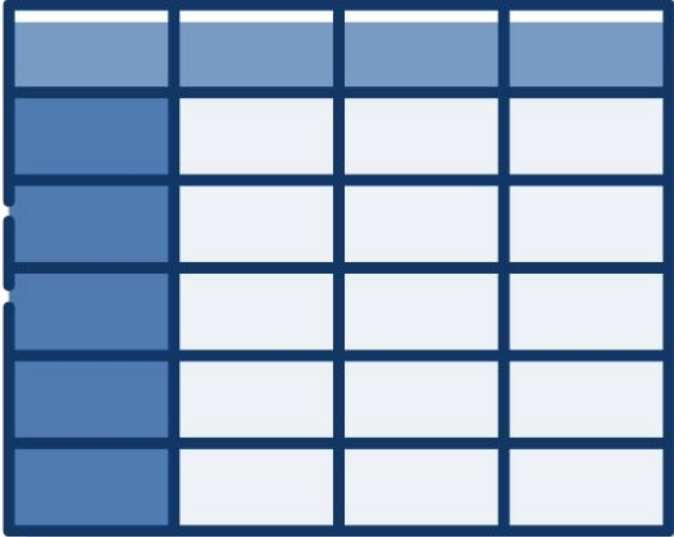
En un modelo relacional, sabemos que los datos se organizarán en tablas estructuradas con relaciones establecidas entre ellas, lo que permite un mejor control y coherencia de los datos.



Migrar un modelo nosql hacia un modelo relacional

Este mejor control y con una coherencia en los datos almacenados, nos facilita poder realizar consultas complejas y análisis más profundos (*métricas*).

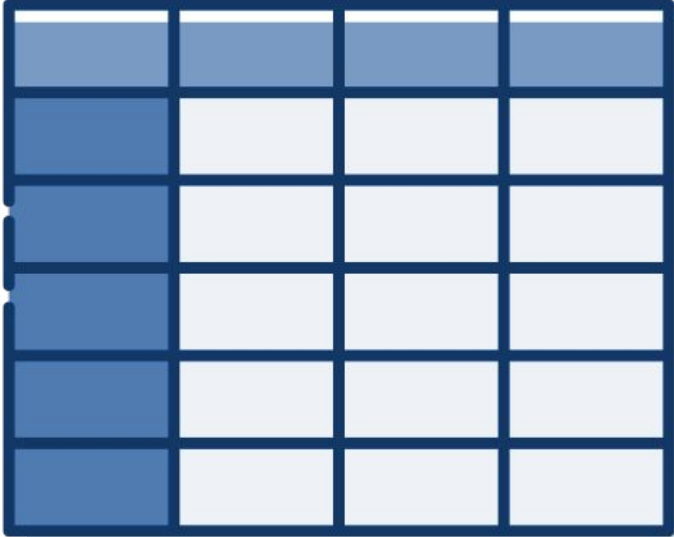
Además, los sistemas de gestión de bases de datos relacionales (RDBMS) ofrecen características como transacciones ACID, que aseguran la integridad y consistencia de los datos.



Migrar un modelo nosql hacia un modelo relacional

Esto es especialmente útil en aplicaciones empresariales donde la confiabilidad y la precisión son fundamentales. Otro beneficio clave es la capacidad de realizar consultas ad hoc y utilizar el lenguaje SQL estándar, lo que facilita el acceso y la manipulación de los datos.

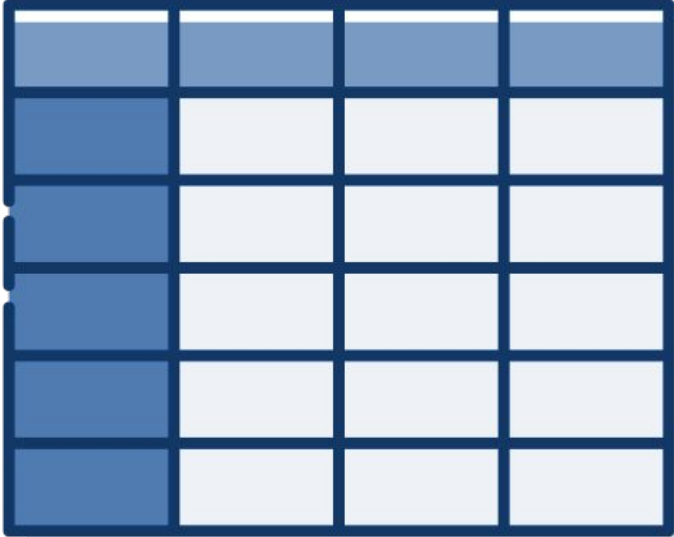
Además, los RDBMS tienen una amplia gama de herramientas y soporte, lo que simplifica el mantenimiento y la escalabilidad del sistema.



Migrar un modelo nosql hacia un modelo relacional

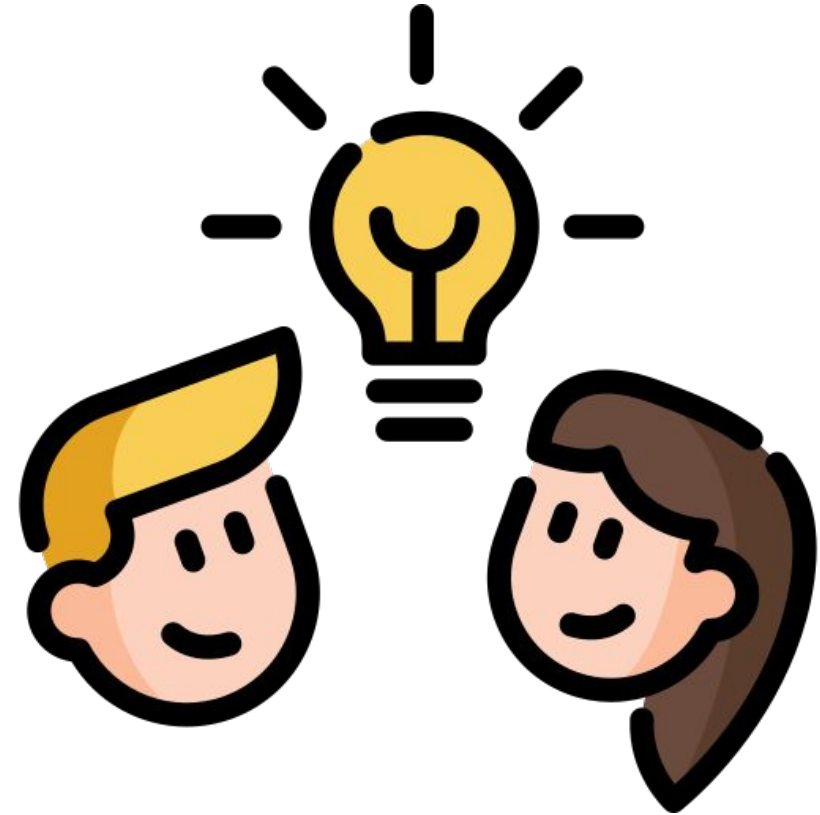
Esto es especialmente útil en aplicaciones empresariales donde la confiabilidad y la precisión son fundamentales. Otro beneficio clave es la capacidad de realizar consultas ad hoc y utilizar el lenguaje SQL estándar, lo que facilita el acceso y la manipulación de los datos.

Además, los RDBMS tienen una amplia gama de herramientas y soporte, lo que simplifica el mantenimiento y la escalabilidad del sistema.



Migrar un modelo nosql hacia un modelo relacional

En resumen, migrar a un modelo de datos relacional ofrece mayor control, consistencia, confiabilidad y facilidad de acceso a los datos, lo que puede ser beneficioso para aplicaciones empresariales y análisis de datos más complejos.



Backup de bases de datos

Backup de bases de datos

Al inicio del trabajo con bb.dd hablamos oportunamente de que MySQL cuenta con diversas **Herramientas de Consola**, o ventana **Terminal**, que nos permiten realizar diferentes tareas; entre ellas, administrar políticas de backup.

mysqldump es el nombre de esta herramientas, y forma parte del motor de base de datos para gestionar copias de seguridad.

```
$/> mysqldump
```



Backup de bases de datos

Debemos definir el comando **mysqldump**, seguido de los parámetros **-u** (*usuario*) **-p** (*contraseña*) y seguido a ello el nombre de la bb.dd o esquema que deseamos respaldar.

Por último, definimos la ruta de destino donde realizaremos el backup en cuestión.

```
mysql Dump  
mysqldump -u [usuario] -p [contraseña] [nombre_de_la_base_de_datos] >  
[ruta_del_archivo_de_backup.sql]
```



Backup de bases de datos

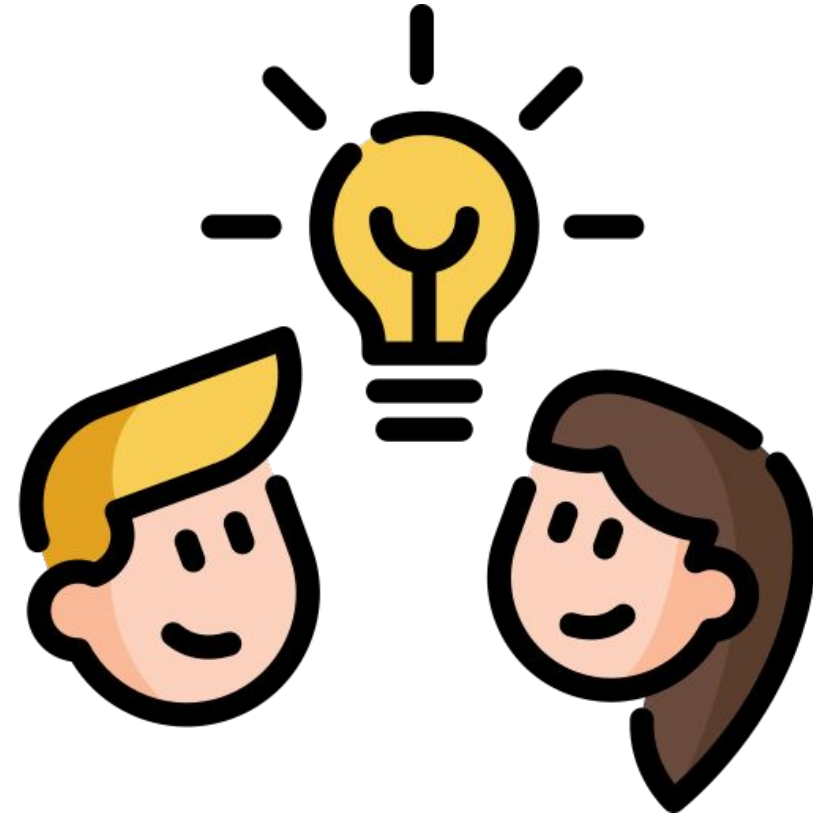
Todo backup que realicemos de una bb.dd siempre se realiza sobre un archivo con extensión **.sql**.

El proceso habitual de backup es básicamente generar la estructura de la bb.dd con sus relaciones y valores por defecto y, a continuación, la referencia de inserción de registros de acuerdo al total de datos almacenados.

```
Mysql Dump  
mysqldump -u root -p mydatabase > /ruta/al/archivo/backup.sql
```

Backup de bases de datos

Si bien, hoy tenemos varios mecanismos modernos y gráficos para realizar estos procedimientos, no siempre tendremos entornos de bases de datos que nos permitan acceder a su información a través de aplicaciones RDBMS.



Backup y MySQL Workbench

Si tenemos acceso a las bases de datos que debemos resguardar utilizando MySQL Workbench, este gestor nos brinda herramientas efectivas para poder llevar a cabo estos procesos más cómodamente.

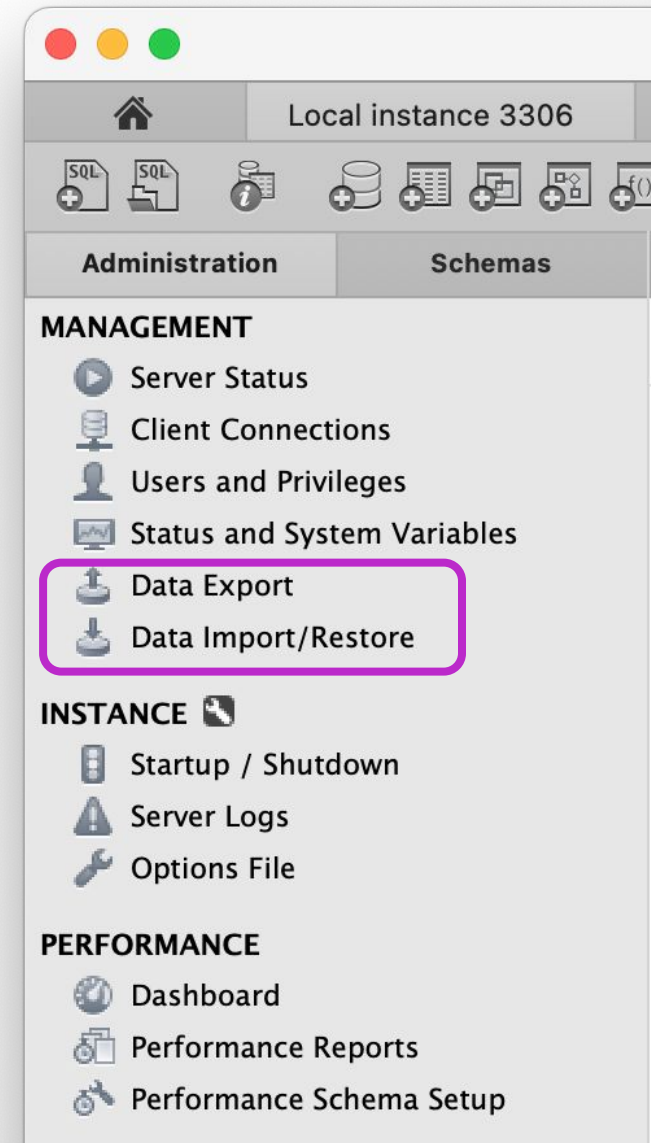


Backup desde MySQL Workbench

Backup y MySQL Workbench

En el apartado **Administration**, encontraremos un set de herramientas para acceder a los procesos de backup y restauración.

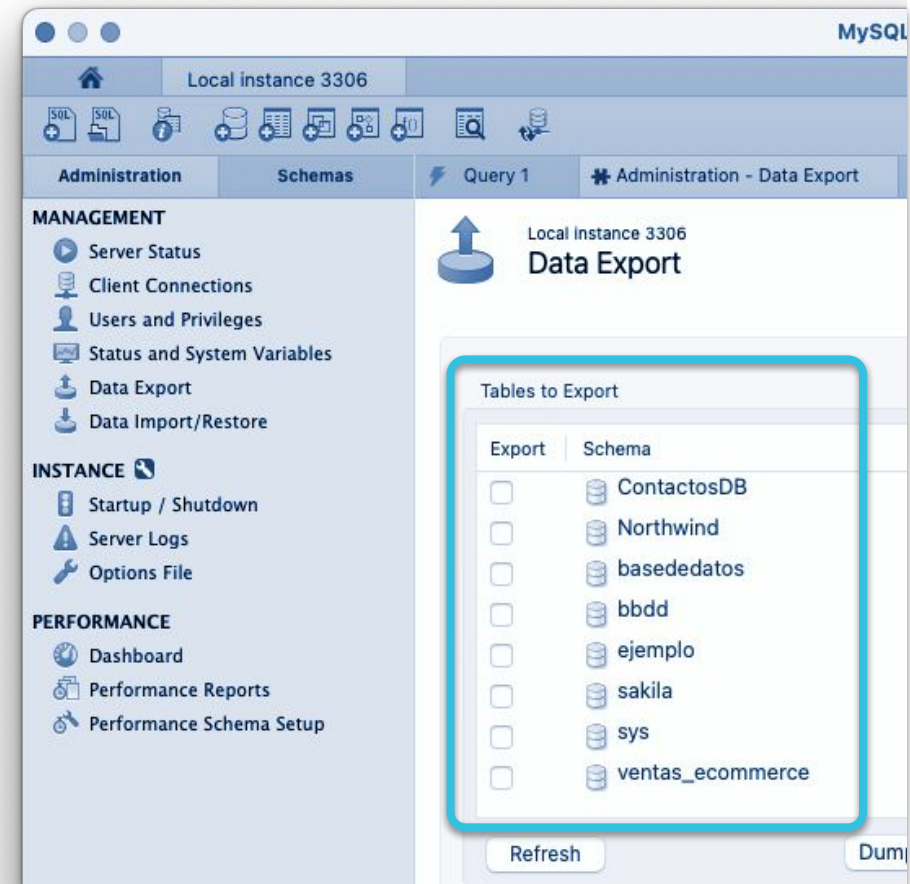
Los mismos se encuentran en las secciones **Data Export** y **Data Import/Restore**.



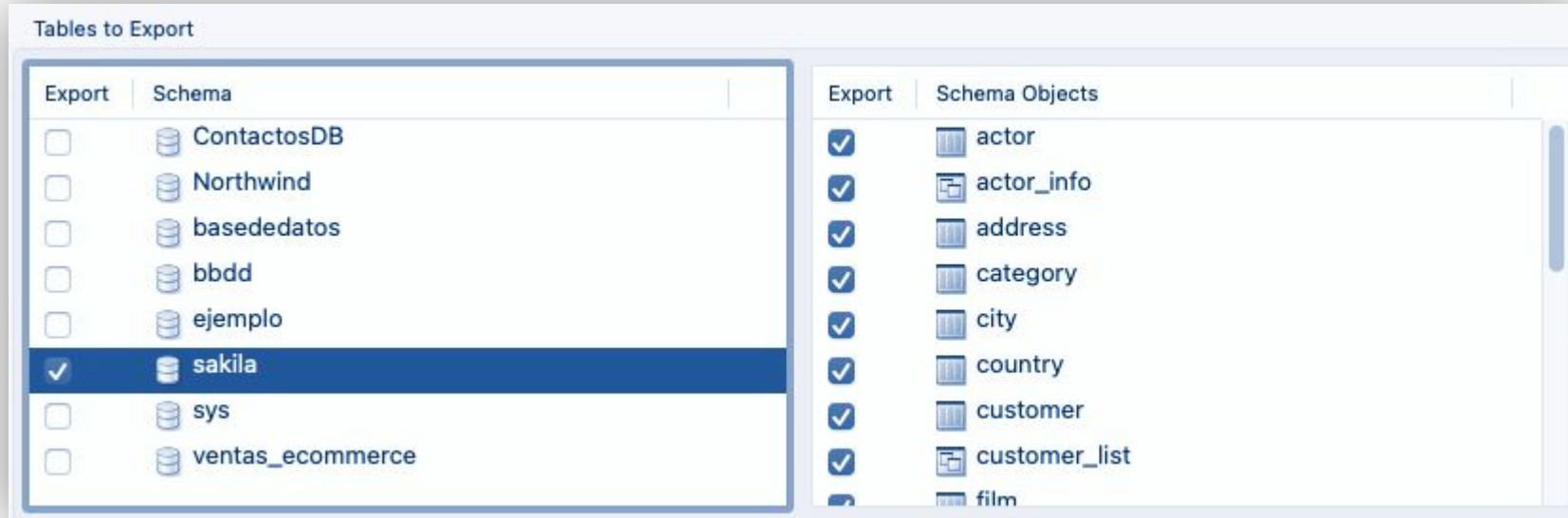
Backup y MySQL Workbench

Cuando ingresamos a la sección **Data Export**, la misma despliega un panel donde podremos ver todas las bb.dd (*esquemas*) disponibles para realizar copias de seguridad.

El proceso en sí puede aplicarse solo a una bb.dd o a varias, en simultáneo.

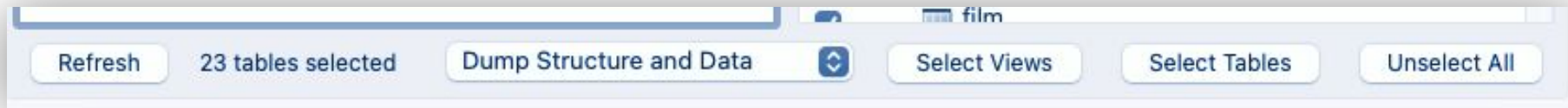


Backup y MySQL Workbench



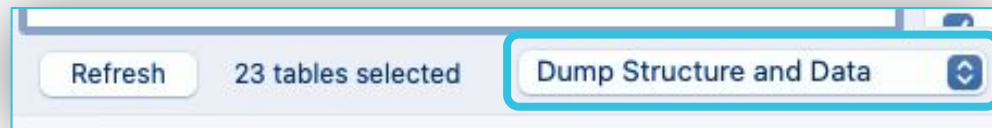
Al seleccionar una base de datos, podremos visualizar en el panel derecho el listado de **Tablas** y **Vistas** de la misma. A su vez, podemos seleccionar todas las tablas y/o vistas que consideremos necesario incluir en el backup, tildando y/o destildando cada casillero lateral.

Backup y MySQL Workbench



Los botones inferiores a los paneles nos permiten seleccionar las **vistas, tablas, deseleccionar todo, refrescar** su contenido (*por si algo cambió por afuera*), y seleccionar qué información vamos a respaldar.

Backup y MySQL Workbench



Tenemos varias formas de definir el tipo de copia de respaldo que deseamos hacer.

Dump Structure and Data: realiza una copia de seguridad de los datos, y de la estructura de la base de datos.

Dump Data Only: realiza una copia de seguridad de los datos de la base de datos.

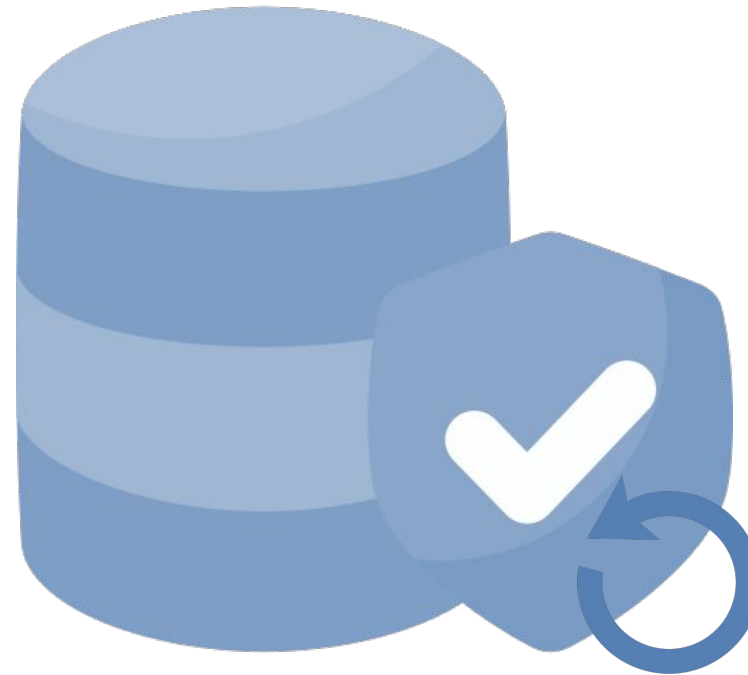
Dump Structure Only: realiza una copia de seguridad de la estructura de la base de datos.

¿Cuándo utilizar las diferentes opciones de backup?

Dump Data Only

Si la base de datos es de uso constante y su mantenimiento es esporádico, podemos elegir esta opción la cual resguarda solo los datos almacenados en ésta.

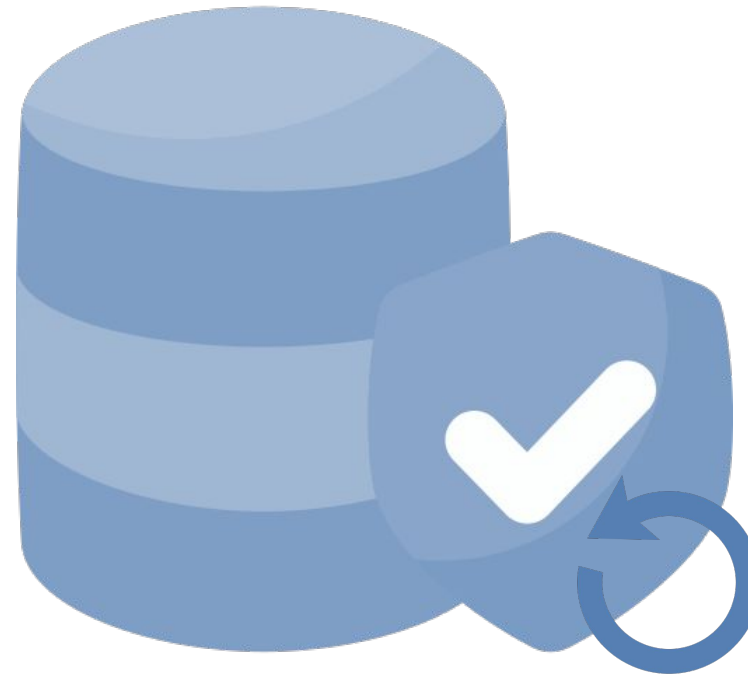
Si es una base de datos de mucha concurrencia, la opción más apropiada de backup de datos será todos los días.



Dump Structure Only

Esta opción se utiliza cuando implementamos nuevas **tablas, vistas, funciones y procedimientos almacenados**, o cuando cambiamos la estructura de algunos de los objetos de la bb.dd.

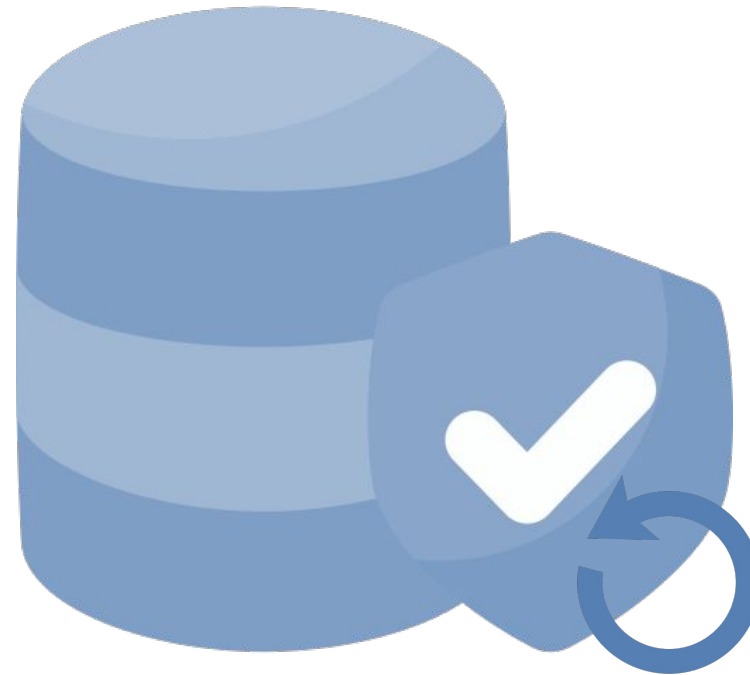
Su uso es esporádico; solo cuando ocurre algún evento de los aquí mencionados.



Dump Data And Structure

Si la base de datos está en un ambiente que no es el de **Producción**, o una base de datos personal donde realizamos poca interacción con la información y/o modificación de su estructura, podemos pensar en utilizar este mecanismo de backup.

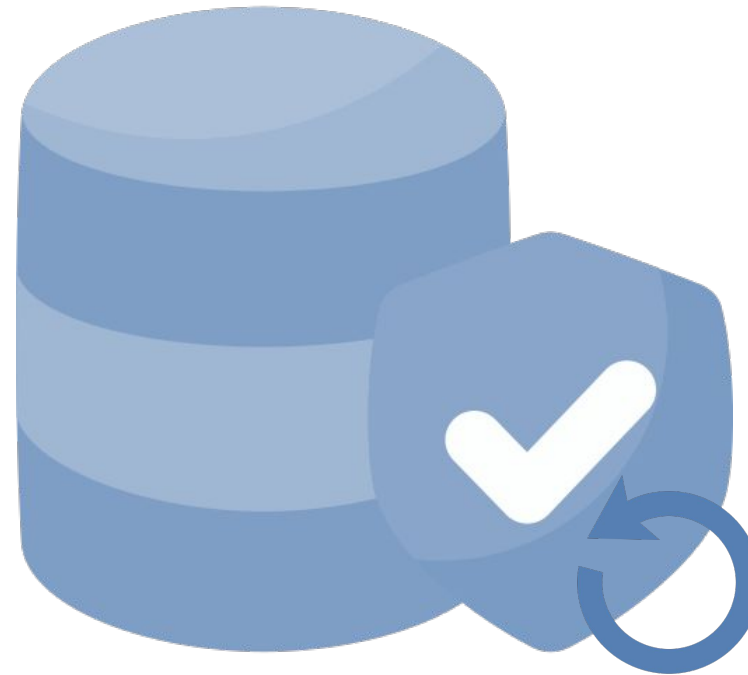
Su uso también puede ser esporádico, ya que es el más lento de todos, porque resguarda toda la información.



Analizar un contenido a migrar

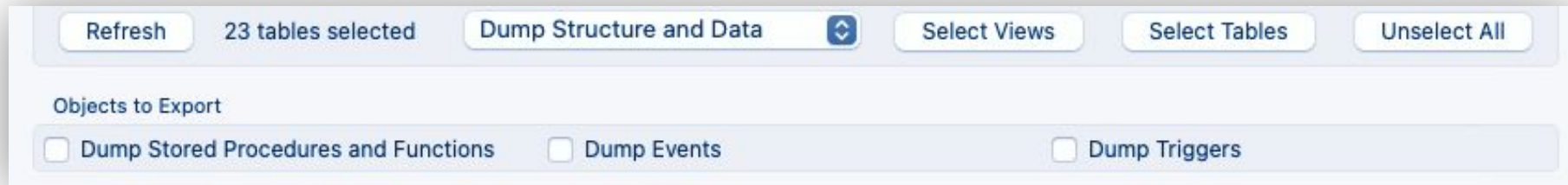
Si la base de datos es de uso constante y su mantenimiento es esporádico, podemos elegir esta opción la cual resguarda solo los datos almacenados en ésta.

Si es una base de datos de mucha concurrencia, la opción más apropiada de backup de datos será todos los días.



Seleccionar los Objetos a exportar

Seleccionar los Objetos a exportar



Los botones inferiores a los paneles nos permiten seleccionar las **vistas, tablas, deseleccionar todo, refrescar** su contenido (*por si algo cambió por afuera*), y seleccionar qué información vamos a respaldar.

Backup y MySQL Workbench



Finalmente queda elegir el tipo de backup a realizar. Entre las opciones tenemos:

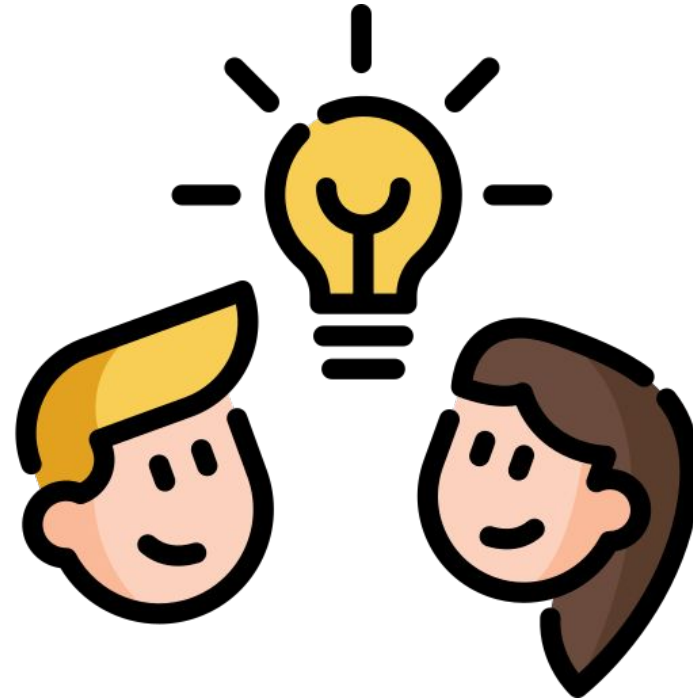
- **Export to Dump Project Folder**, vuelca a una carpeta de proyecto la(s) base(s) de datos
- **Export to Self-Contained File**, genera un archivo **.sql** con la información seleccionada

Esta opción permite seleccionar la ruta hacia la carpeta donde queremos guardar el backup. Cada objeto Tabla se exportará en un archivo separado. Esto nos garantiza que, de tener que restaurar, podremos seleccionar cuál(es) tabla(s) recuperar.

Backup y MySQL Workbench

Si la base de datos se encuentra en un servidor, recomendamos realizar la copia de datos local y luego, dicha carpeta, sea trasladada a un disco de red, disco externo, Cloud Storage, o cualquier otro medio de almacenamiento seguro.

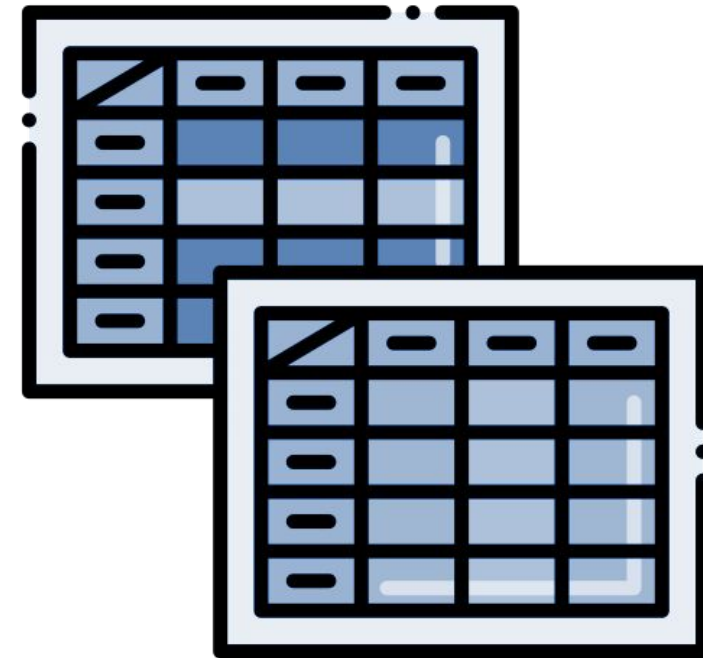
El backup local permitirá reducir los tiempos de este proceso.



Backup y MySQL Workbench

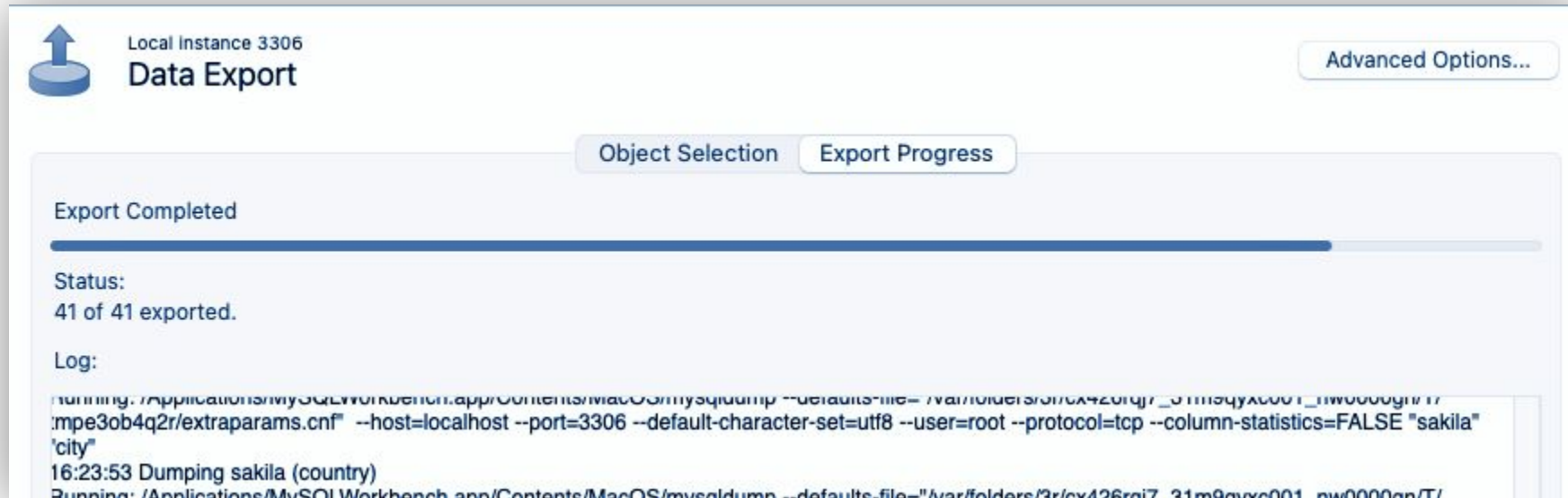
Y, cada carpeta donde realicemos un nuevo backup, debemos nombrarla con la fecha del día, y hasta incluir la hora de realización del backup.

Esto facilitará ubicar rápidamente un backup antiguo en un disco que almacena el historial de backups.



El proceso de Backup

El proceso de Backup

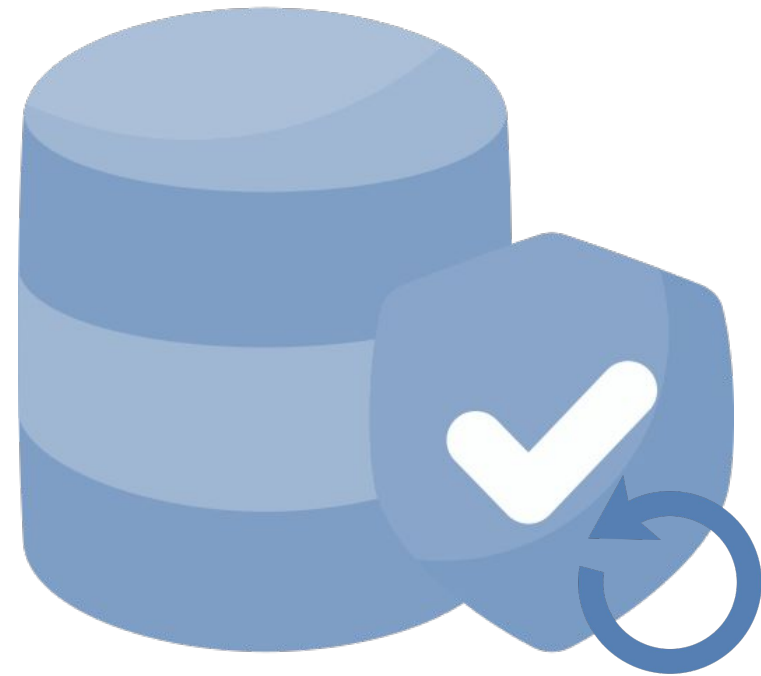


Cuando pulsamos el botón **Iniciar Proceso**, este comenzará a realizar el backup de la o las bases de datos y objetos seleccionados. Mediante la Barra de progreso y la ventana de LOG, podemos seguir el avance. Su tiempo de demora será acorde al tamaño de la información seleccionada para respaldar.

El proceso de Backup

```
21:55:52 Export of /Users/developer/dumps/Dump20230611.sql has finished
```

Dentro del LOG encontraremos un mensaje similar a este, cuando el backup ha llegado a su fin. Solo nos queda consultar la carpeta seleccionada para verificar que el o los archivo(s) haya(n) sido creado(s).



Restauración de base de datos

Restauración de base de datos

El proceso de restauración de bases de datos de la mano de MySQL Workbench es similar al que realizamos recientemente.

Las interfaces gráficas nos ayudan a trabajar más rápido y mejor en cuanto a selección de los recursos a restaurar. Veamos entonces, qué nos ofrece MySQL Workbench a través de la sección **Data Import/Restore**.



Restauración de base de datos

Ingresamos al apartado **Administration > Data Import/Restore**, donde encontraremos un panel similar a **Data Export**.

El proceso que debemos realizar a continuación, es justamente el inverso al realizado anteriormente, de cara a poder recuperar los datos u objetos de forma total o parcial en alguna base de datos existente.



Restauración de base de datos

El panel de restauración tiene una interfaz similar al anterior. Aquí se concentran dos puntos específicos, desde donde debemos seleccionar qué tipo de backup deseamos recuperar.

Si realizamos un backup a través del método **Dump Project Folder**, debemos tildar entonces la misma opción y elegir la carpeta de origen desde donde restauraremos la copia de seguridad.

Local Instance 3306
Data Import

Import from Disk Import Progress

Import Options

☒ Import from Dump Project Folder

Load Folder Contents

☐ Import from Self-Contained File

Select the SQL/dump file to import. Please note that the whole file will be imported.

Default Schema to be Imported To

Default Target Schema: New... The default schema to import the dump into. NOTE: this is only used if the dump file doesn't contain its schema, otherwise it is ignored.

Select Database Objects to Import (only available for Project Folders)

Import	Schema
<input type="checkbox"/>	<input type="checkbox"/>

Dump Structure and Data ☒ Select Views Select Tables Unselect All

Press [Start Import] to start... Start Import

Restauración de base de datos

En cambio, si buscamos recuperar un backup desde el método **Self-Contained File**, debemos seleccionar esta otra opción y elegir a continuación el archivo **.sql** que contiene el backup de la base de datos.

En ambos casos, disponemos de los botones laterales que nos permiten navegar dentro del sistema de archivos de la unidad donde estos se encuentren alojados.

Local Instance 3306
Data Import

Import from Disk Import Progress

Import Options

☒ Import from Dump Project Folder ..
Select the Dump Project Folder to import. You can do a selective restore.
Load Folder Contents

☐ Import from Self-Contained File ..
Select the SQL/dump file to import. Please note that the whole file will be imported.

Default Schema to be Imported To

Default Target Schema: New...
The default schema to import the dump into.
NOTE: this is only used if the dump file doesn't contain its schema, otherwise it is ignored.

Select Database Objects to Import (only available for Project Folders)

Import Schema

Import Schema Objects

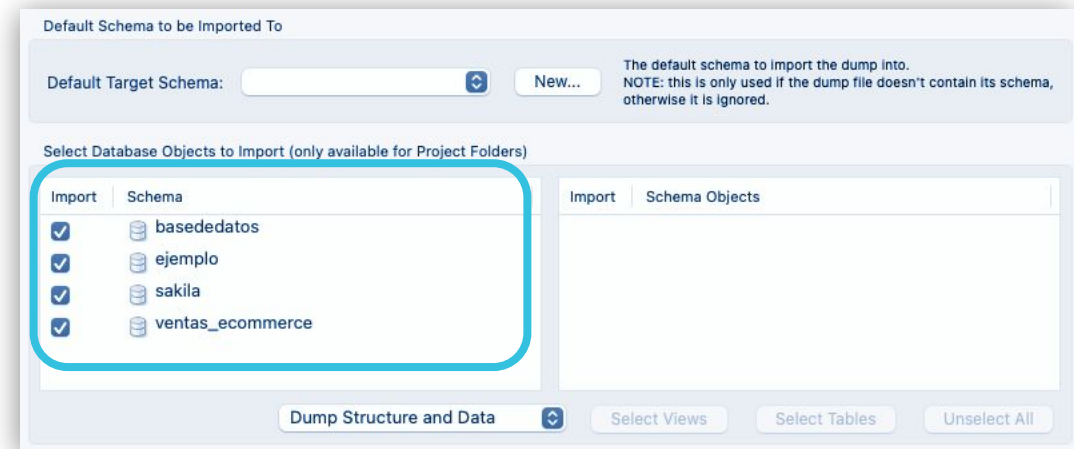
Dump Structure and Data ☒ Select Views Select Tables Unselect All

Press [Start Import] to start... Start Import

Restauración de base de datos

El paso siguiente será validar de todas las posibles bases de datos elegidas anteriormente para formar parte del backup, cuál de ellas necesitamos para restaurar un backup específico.

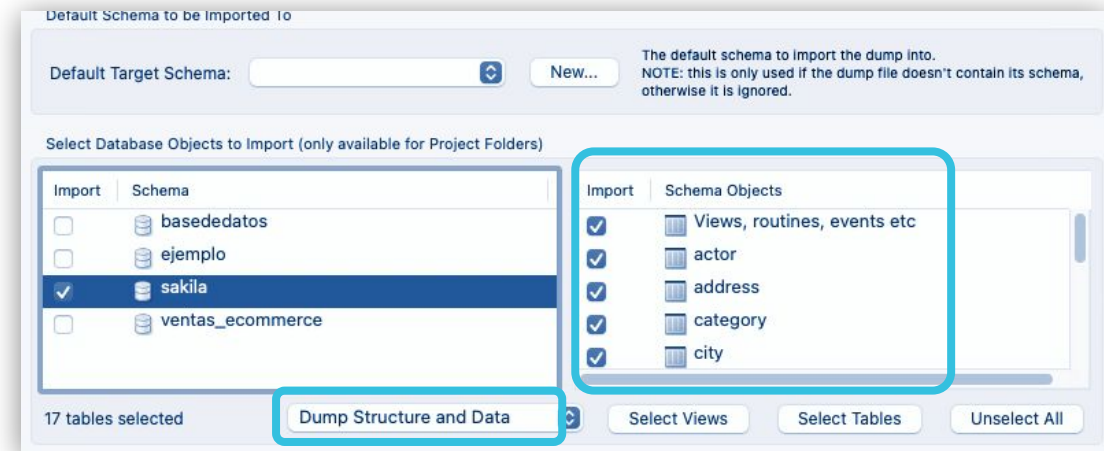
Podemos dejar a todas las opciones tildadas o solo seleccionar aquella o aquellas base(s) de datos desde donde deseamos recuperar la información.



Restauración de base de datos

Al elegir una base de datos, veremos todos los objetos de esta y podremos seleccionar de forma parcial o total cuáles de ellos queremos restaurar.

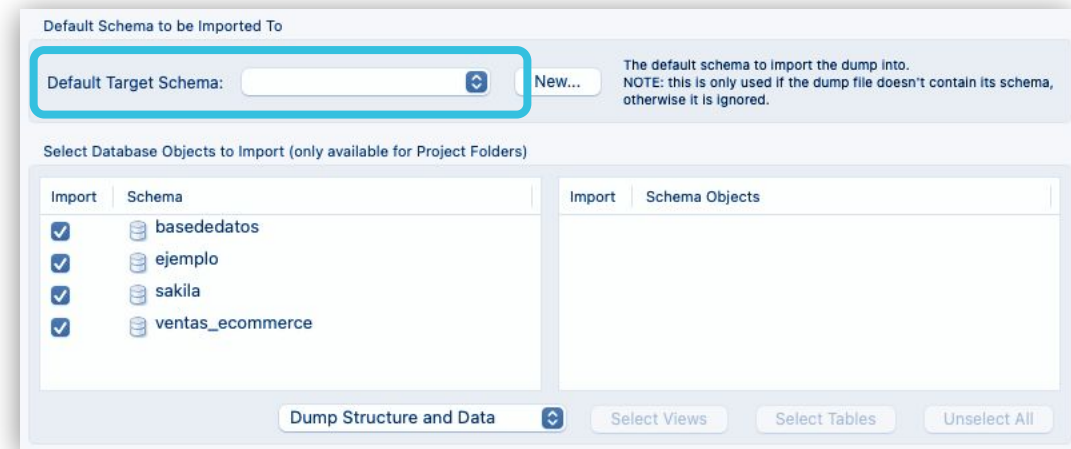
Debajo también podemos elegir cuál es el proceso de restauración (*solo su estructura, solo sus datos, o estructura y datos*).



Restauración de base de datos

Si vamos a trabajar con un solo esquema, debemos elegir el esquema de destino desde el apartado superior, para saber dónde se restaurará la información en cuestión.

Más allá de que debamos restaurar información de más de una bb.dd, recomendamos siempre realizar el proceso de a una base de datos por vez, para así tener un mejor control de los resultados de restauración de datos.

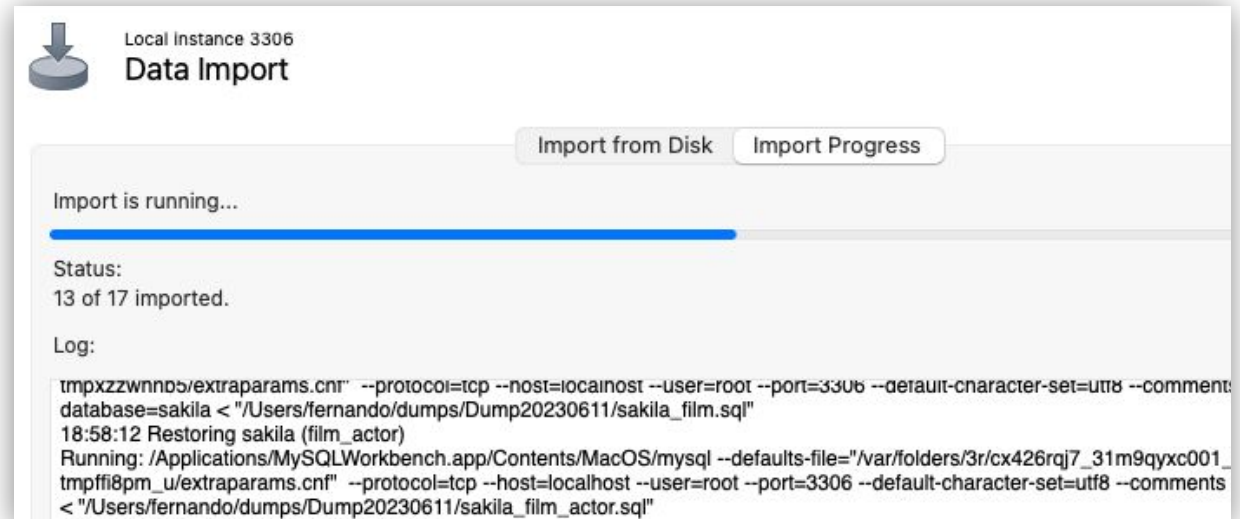


Iniciar Restauración

Iniciar Restauración

Finalmente, ajustados todos los parámetros y seleccionados todos los objetos involucrados, pulsamos el botón **Start Import** para que dé inicio al proceso de restauración.

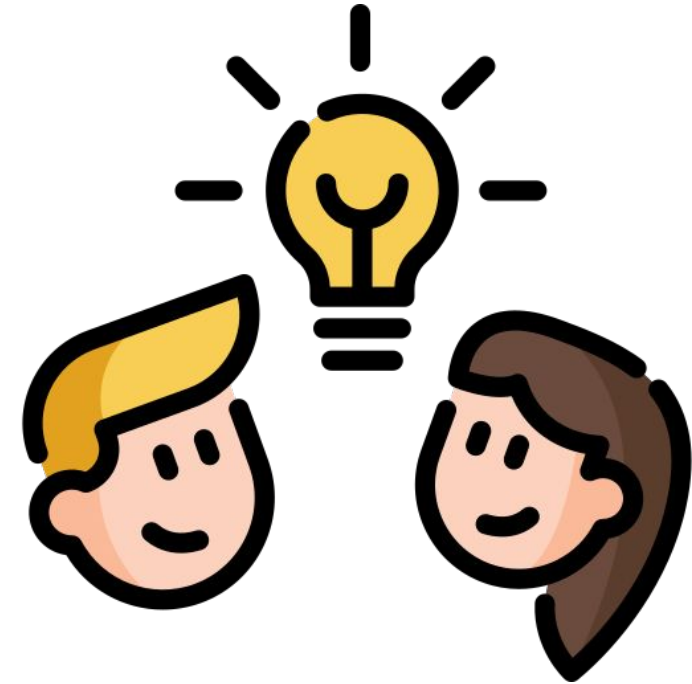
Veremos el progreso del mismo en el mismo panel donde estamos trabajando.



Iniciar Restauración

Los tiempos de recuperación pueden ser muy diferentes en cada caso, dependiendo del procesador, RAM, velocidad del disco de almacenamiento, tamaño del archivo de base de datos, etcétera.

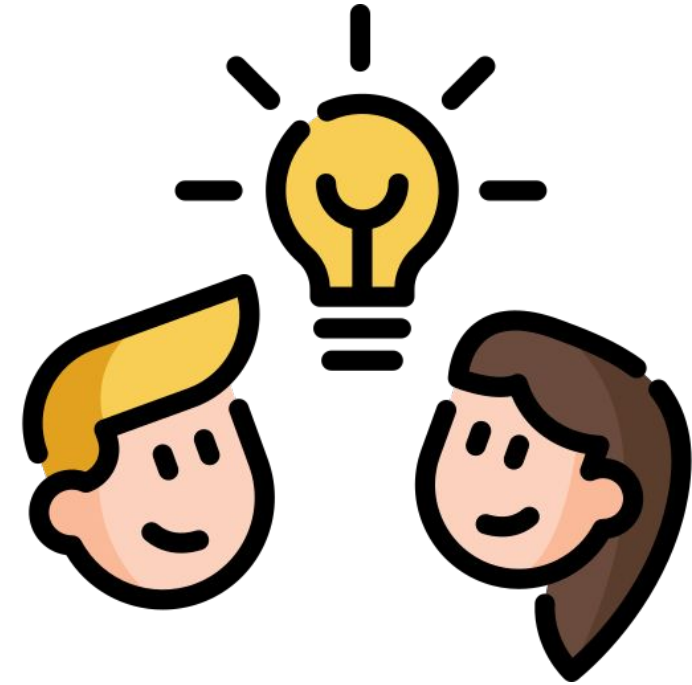
En este último caso, a mayor tamaño de la base de datos, mayor será el tiempo estimado en recuperar la información respaldada.



Iniciar Restauración

Y como escenario ideal para recuperar datos, recomendamos que el motor de base de datos esté dedicado a este proceso. Evita que la base de datos esté brindando acceso a otros usuarios y/o aplicaciones.

El escenario ideal es poner el servidor en mantenimiento, así podrás acortar los tiempos de recuperación de la información.



Sección práctica

Tomaremos la base de datos Northwind para implementar en ella un proceso de Backup.

Realizaremos ambos pasos (backup - restauración), sobre su estructura y datos.



Prácticas

Practica con la bb.dd. Northwind instalada en Mysql Workbench. Realiza las siguiente tareas:

- Prepara el ambiente para realizar una copia de seguridad
- Elige la opción **Dump to Self-Contained File**
- Inicia el proceso y espera a que finalice el mismo
- Ya finalizado, Realiza la restauración del mismo con la herramienta **Data Import/Restore**



Prácticas

Ya finalizado, Realiza la restauración del mismo con la herramienta **Data Import/Restore**.

- Elimina primero la bb.dd Northwind instalada en tu motor de base de datos
- Crea un nuevo Esquema con el mismo nombre, y agrega en éste la sigla “v2.0”
Ejemplo: **NorthwindV2.0**
- Finalmente restaura el contenido de tablas, vistas y datos.



Muchas gracias.



Ministerio de Economía
Argentina

Secretaría de
Economía del Conocimiento

*primero
la gente*