

Plan de instalación, migración y respaldo - CodeProject.

Introducción, objetivos y justificación.

Introducción.

El lector se encontrará con un documento que tiene como propósito mostrar las actividades, los tiempos de ejecución y los responsables involucrados al momento de elaborar las migraciones y el proceso de respaldo de datos del sistema para el control de inventarios dirigido a la **Empresa Comercializadora de Ropa Infantil Violetas**.

Objetivo.

Elaborar un cronograma con las actividades, los tiempos de ejecución y los responsables de las actividades de migración y respaldo de datos.

Objetivos específicos.

- Identificar cuáles son los responsables de los planes de migración, respaldo e instalación.
- Detallar, por medio de una tabla, las fechas, actividades, tiempos de ejecución y los responsables.

Justificación.

Consideramos que elaborando el cronograma de actividades se facilita el proceso de documentación de software. Del mismo modo, se pondrá en evidencia los métodos con los que se desarrollaron las pruebas de migración, respaldo e instalación de datos.

Documentación exportación y migración de datos.

¿Qué es la exportación de datos?

La migración de datos consiste en la transferencia de materiales digitales de un origen de datos a otro, transformando la forma lógica del ente digital de modo que el objeto conceptual pueda ser restituido o presentado por un nuevo equipo o programa informático.



Exportación y migración en MySQL Workbench.

[/plan-de-instalacion-migracion-y-respaldo-codeproject./documentacion-exportacion-y-migracion-de-datos.-pendiente-/back-up-mysql-workbench.](#)



Exportación y migración en PostgreSQL.

[/plan-de-instalacion-migracion-y-respaldo-codeproject./documentacion-exportacion-y-migracion-de-datos.-pendiente-/back-up-postgresql.](#)



Exportación y migración en SQLite.

[/plan-de-instalacion-migracion-y-respaldo-codeproject./documentacion-exportacion-y-migracion-de-datos.-pendiente-/back-up-sqlite.](#)

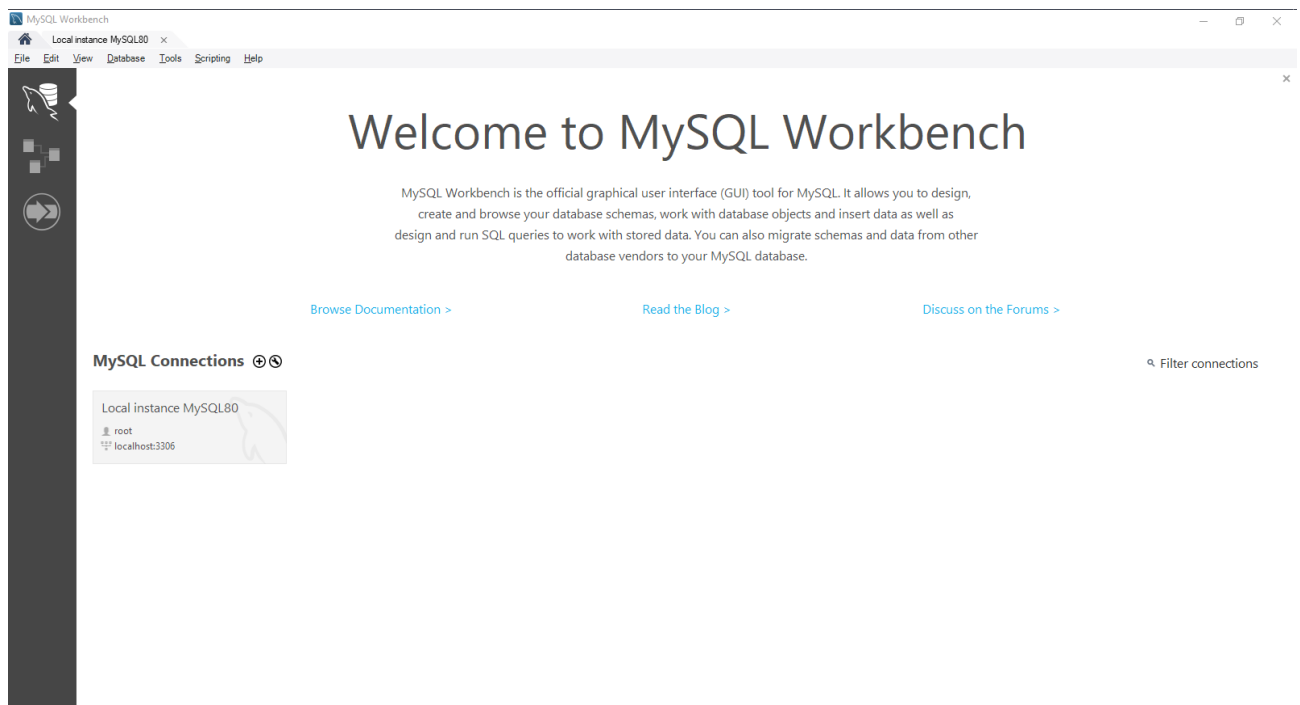
Exportación y migración en MySQL Workbench.

Exportación.

- El primer paso para hacer la exportación de datos de una base de datos a otra en **MySQL Workbench** es iniciar la instancia local.

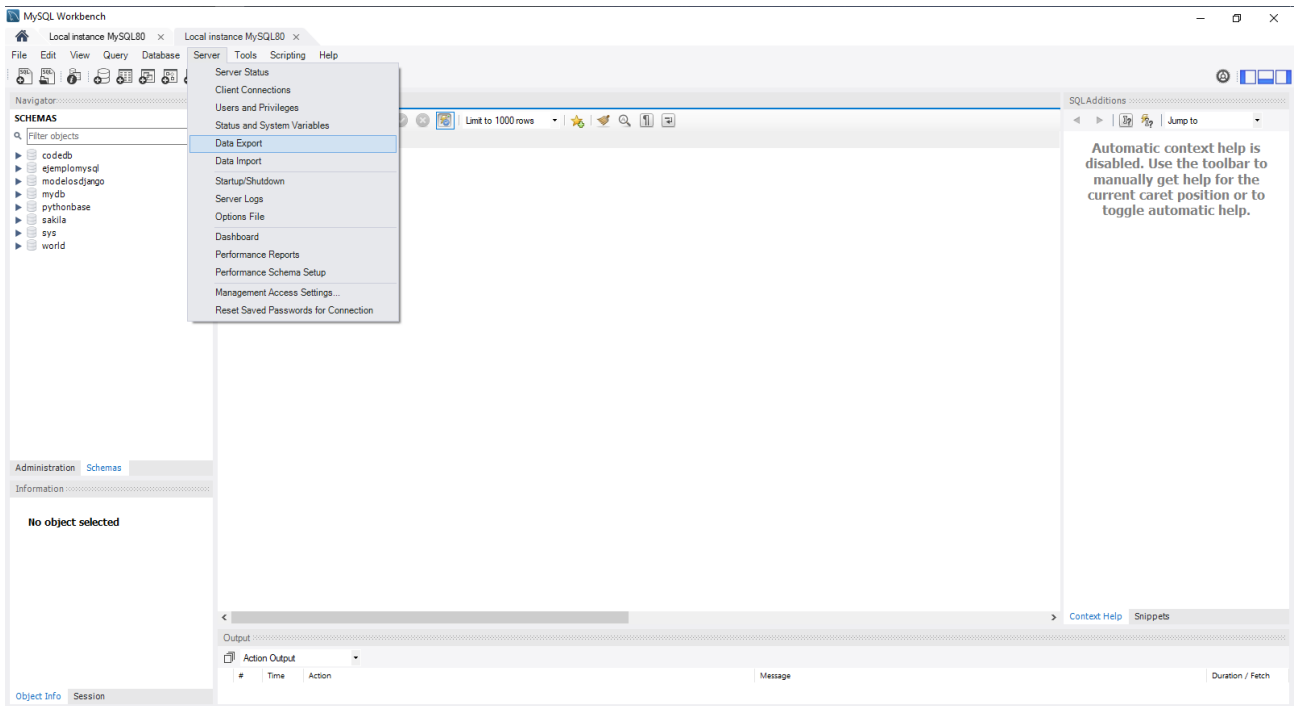


Una instancia es una copia del ejecutable que se ejecuta como un servicio de sistema operativo. Cada instancia administra varias bases de datos del sistema y una o varias bases de datos de usuario. Cada equipo puede ejecutar varias instancias.



Primer paso.

- Después de eso, el usuario debe ubicarse en el menú superior y seleccionar la opción **Server > Data Export**.



Segundo paso.

- El siguiente paso es seleccionar la base de datos de la que queremos exportar los datos y la estructura. Debemos asegurarnos de tener las siguientes opciones marcadas, la ubicación de la carpeta donde se guardará el archivo a exportar presente, y debemos dar 'clic' en ***Start Export***.



Es aconsejable tomar como ejemplo la siguiente captura.

Query 1 Administration - Data Export x

Local instance MySQL80

Data Export

Advanced Options...

Object Selection Export Progress

Tables to Export

Exp...	Schema
<input type="checkbox"/>	codedb
<input type="checkbox"/>	ejemplomysql
<input checked="" type="checkbox"/>	modelosdjango
<input type="checkbox"/>	mydb
<input type="checkbox"/>	pythonbase
<input type="checkbox"/>	sakila
<input type="checkbox"/>	sys
<input type="checkbox"/>	world

Refresh

Exp... Schema Objects

Dump Structure and Dat v Select Views Select Tables Unselect All

Objects to Export

☐ Dump Stored Procedures and Functions ☐ Dump Events ☐ Dump Triggers

Export Options

☐ Export to Dump Project Folder C:\Users\Sebastián\Documents\dumps\Dump20210921 ...

Each table will be exported into a separate file. This allows a selective restore, but may be slower.

☒ Export to Self-Contained File C:\Users\Sebastián\Documents\dumps\ExportModelosDjango.sql ...

All selected database objects will be exported into a single, self-contained file.

☒ Create Dump in a Single Transaction (self-contained file only) ☐ Include Create Schema

Press [Start Export] to start... Start Export

Tercer paso.

- Finalizada la exportación, debería mostrarse un mensaje satisfactorio.

Query 1 Administration - Data Export x

Local instance MySQL80

Data Export

Advanced Options...

Object Selection Export Progress

Export Completed

Status:
23 of 23 exported.

Log:


```
14:58:17 Dumping modelosdjango (all tables)
Running: mysqldump.exe --defaults-file="C:\Users\SEBAST~1\AppData\Local\Temp\tmpfhnuays.cnf" --host=localhost --port=3306 --default-character-set=utf8 --user=root --protocol=tcp --single-transaction=TRUE --skip-triggers "modelosdjango"
14:58:18 Export of C:\Users\Sebastián\Documents\dumps\ExportModelosDjango.sql has finished
```

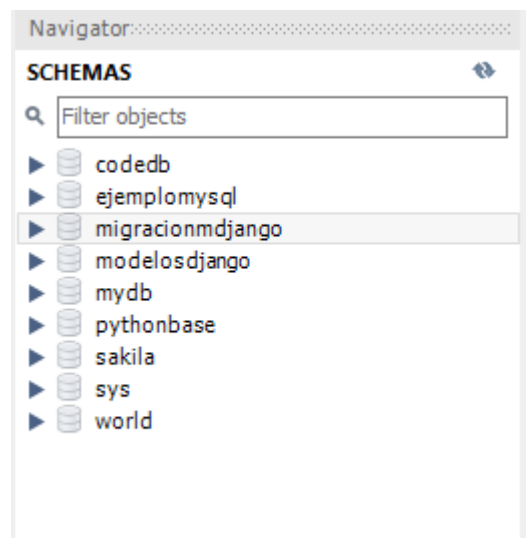
Stop Export Again

Cuarto paso.

Importación.

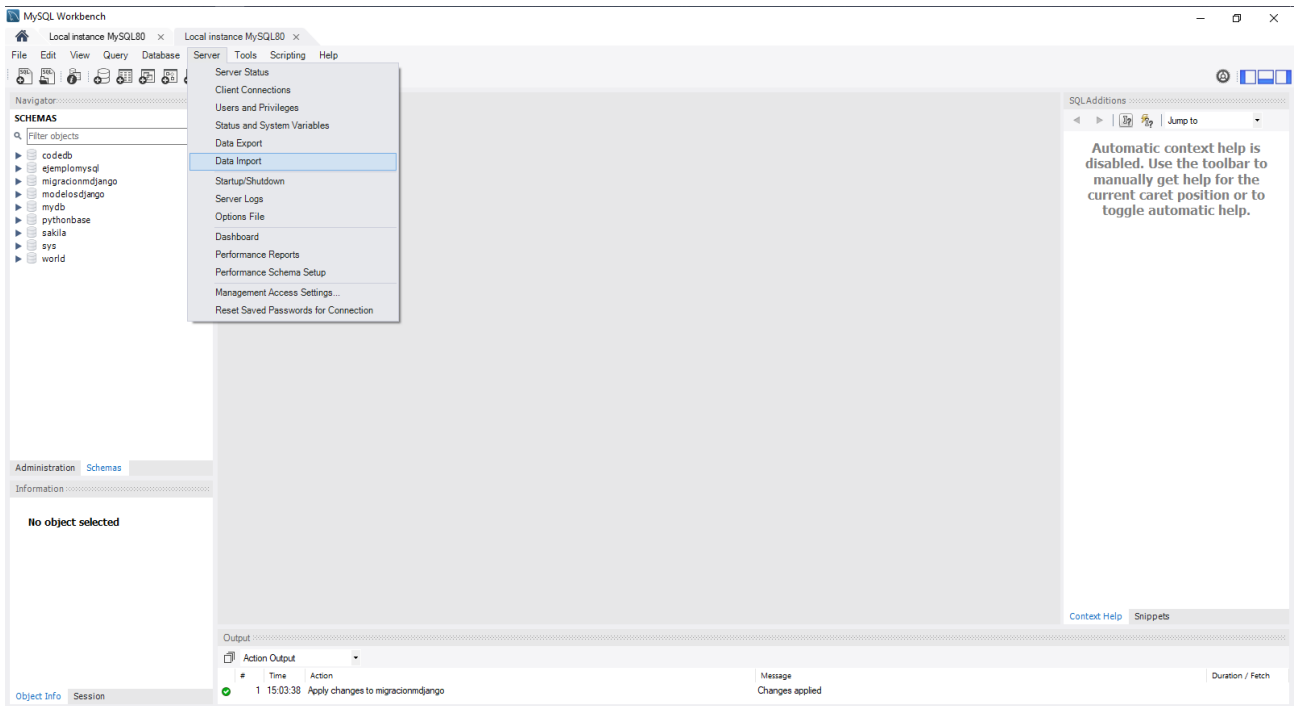
- El primer paso para hacer la exportación de datos de una base de datos a otra en **MySQL Workbench** es iniciar la instancia local.
- Si lo que deseamos es importar los datos de una base de datos a otra, debemos crear una base de datos nueva para almacenar todo allí. En este caso, la nueva base de datos se llama **migracionmdjango** y vamos a importar datos desde **modelosdjango**.

 Para crear una nueva base de datos, debemos simplemente dar 'clic' derecho en el apartado de **Schemas** y posteriormente elegir la opción que dice **Create Schema...** Le asignamos un nombre y así quedaría lista nuestra nueva base de datos.



Segundo paso.

- Después de eso, debemos ubicarnos en el menú superior y seleccionar la opción **Server > Data Import**.

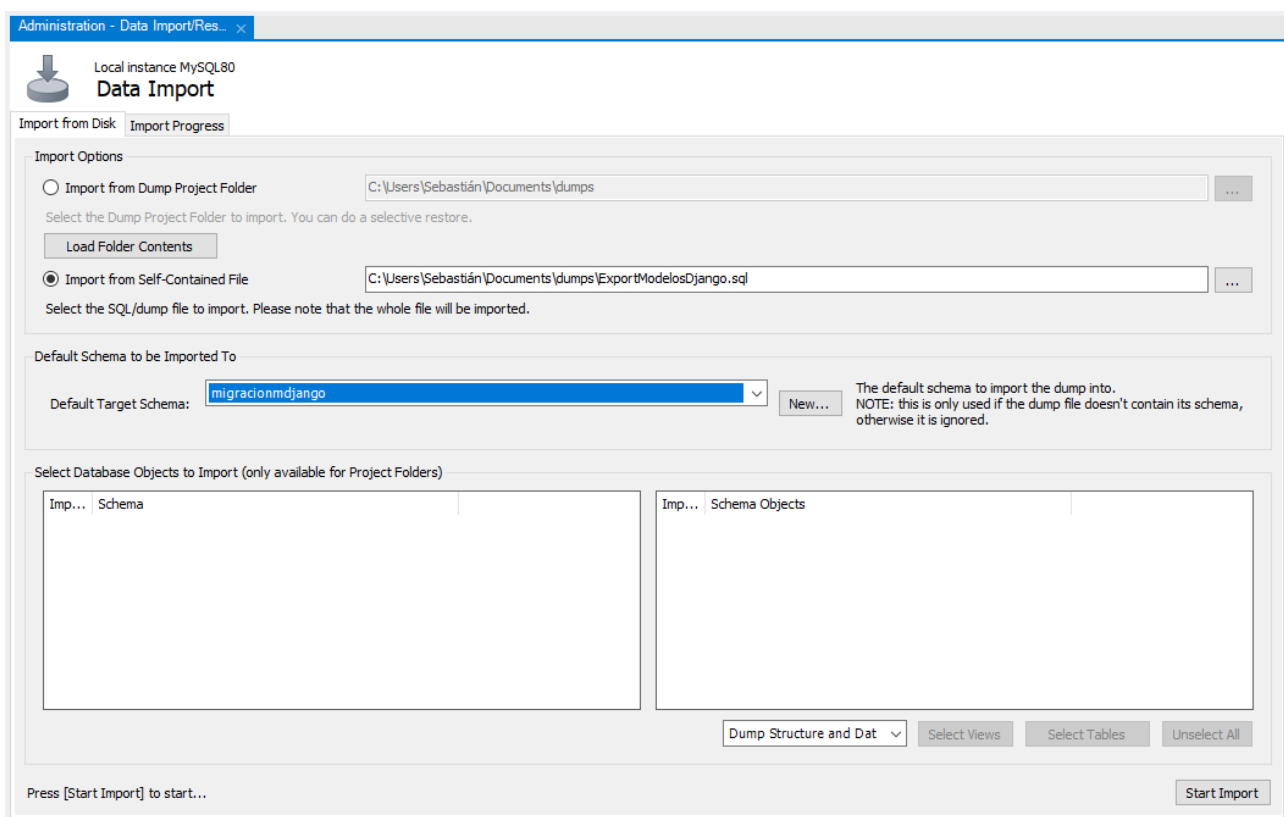


Tercer paso.

- El siguiente paso es seleccionar la base de datos de la que queremos importar los datos y la estructura. Debemos asegurarnos de tener las siguientes opciones marcadas, también debemos ubicar el archivo del que queremos importar la información al archivo al que vamos a importar la información, y dar 'clic' en ***Start Import***.

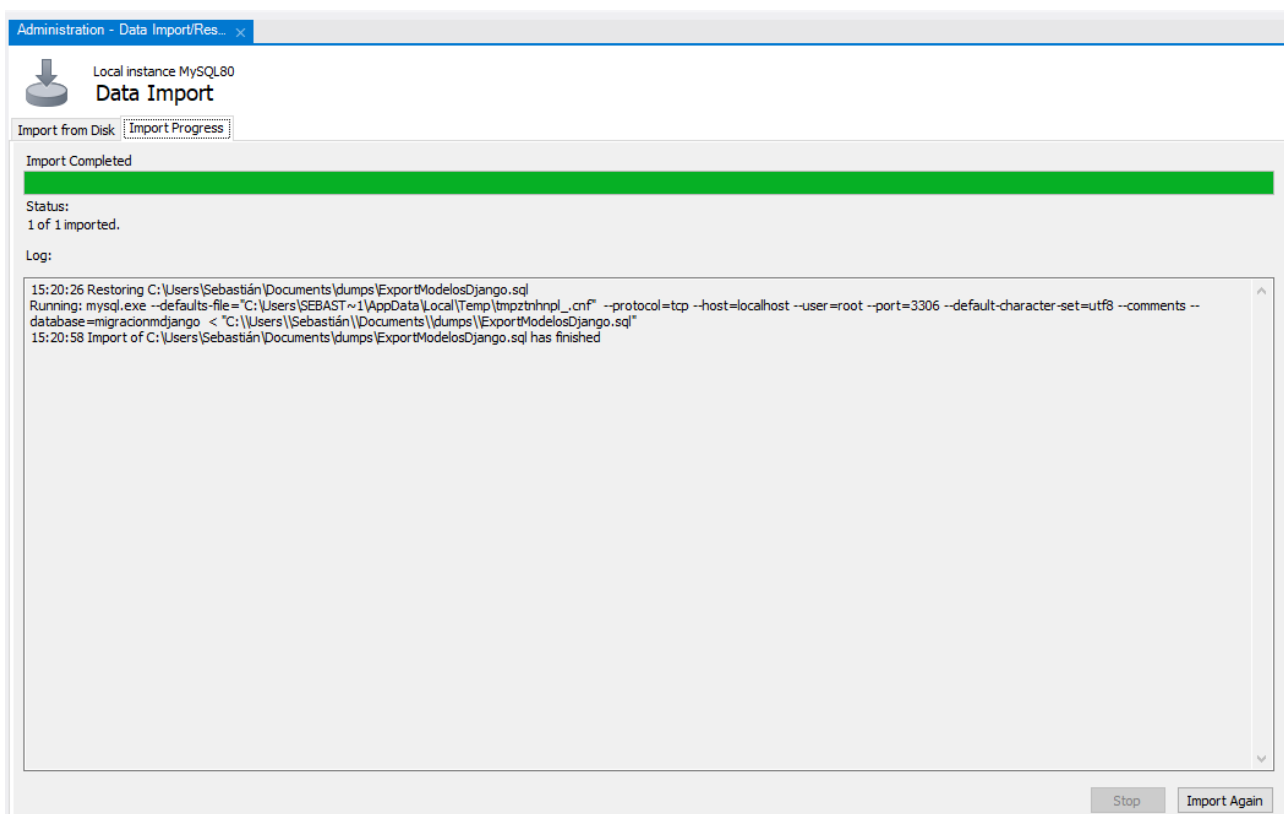


Es aconsejable tomar como ejemplo la siguiente captura.



Cuarto paso.

- Finalizada la importación, debería mostrarse un mensaje satisfactorio.

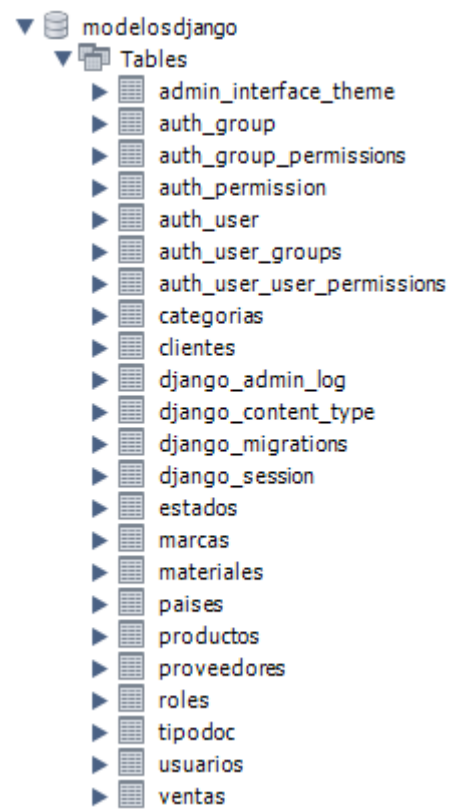
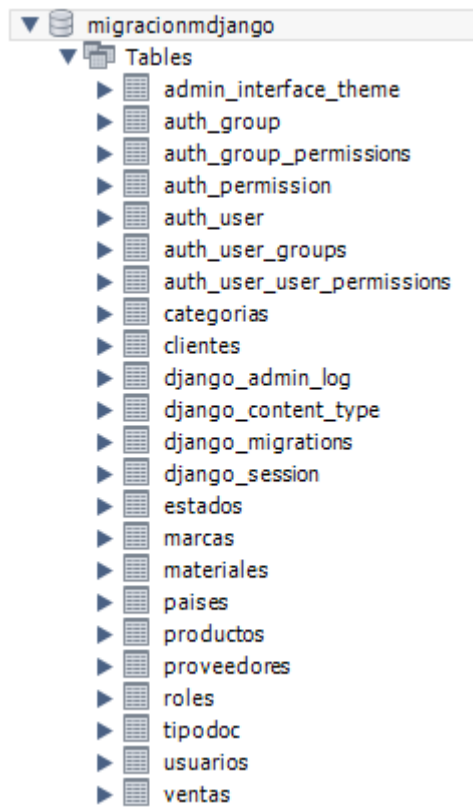


Quinto paso.

- Para comprobar que la importación se logró con éxito, debemos comparar las dos bases de datos.

Base de datos nueva.

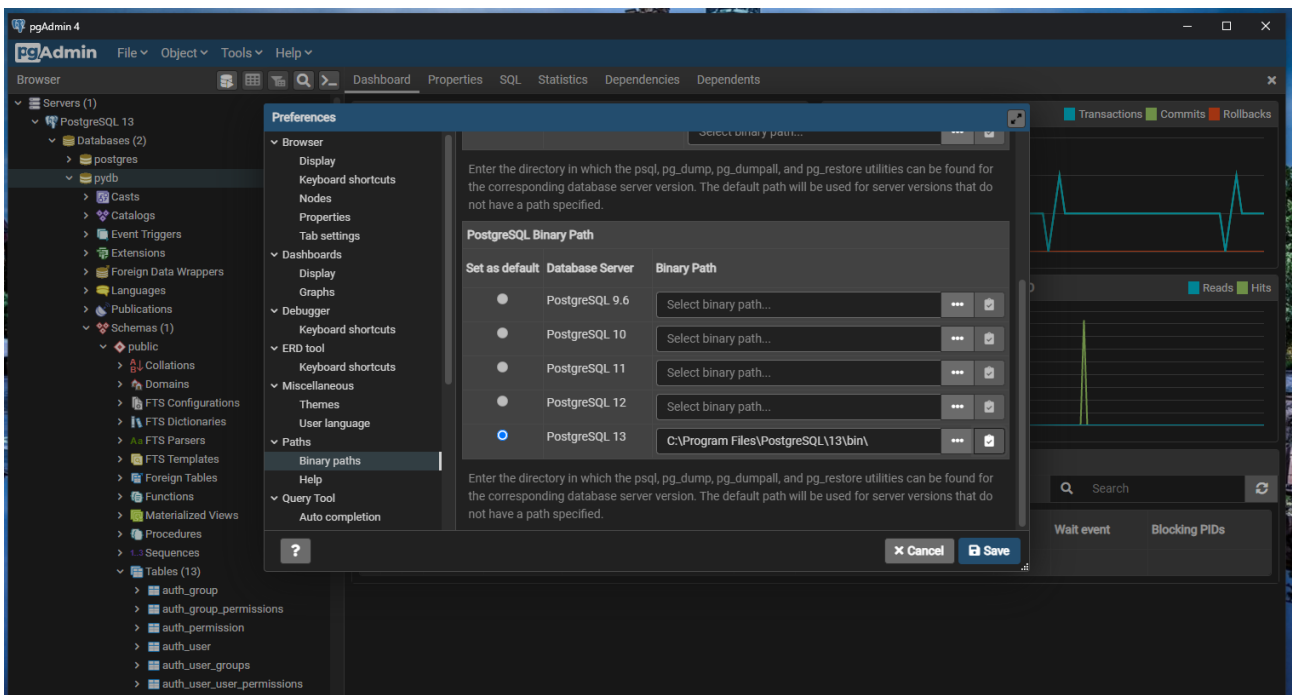
Base de datos original.



Exportación y migración en PostgreSQL.

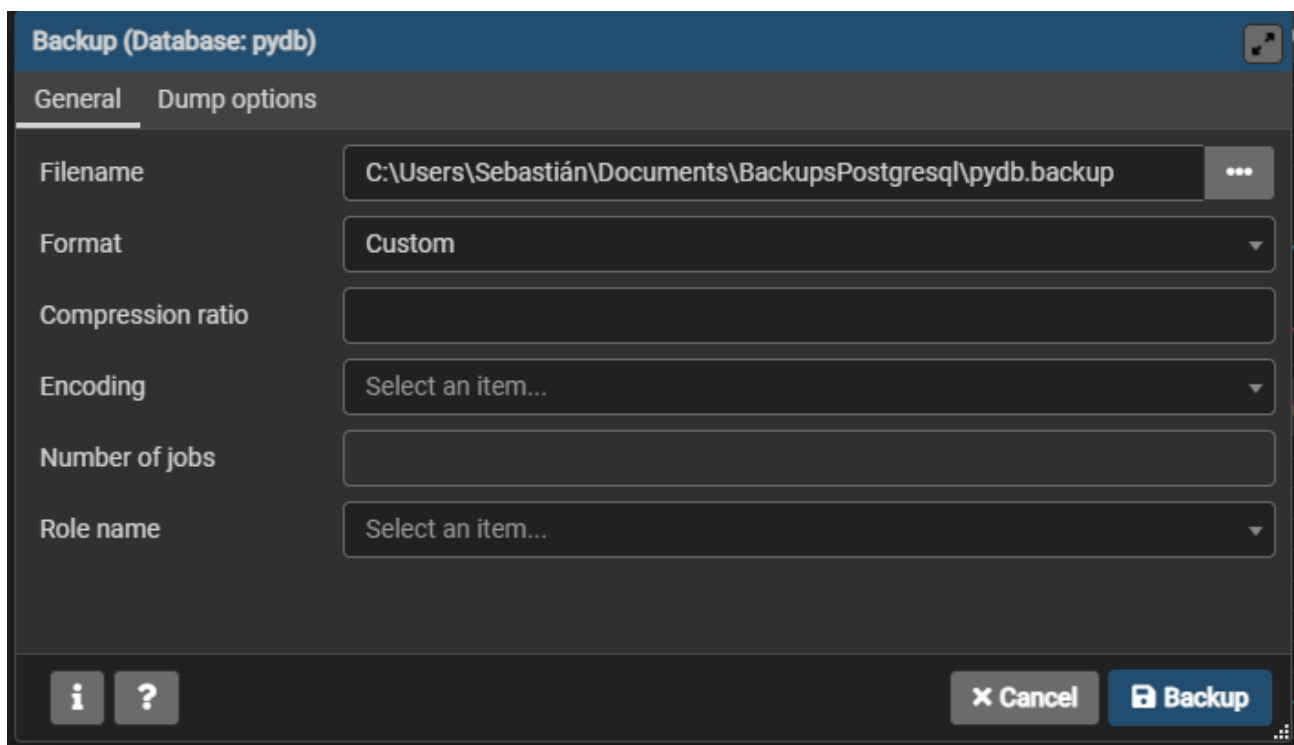
Exportación.

- El primer paso para hacer la exportación de nuestra base de datos en **PostgreSQL** es configurar el PATH —si es necesario—. Para eso debemos ejecutar **pgAdmin 4 > File > Preferences > Paths > Binary Paths** y proceder a agregar la ubicación de nuestro PostgreSQL.



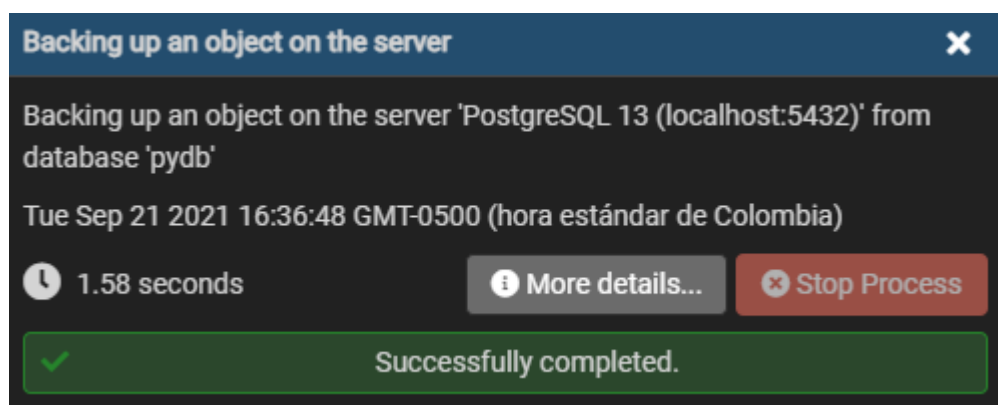
Primer paso.

- Después de esto, debemos dar 'clic' derecho sobre la base de datos de la que vamos a hacer la exportación y seleccionamos la opción que dice **Backup**. Deberá mostrarse la siguiente interfaz. Debemos seleccionar la ubicación donde deseamos guardar nuestro back-up y dar 'clic' en **Backup**.



Segundo paso.


- Deberá salirnos un mensaje de confirmación como el siguiente y con eso confirmaríamos que el back-up de nuestra base de datos se realizó correctamente.



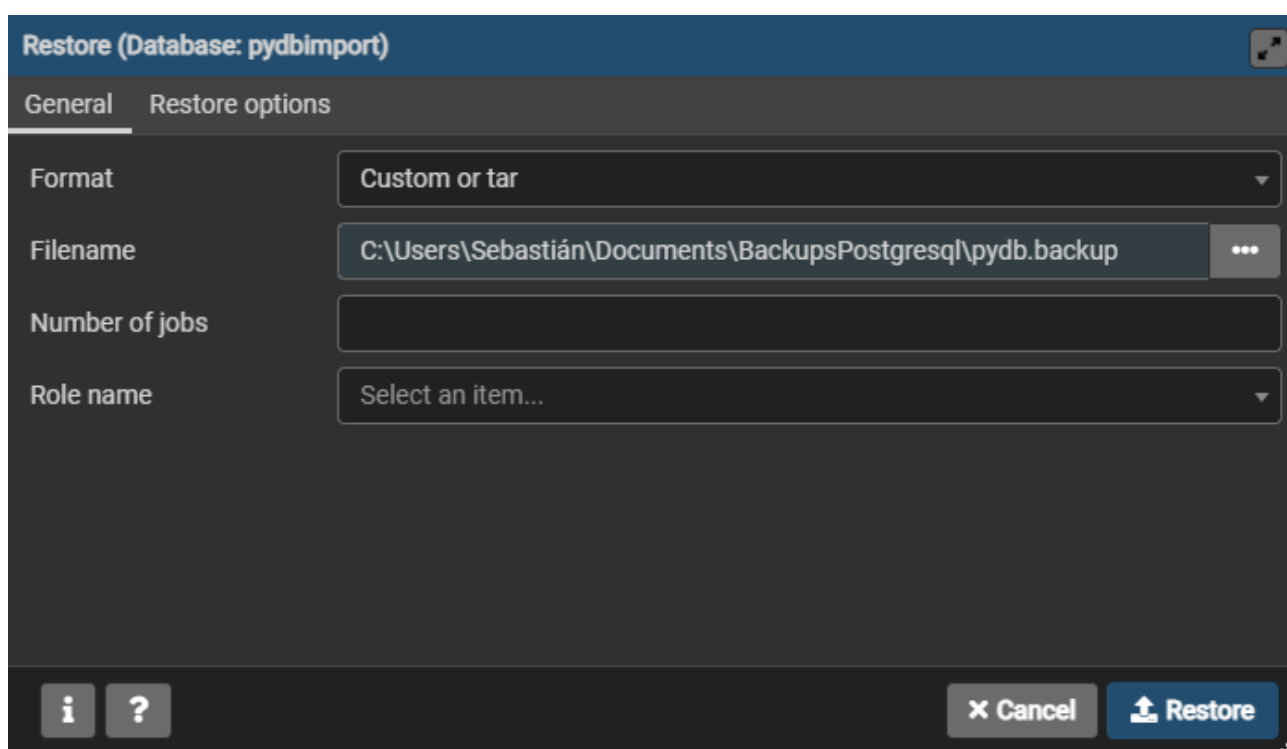
Tercer paso.

Importación.

- Para realizar la importación de datos de una base de datos a otra, debemos crear una nueva base de datos.

-  Si desea crear una base de datos nueva, sólo debe ubicarse en el panel izquierdo del SGBD, dar 'clic' derecho en **Databases > Create Database**. Debe asignarle un nombre, y con eso estaría creada la base de datos.

- El siguiente paso es dar 'clic' derecho en la base de datos que acabamos de crear y seleccionar **Restore**.
- En **Filename** debemos seleccionar el backup que hicimos en el proceso anterior. Damos 'clic' en **Restore**.



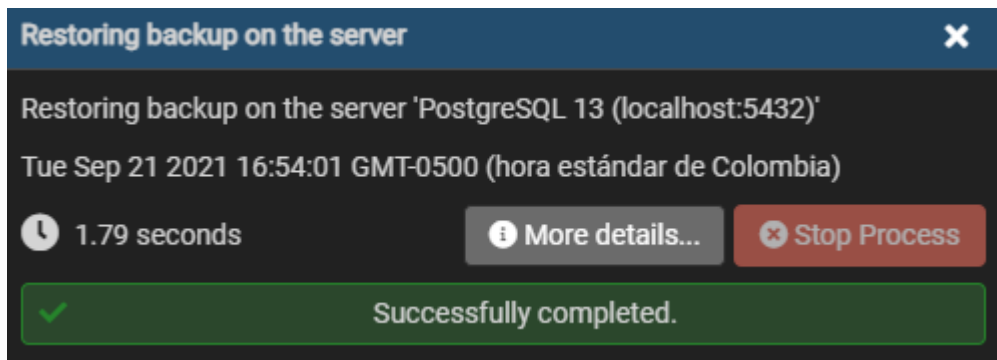
The screenshot shows a 'Restore (Database: pydbimport)' dialog box with two tabs: 'General' and 'Restore options'. The 'General' tab is active. It contains the following fields:

- Format:** A dropdown menu showing 'Custom or tar'.
- Filename:** A text field showing 'C:\Users\Sebastián\Documents\BackupsPostgresql\pydb.backup' with a file selection icon (three dots) to its right.
- Number of jobs:** An empty text field.
- Role name:** A dropdown menu showing 'Select an item...'.

At the bottom of the dialog, there are four buttons: an information icon (i), a help icon (?), a 'Cancel' button with a red X, and a 'Restore' button with an upward arrow icon.

Cuarto paso.

- Deberá salirnos un mensaje de confirmación como el siguiente y con eso confirmaríamos que la importación de nuestra base de datos se realizó correctamente.

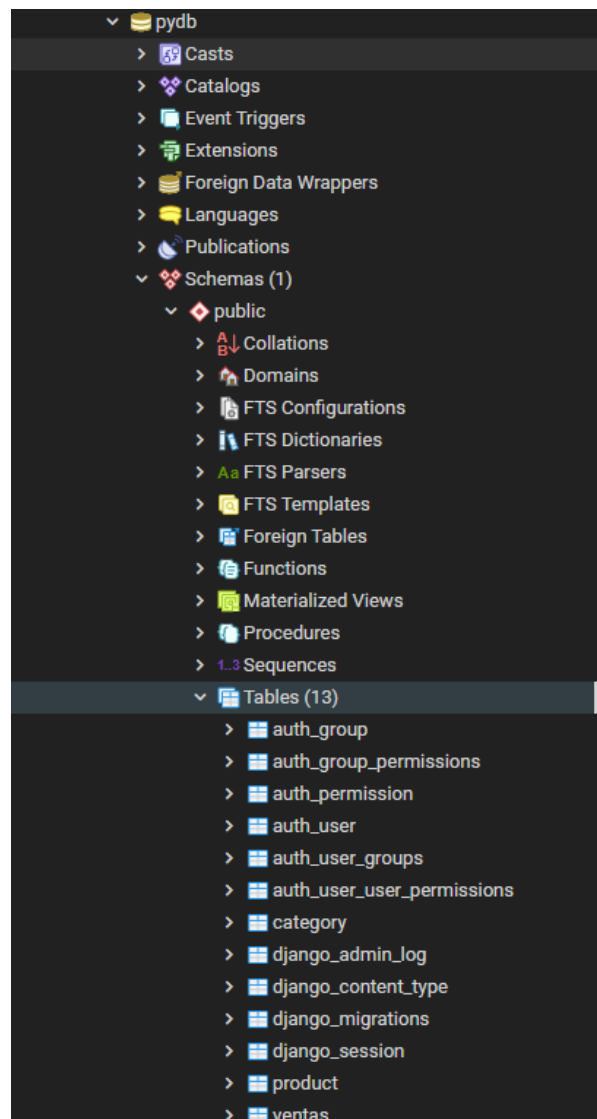


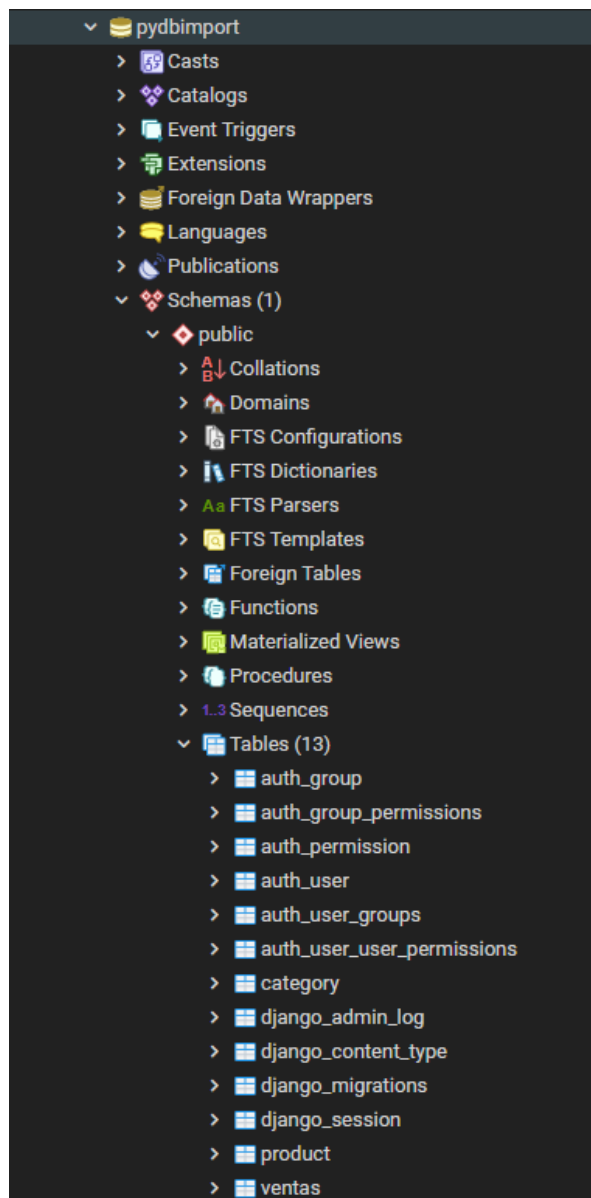
Quinto paso.

- Para comprobar que la importación se logró con éxito, debemos comparar las dos bases de datos.

Base de datos nueva.

Base de datos original.

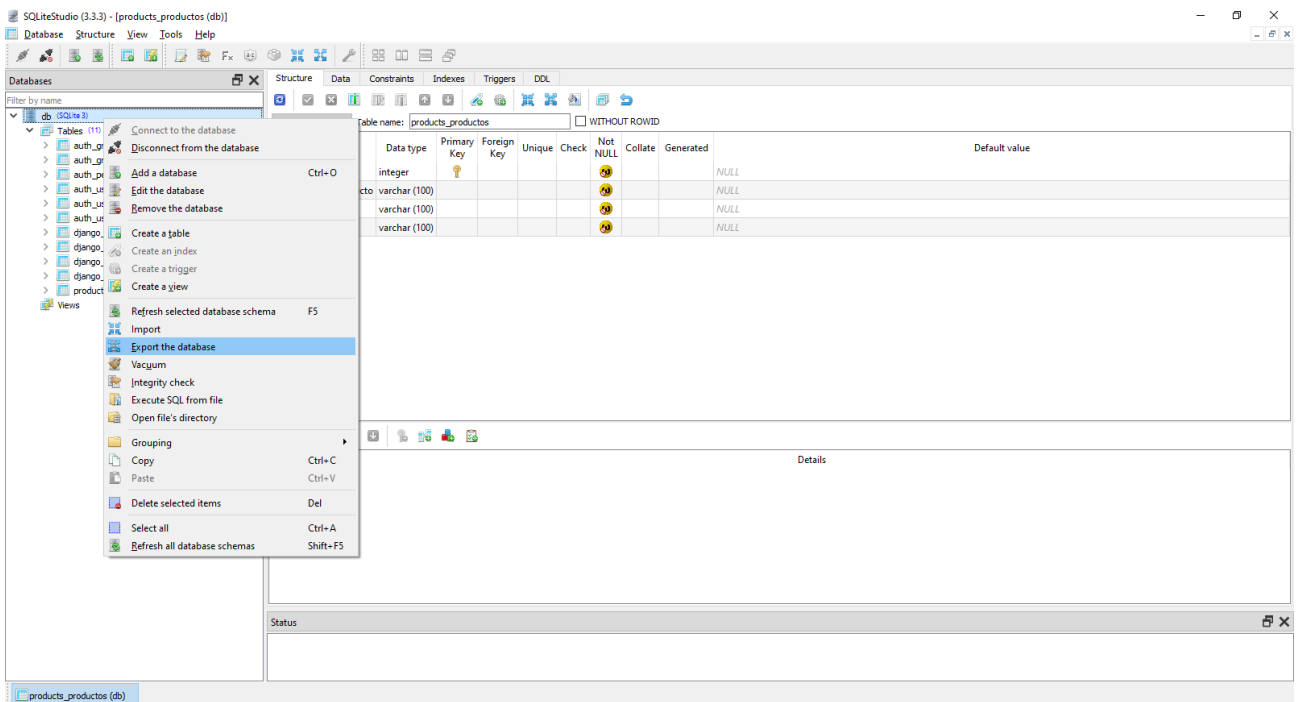




Exportación y migración en SQLite.

Exportación.

- El primer paso para hacer la exportación de nuestra base de datos en **SQLite** es dar 'clic' derecho sobre la base de datos > **Export the database**.



Primer paso.


- Después de ello, saldrá una interfaz como la siguiente.



Debe seleccionar todas las casillas para que la base de datos se exporte completa.

?


×


←  Export


Select database objects to export


Database:


db


▼ ☒  Tables


☒  auth_group


☒  auth_group_permissions


☒  auth_permission


☒  auth_user


☒  auth_user_groups


☒  auth_user_user_permissions

☒  django_admin_log

☒  django_content_type

☒  django_migrations

☒  django_session

☒  products_productos

Select all

Deselect all

☒ Export data from tables

Cancel

Next

Segundo paso.

- Continuando con el proceso, debemos seleccionar la ubicación donde se almacenará nuestro backup y dar 'clic' en **Finish**.

Export

Export format and options

Export format: SQL

Output:

☒ File

☐ Clipboard

Exported text encoding: System

Export format options:

☐ Use SQL formatter to format exported SQL statements


☐ Format DDL statements only (excludes "INSERT" statements)

☐ Generate "DROP IF EXISTS" statement before "CREATE" statement

Cancel Finish

Tercer paso.

- Con este paso finalizaría la exportación de datos desde el SGBD SQLite. Debemos comprobar que la base de datos se haya exportado satisfactoriamente.

 [17:50:39] Export to the file 'C:/Users/Sebastián/Documents/BackupsSQLite/dbexport.sql' was successful.

Cuarto paso.

Importación.

- Para realizar la importación de datos de **SQLite**, debemos utilizar la siguiente herramienta: **DMToolkit**.
- Para descargarla debemos ingresar al siguiente link, elegir la versión que sea compatible con nuestro sistema operativo, e instalarla.

(https://www.dbsofts.com/es/articles/mysql_a_postgresql/).

[Inicio](#)[Descarga](#)[Compra](#)[Acceso del cliente](#)[FAQ](#)[Soporte](#)[Documentos](#)[Quienes somos](#)

Migrando datos de MySQL a PostgreSQL

Kit de herramientas de migración de bases de datos es un conjunto de herramientas potente y de alto rendimiento que puede migrar datos a través de varios formatos de bases de datos, como MySQL, PostgreSQL y así.

Este artículo describe cómo migrar rápidamente datos desde MySQL a PostgreSQL por [Kit de herramientas de migración de bases de datos](#)! ¡Ayude a completar tareas complejas de migración de bases de datos y ahorre mucho tiempo!

Software requerido:

[DMToolkit_x64.zip](#)

(60.7 MiB)

64-bit Aplicación de Windows for Kit de herramientas de migración de bases de datos 10.2.27 (2021-09-03).

(md5: b8004cedec21165b07e42d0eeffbed81)

[DMToolkit_win32.zip](#)

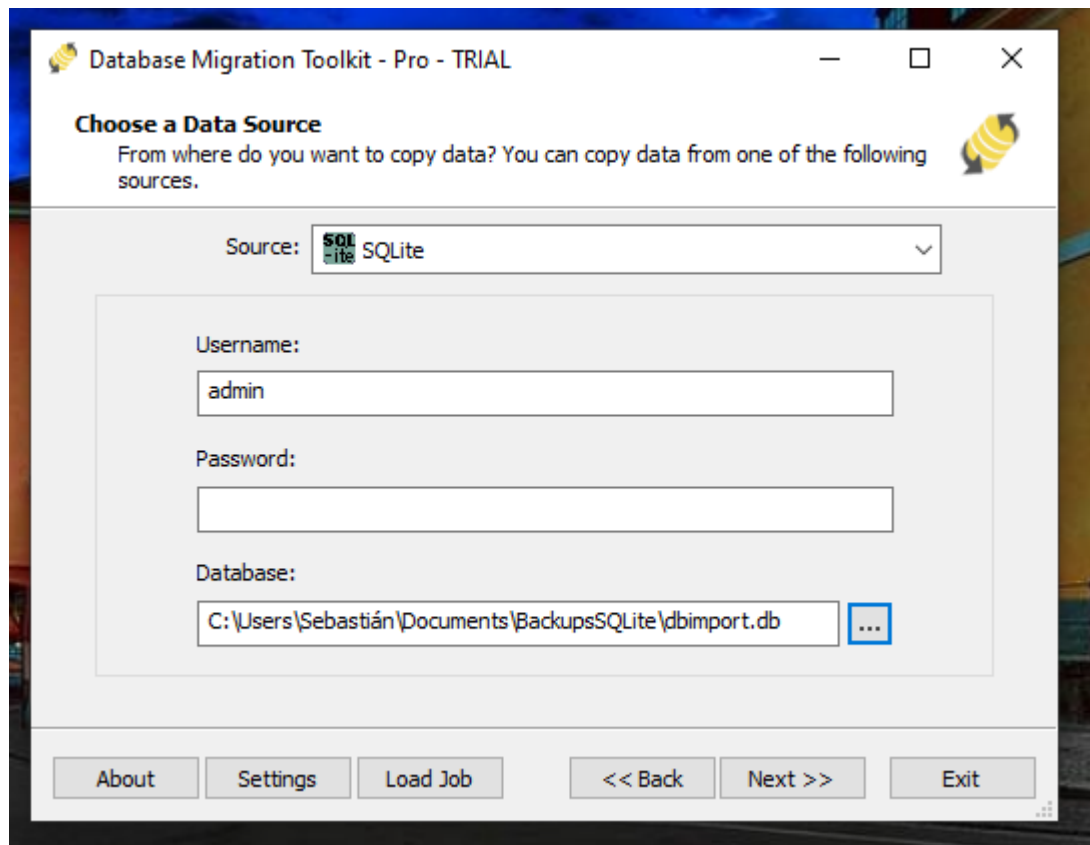
(57.9 MiB)

32-bit Aplicación de Windows for Kit de herramientas de migración de bases de datos 10.2.27 (2021-09-03).

(md5: e0068da9f2626e447e947ecd17a3c439)

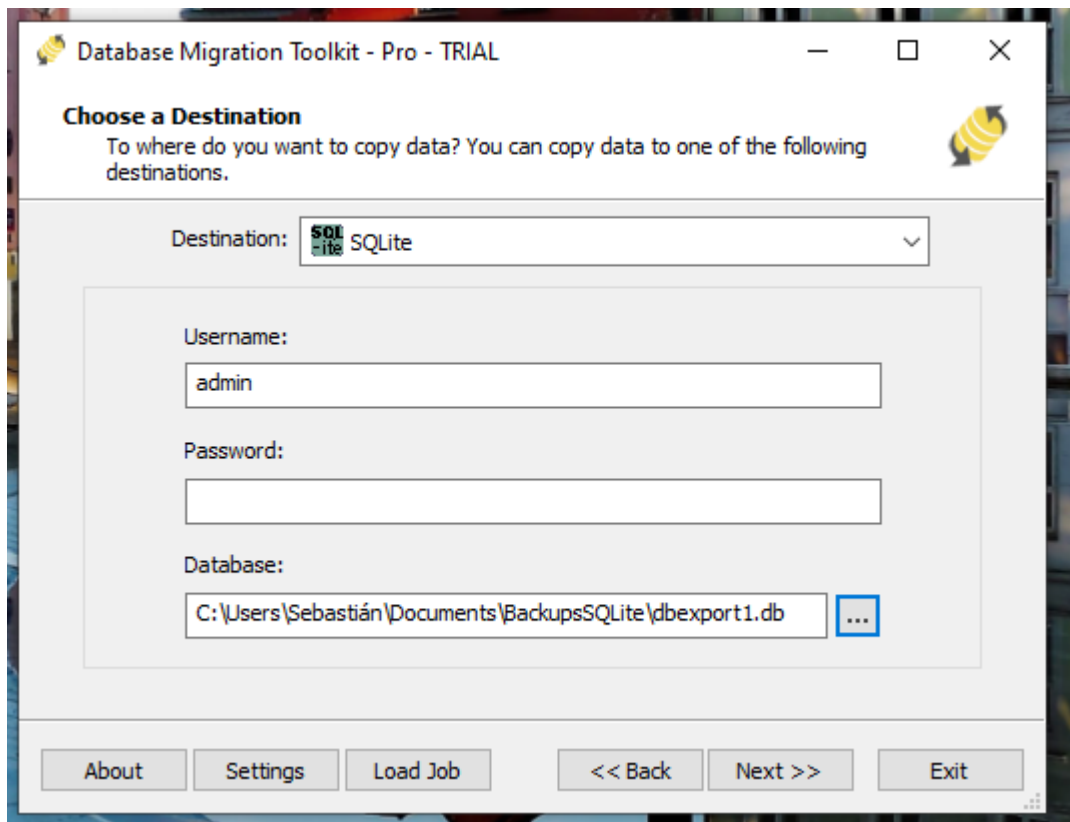
Primer paso.

- El proceso de instalación es muy sencillo. Sólo debemos dar 'clic' en **Siguiente** a todo, elegir la ubicación donde se almacenará la aplicación, y con eso finalizaría el proceso de instalación.
- Teniendo **DMToolkit** instalado, debemos ejecutar el programa y seleccionar de dónde queremos importar nuestros datos. Seleccionamos, en este caso, **SQLite**, agregamos la información adicional y configuramos la base de datos de donde traeremos la información.



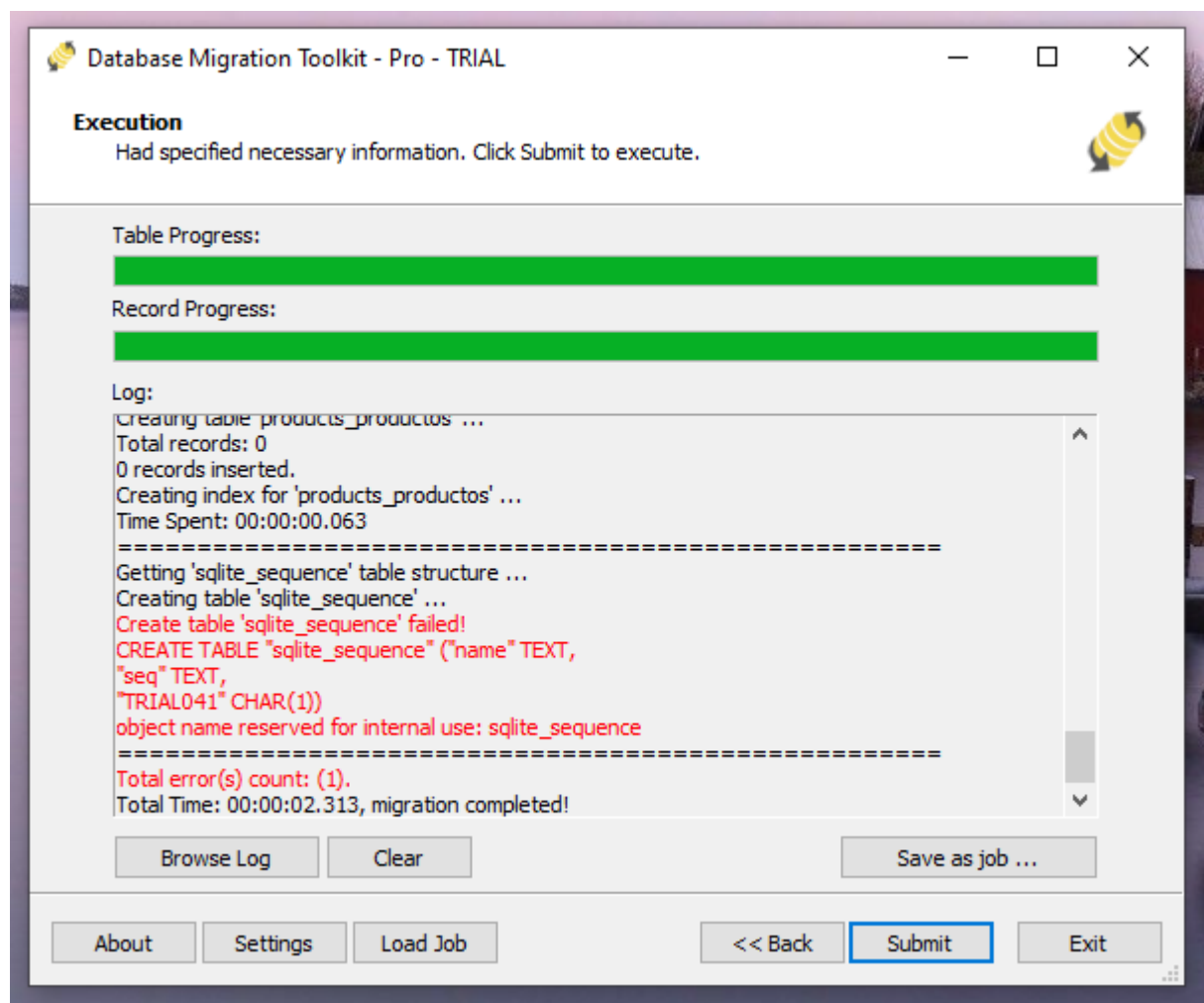
Segundo paso.

- Después, el programa nos pedirá que ingresemos la base de datos a la que queremos importar los datos. Seleccionamos nuestra base de datos, y damos 'clic' en **Next**.



Tercer paso.

- Bien, ahora que agregamos esa información, la herramienta nos solicitará seleccionar las tablas que queremos importar. Seguiremos con la migración dando 'clic' en **Next**. Y después **Submit**.

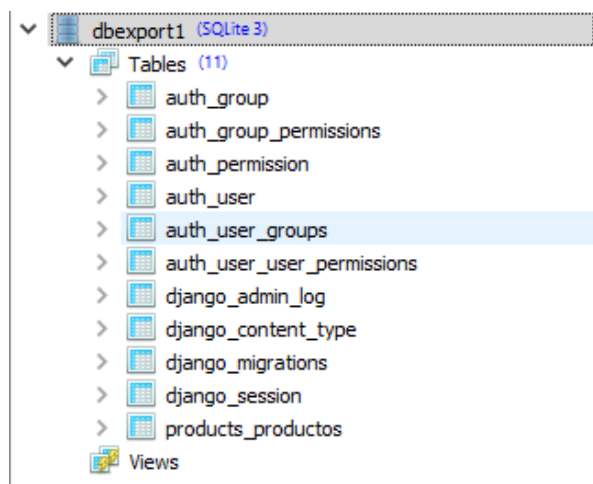
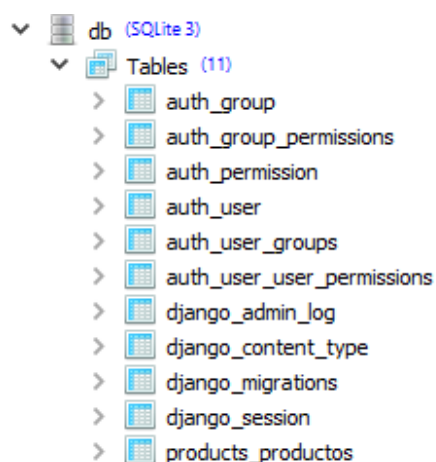


Quinto paso.

- El último paso es comprobar que la importación de datos se hizo correctamente.

Base de datos original.

Base de datos nueva.



- Con esto habrá acabado el proceso de importación de datos en **SQLite**.

Cronograma, actividades, tiempos de ejecución y responsables.

Plan de migración de datos.

Objetivo.

El proceso de migración de datos es una de las cosas más indispensables a la hora de desarrollar algún software. Nos permite transferir datos de un lugar a otro previendo posibles errores del sistema y facilitando a su vez la portabilidad de los datos.

Es por eso que a continuación verá un cronograma con las actividades, los tiempos de ejecución los responsables que estuvieron involucrados en la migración de datos del sistema de inventarios.

Núm. Actividad	Actividades	Fecha de Inicio	Fecha de Finalización	Responsables	¿Cumple?
1	Planeación	24 de Julio	1 de Agosto	Joan Sebastián Prieto Contreras	Sí
2	Instalación de herramientas	1 de Agosto	8 de Agosto	Joan Sebastián Prieto Contreras	Sí
3	Configuración de software y hardware	8 de Agosto	25 de Agosto	Joan Sebastián Prieto Contreras	Sí
4	Copiado de datos	28 de Agosto	12 de Septiembre	Joan Sebastián Prieto Contreras	Sí
5	Pausa en la producción	—	—	—	—

6	Gestión técnica de proveedores	–	–	–	–
7	Verificación técnica de resultados	–	–	–	–
8	Documentación del proceso	18 de Septiembre	20 de Septiembre	Joan Sebastián Prieto Contreras	Sí

Plan de instalación de datos.

Objetivo.

A continuación verá un cronograma con las actividades, los tiempos de ejecución los responsables que estuvieron involucrados en la instalación de datos del sistema de inventarios.

Núm. Actividad	Actividades	Fecha de Inicio	Fecha de Finalización	Responsables	¿Cumple?
1	Planeación	24 de Julio	1 de Agosto	Joan Sebastián Prieto Contreras	Sí
2	Instalación de herramientas	5 de Agosto	10 de Agosto	Joan Sebastián Prieto Contreras	Sí
3	Configuración de software y hardware	12 de Agosto	23 de Agosto	Joan Sebastián Prieto Contreras	Sí

4	Copiado de datos	28 de Agosto	12 de Septiembre	Joan Sebastián Prieto Contreras	Sí
5	Pausa en la producción	–	–	–	–
6	Gestión técnica de proveedores	–	–	–	–
7	Verificación técnica de resultados	–	–	–	–
8	Documentación del proceso	18 de Septiembre	20 de Septiembre	Joan Sebastián Prieto Contreras	Sí

Plan de respaldo.

Objetivo

Con el plan de respaldo se busca realizar la exportación de la información almacenada en la base de datos. Es necesario generar la migración de la base de datos para tener un respaldo previendo cualquier contratiempo que pueda ocurrir.

Núm. Actividad	Actividades	Fecha de Inicio	Fecha de Finalización	Responsables	¿Cumple?
1	Planeación	24 de Julio	1 de Agosto	Joan Sebastián Prieto Contreras	Sí

2	Instalación de herramientas	5 de Agosto	10 de Agosto	Joan Sebastián Prieto Contreras	Sí
3	Configuración de software y hardware	12 de Agosto	23 de Agosto	Joan Sebastián Prieto Contreras	Sí
4	Copiado de datos	En proceso	En proceso	Joan Sebastián Prieto Contreras	En proceso
5	Pausa en la producción	–	–	–	–
6	Gestión técnica de proveedores	–	–	–	–
7	Verificación técnica de resultados	–	–	–	–
8	Documentación del proceso	18 de Septiembre	20 de Septiembre	Joan Sebastián Prieto Contreras	Sí