

INFORME FINAL DE PROYECTO

UF1470 – Administración de Sistemas Gestores de Bases de Datos (SGBD)

Alumno:

Juan de la Morena Marzalo

Curso:

IFCT0310 - Administración de Bases de Datos

Proyecto:

Implantación y gestión de bases de datos MySQL y SQLite Automatización de backups y auditoría de datos Monitorización del sistema y optimización de consultas

Repositorio GitHub:

🔗 https://github.com/JuanHoob/Trabajo_IFCT0310_1470.git

Resumen: Este documento detalla el proceso completo de creación, gestión, automatización y monitorización de bases de datos relacionales utilizando MySQL y SQLite, siguiendo buenas prácticas de administración de sistemas de bases de datos. Incluye capturas de evidencia, scripts SQL, automatizaciones programadas y optimizaciones de rendimiento.

Índice

Parte 1 – MySQL: Configuración y Administración

- 1. Tipos de bases de datos en DBeaver
- 2. Usuarios DROP IF EXISTS
- 3. Crear base de datos MySQL
- 4. Crear tabla clientes (MySQL)
- 5. Crear índice para optimizar consultas

Parte 2 – SQLite: Creación e Inserción de Datos

- 6. Crear base de datos SQLite (nombre)
- 7. Definir ruta del archivo SQLite
- 8. Crear base de datos SQLite
- 9. Crear tabla clientes SQLite (con DROP IF EXISTS)
- 10. Insertar registros de prueba

Parte 3 – Automatización y Seguridad

- 11. Backup manual de MySQL
- 12. Crear procedimiento almacenado (MySQL)
- 13. Ejecutar procedimiento almacenado - correcto
- 14. Ejecutar procedimiento almacenado - error
- 15. Mostrar procesos en ejecución
- 16. Kill de procesos bloqueados o pesados
- 17. Crear Trigger de auditoría
- 18. Ejecutar UPDATE para probar Trigger
- 19. Visualizar registros en log_clientes

Parte 4 – Planificación de Tareas y Backup

- 20. Script .bat de backup Windows
- 21. Crear tarea programada en Windows
- 22. Nombre y descripción de la tarea
- 23. Asignación de periodicidad diaria
- 24. Seleccionar el script .bat
- 25. Opciones avanzadas: ejecutar con privilegios
- 26. Resultado de la tarea programada
- 27. Backup generado automáticamente

Parte 5 – Monitorización del Rendimiento

- 28. Acceso al Dashboard de DBeaver

- 29. Vista general del Dashboard de Monitorización
- 30. Estado general del servidor (Server Status)
- 31. Monitorizar procesos activos (SHOW PROCESSLIST)
- 32. EXPLAIN de consultas para optimización

8. Anexo: Conversión de Markdown a PDF

Descripción General

Este informe técnico constituye la entrega oficial del módulo **UF1470 – Administración de Sistemas Gestores de Bases de Datos (SGBD)**, correspondiente al certificado de profesionalidad **IFCT0310 – Administración de Bases de Datos**. El trabajo se desarrolla en un entorno simulado que representa una empresa ficticia llamada **DataSolutions S.A.**, especializada en servicios de gestión y análisis de información empresarial.

El objetivo de esta práctica ha sido poner en funcionamiento una infraestructura básica de base de datos relacional, que pueda soportar operaciones habituales de consulta, inserción y mantenimiento de datos, así como implementar medidas de auditoría, seguridad y automatización que forman parte de las responsabilidades habituales de un administrador de bases de datos (DBA).

Durante el desarrollo del informe se han abordado varias tareas técnicas distribuidas en cinco grandes bloques:

Diseño y configuración inicial

Se realizaron operaciones de creación y configuración de bases de datos utilizando **MariaDB MySQL** y **SQLite**, incluyendo la definición de esquemas, creación de tablas, e inserción de registros de prueba. Se utilizó el cliente visual **DBeaver** como interfaz de administración para facilitar la conexión y gestión de ambos sistemas.

Seguridad y gestión de usuarios

Se definieron usuarios con distintos niveles de permisos, incluyendo perfiles con acceso restringido a determinadas operaciones o tablas. Esta separación de roles permite garantizar la integridad de los datos y limitar posibles errores o accesos indebidos por parte de los distintos actores del sistema.

Optimización, procedimientos y auditoría

Se diseñaron e implementaron índices para mejorar el rendimiento de consultas complejas. Asimismo, se desarrolló un **procedimiento almacenado** en MySQL con validación lógica para la inserción de registros, y se configuró un **trigger de auditoría en SQLite** para llevar un control sobre las actualizaciones realizadas en las tablas principales.

Automatización de tareas críticas

Se creó un script de copia de seguridad para exportar los datos de forma periódica y se configuró el **Programador de Tareas de Windows** para su ejecución automática. Esta tarea es fundamental en cualquier sistema productivo para minimizar el riesgo de pérdida de información ante fallos.

Monitorización y supervisión del sistema

Se utilizaron las herramientas de monitorización de **DBeaver** y comandos SQL de diagnóstico para identificar cuellos de botella, procesos bloqueantes y rendimiento de las consultas. Además, se exploraron comandos de gestión de procesos y análisis de planes de ejecución mediante **EXPLAIN**.

Este informe ha sido documentado paso a paso y contiene:

- Código SQL comentado para cada sección.
- Capturas de pantalla que demuestran la correcta ejecución de cada acción.
- Scripts reales listos para ser usados o adaptados.
- Un enfoque pedagógico pensado para que cualquier lector —incluso sin experiencia previa— pueda seguir el flujo de trabajo, entender las decisiones técnicas y replicar el proceso.

El informe ha sido redactado en **formato Markdown**, y exportado a PDF mediante el editor **Typora**, con estructura paginada, índice automático e imágenes embebidas. Todo el código y material utilizado está disponible también en el siguiente repositorio público de GitHub para su consulta técnica:

🔗 https://github.com/JuanHoob/Trabajo_IFCT0310_1470.git

Esta documentación representa, además de una prueba de competencias técnicas, un modelo realista de cómo se debe estructurar un proyecto de administración de bases de datos desde cero, considerando criterios de **seguridad, rendimiento, documentación y automatización**.

Parte 1 - MySQL: Configuración y Administración

1. Creación de la Base de Datos

```
CREATE DATABASE DataSolutionsDB;
```

Explicación: Crea la base de datos vacía donde se alojarán todas las tablas y datos de clientes.

2. Gestión de Usuarios y Permisos

```
-- Usuario consultor (solo lectura)
CREATE USER 'consultor'@'localhost' IDENTIFIED BY
'TuContraseña123';
GRANT SELECT ON DataSolutionsDB.* TO 'consultor'@'localhost';

-- Usuario admin_ventas (lectura, inserción, actualización en tabla
clientes)
CREATE USER 'admin_ventas'@'localhost' IDENTIFIED BY
'TuContraseña123';
GRANT SELECT, INSERT, UPDATE
ON DataSolutionsDB.clientes
TO 'admin_ventas'@'localhost';
```

Explicación: Se crean dos perfiles de acceso para separar responsabilidades y minimizar riesgos.

Nota: he utilizado en el ejemplo de la captura de pantalla el comando DROP if exist ya que un nombre genérico puede que se haya creado de antemano y quieras pisarlo, de esta forma no te daría error y puedes estar seguro de crear tu usuario en el lugar que quieras.

Consideraciones: Se ha de tener en cuenta que durante todo el ejercicio que la nomenclatura cambia según la base de datos que se utilice Mqlite, MariaDB o MySQL.

3. Optimización de Consultas

Consulta original

```
SELECT *
  FROM clientes
 WHERE ciudad = 'Madrid'
   AND fecha_registro > '2024-01-01';
```

Problema detectado

Sin índice, MySQL realiza un *full table scan*, revisando fila a fila, lo cual se vuelve muy lento si la tabla crece.

Solución propuesta

```
CREATE INDEX idx_ciudad_fecha
  ON clientes(ciudad, fecha_registro);
```

Detalle: Un índice compuesto en (*ciudad, fecha_registro*) permite filtrar rápidamente por ambos campos. *Consideraciones:* Si la columna *ciudad* tiene poca selectividad (p. ej. muchos "Madrid"), el beneficio será menor. Agregar índices también impacta ligeramente las inserciones/actualizaciones.

4. Gestión de Procesos Lentos

```
SHOW PROCESSLIST;
-- Identificar ID del proceso pesado:
KILL <ID_DEL_PROCESO>;
```

Explicación: Con **SHOW PROCESSLIST** vemos las consultas activas y podemos abortar las que bloquean recursos.

Parte 2 - SQLite: Creación e Inserción de Datos

1. Creación de la Tabla

```
CREATE TABLE clientes (
    id INTEGER PRIMARY KEY,
    nombre TEXT,
    apellido TEXT,
    ciudad TEXT,
    fecha_registro DATE
);
```

2. Inserción de Registros de Ejemplo

```
INSERT INTO clientes (nombre, apellido, ciudad, fecha_registro)
VALUES
    ('Juan', 'Pérez', 'Madrid', '2024-02-15'),
    ('Ana', 'López', 'Barcelona', '2023-11-03'),
    ('Luis', 'Martínez', 'Sevilla', '2024-01-21'),
    ('Clara', 'Ramírez', 'Valencia', '2022-09-12'),
    ('Mario', 'Gómez', 'Madrid', '2024-03-01');
```

Nota: SQLite no gestiona usuarios; la seguridad se delega al sistema de archivos.
Además se ha de tener en cuenta en el ejercicio que hay que cambiar de
nomenclatura para la creación de tabla según la base de datos que se esté usando.

Parte 3 - Automatización y Seguridad

1. Copia de Seguridad (MySQL)

```
CREATE TABLE clientes_backup AS
SELECT * FROM clientes;
```

Explicación: Clona los datos en una tabla de respaldo instantánea.

2. Procedimiento Almacenado de Validación

```
DELIMITER //

CREATE PROCEDURE insertar_cliente (
    IN p_nombre VARCHAR(50),
    IN p_apellido VARCHAR(50),
    IN p_ciudad VARCHAR(50),
    IN p_fecha DATE
)
BEGIN
    IF p_fecha > CURDATE() THEN
        SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'La fecha no puede ser futura.';
    ELSE
        INSERT INTO clientes (nombre, apellido, ciudad, fecha_registro)
            VALUES (p_nombre, p_apellido, p_ciudad, p_fecha);
    END IF;
END //

DELIMITER ;
```

Desafíos: Manejar el cambio de delimitador correctamente y asegurar privilegios para **SIGNAL**.

3. Trigger de Auditoría (SQLite)

```
CREATE TABLE log_clientes (
    fecha_modificacion TEXT,
    usuario TEXT,
    operacion TEXT,
    datos_antiguos TEXT,
    datos_nuevos TEXT
);

CREATE TRIGGER trigger_log_update
AFTER UPDATE ON clientes
BEGIN
    INSERT INTO log_clientes (
        fecha_modificacion, usuario, operacion,
        datos_antiguos, datos_nuevos
    ) VALUES (
        datetime('now'),
        'usuario_windows',
        'UPDATE',
        json_object(
            'id', OLD.id,
            'nombre', OLD.nombre,
            'apellido', OLD.apellido,
            'ciudad', OLD.ciudad,
            'fecha_registro', OLD.fecha_registro
        ),
        json_object(
            'id', NEW.id,
            'nombre', NEW.nombre,
            'apellido', NEW.apellido,
            'ciudad', NEW.ciudad,
            'fecha_registro', NEW.fecha_registro
        )
    );
END;
```

Detalle: SQLite 3.38+ requerido para funciones JSON; los triggers no admiten múltiples sentencias sin envolver.

Parte 4 - Planificación de Tareas

Script de Backup Automático (Windows)

Guárdalo como `tareas_programadas/backup_windows.bat`:

```
@echo off  
set FECHA=%DATE:~6,4%-%DATE:~3,2%-%DATE:~0,2%  
mysqldump -u root -pTuContraseña DataSolutionsDB ^  
> "C:\backups\DataSolutionsDB_%FECHA%.sql"
```

Automatización de Backup - Programador de Tareas de Windows

Una vez creado el script de copia de seguridad `backup_windows.bat`, se procedió a su automatización mediante el Programador de tareas de Windows.

1. Apertura del Programador de tareas

- Se accede a **Programador de tareas** desde el menú Inicio de Windows.

2. Creación de una nueva tarea básica

- Se selecciona la opción **Crear tarea básica**.
- Se asigna:
 - **Nombre:** Backup automático base de datos
 - **Descripción:** Realiza automáticamente una copia de seguridad de la base de datos MySQL cada día.

3. Configuración de la frecuencia

- Se elige ejecutar la tarea **Diariamente**.

4. Programación de la hora de ejecución

- Se establece una **hora fija** de ejecución diaria (por ejemplo, 20:00 h).

5. Selección de la acción

- Se configura la tarea para **Iniciar un programa**.
- Se selecciona el archivo `backup_windows.bat` previamente creado.

6. Revisión y finalización

- Se revisa el resumen de la tarea.

- Se pulsa en **Finalizar** para dejar programada la ejecución automática.
-

Observaciones Importantes

- El archivo `backup_windows.bat` debe tener permisos de ejecución adecuados.
 - La contraseña de MySQL debe estar escrita en el script si se desea ejecución sin intervención manual.
 - La tarea se ejecutará siempre que el equipo esté encendido a la hora programada.
 - La copia de seguridad se almacenará en la ruta especificada en el `.bat` (`C:\backups\`).
-
-

Parte 5 - Monitoreo de Rendimiento

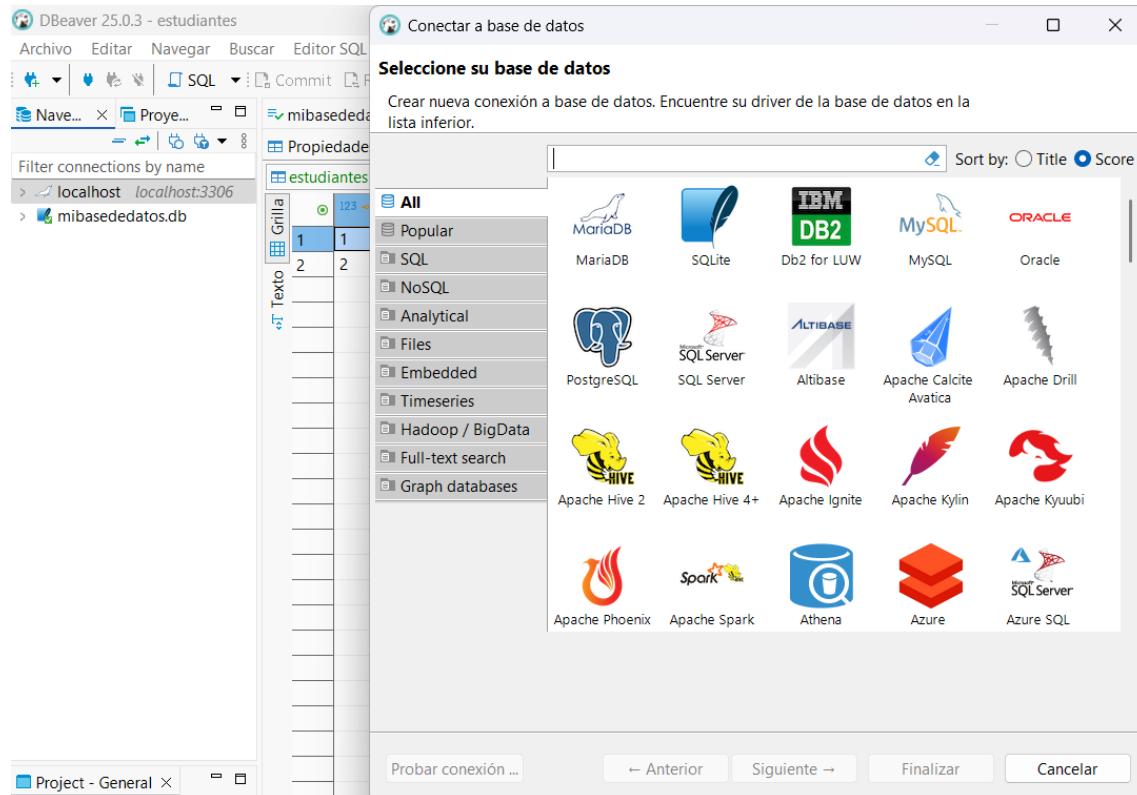
- **MySQL Workbench:** Performance Dashboard y Query Analyzer
- **DBeaver:** Monitorización en vivo
- Comandos clave:

```
SHOW STATUS LIKE '%slow_queries%';
SHOW PROCESSLIST;
EXPLAIN ANALYZE SELECT ...;
```

Capturas de Pantalla Organizadas

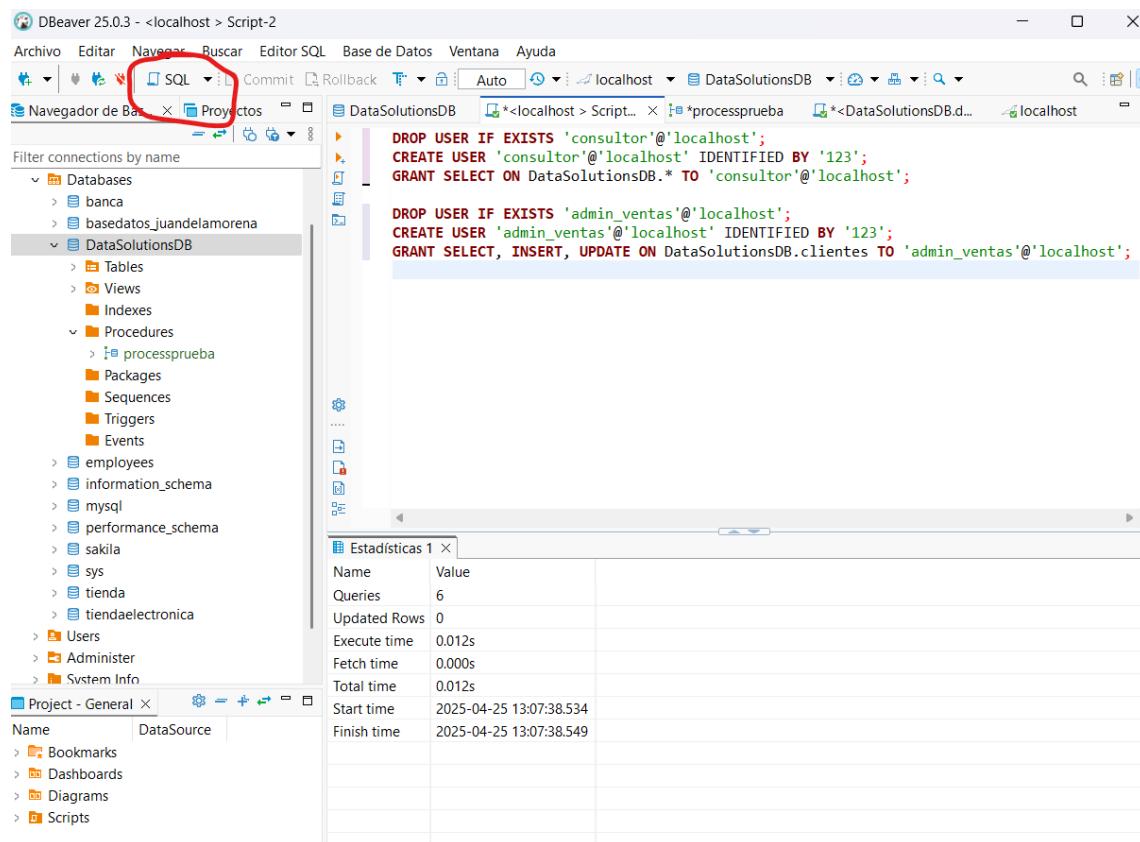
Parte 1 – MySQL: Configuración y Administración

1. Vista general de tipos de bases de datos en DBeaver



[Ver imagen a tamaño real](#)

2. Usuarios DROP IF EXISTS



DBeaver 25.0.3 - <localhost> Script-2

Archivo Editar Navegar Buscar Editor SQL Base de Datos Ventana Ayuda

Navegador de Bases de Datos Proyectos

Filter connections by name

Databases

- banca
- basedatos_juandelamorena
- DataSolutionsDB
 - Tables
 - Views
 - Indexes
 - Procedures
 - processprueba
 - Packages
 - Sequences
 - Triggers
 - Events
- employees
- information_schema
- mysql
- performance_schema
- sakila
- sys
- tienda
- tiendaelectronica

Users

Administer

System Info

Project - General

Nombre Fuente de datos

Bookmarks

Dashboards

Diagrams

Scripts

SQL Commit Rollback Auto localhost DataSolutionsDB localhost localhost localhost

Drop user if exists 'consultor'@'localhost';
Create user 'consultor'@'localhost' identified by '123';
Grant select on DataSolutionsDB.* to 'consultor'@'localhost';

Drop user if exists 'admin_ventas'@'localhost';
Create user 'admin_ventas'@'localhost' identified by '123';
Grant select, insert, update on DataSolutionsDB.clientes to 'admin_ventas'@'localhost';

Estadísticas 1

Name	Value
Queries	6
Updated Rows	0
Execute time	0.012s
Fetch time	0.000s
Total time	0.012s
Start time	2025-04-25 13:07:38.534
Finish time	2025-04-25 13:07:38.549

[Ver imagen a tamaño real](#)

3. Crear base de datos MySQL

The screenshot shows the DBVisualizer 2.5.0.3 interface. In the top navigation bar, the 'Base de Datos' tab is selected. On the left, the 'Navegador de Bases de Datos' (localhost:3306) is expanded, showing the 'Databases' section. A context menu is open over the 'Ver Databases' item, listing options like 'Crear Nuevo Database', 'Copiar', 'Pegar', and 'Renovar'. To the right, the 'estudiantes' table is displayed in a grid view with columns 'id', 'nombre', and 'apellidos'. The table contains two rows: one for Ana García and one for Juan Pérez. At the bottom, a message indicates '2 row(s) fetched - 0.001s (0.001s fetch)'.

[Ver imagen a tamaño real](#)

4. Crear tabla clientes (MySQL)

The screenshot shows the DBeaver interface with the following details:

- Toolbar:** Archivo, Editar, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, Ayuda.
- Connection:** Auto, localhost, DataSolutionsDB.
- Left Panel (Project - General):** Shows Databases, Tables, Views, Indexes, Procedures, Packages, Sequences, Triggers, Events, employees, information_schema, mysql, performance_schema, sakila, sys, tienda, tiendaelectronica, and Users.
- Central Panel (SQL Editor):** A query window titled "DataSolutionsDB" containing the following SQL code:

```
USE DataSolutionsDB;
CREATE TABLE clientes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    apellido VARCHAR(100),
    ciudad VARCHAR(100),
    fecha_registro DATE
);
```
- Bottom Panel (Estadísticas 1):** Statistics for the query execution:

Name	Value
Queries	5
Updated Rows	0
Execute time	0.014s
Fetch time	0.000s
Total time	0.014s
Start time	2025-04-25 13:01:31.073
Finish time	2025-04-25 13:02:30.171

[Ver imagen a tamaño real](#)

5. Crear índice para optimizar consultas

The screenshot shows the DBeaver 25.0.3 interface. The left sidebar displays the project structure under 'Proyectos' and the database structure under 'DataSolutionsDB'. The main area shows a SQL query window with the following code:

```
CREATE INDEX idx_ciudad_fecha ON clientes (ciudad, fecha_registro);
```

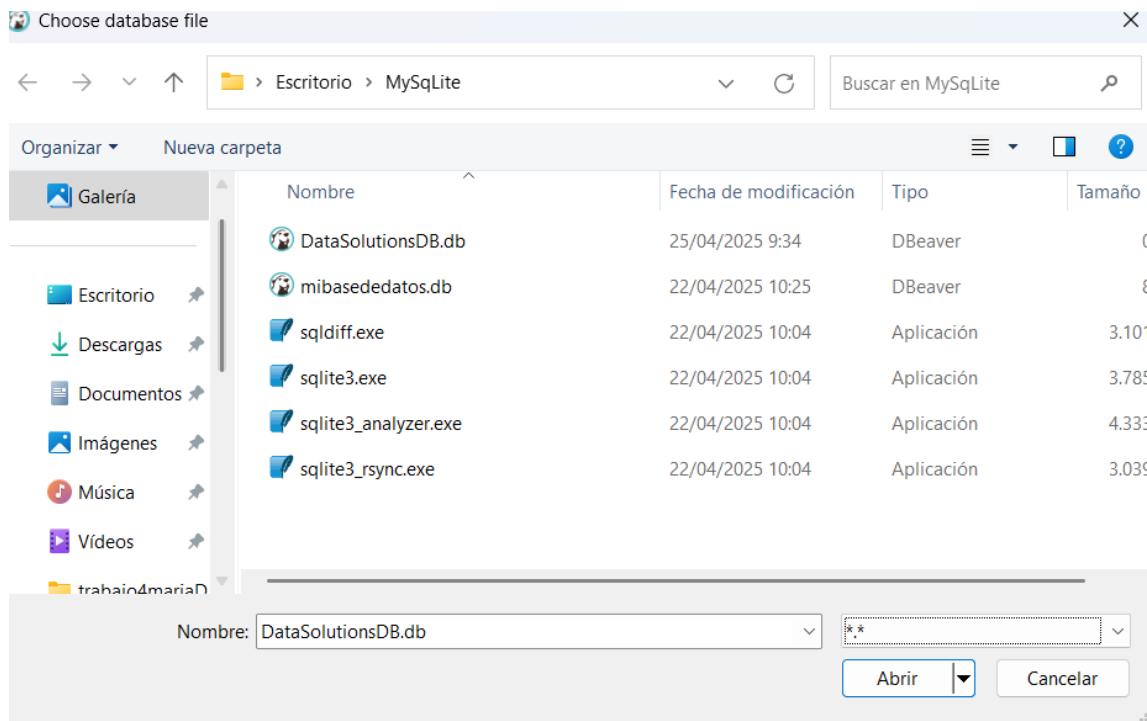
Below the query window, there is a statistics panel titled 'Estadísticas 1' containing the following information:

Name	Value
Updated Rows	5
Execute time	0.006s
Start time	Fri Apr 25 13:17:49 CEST 2025
Finish time	Fri Apr 25 13:17:49 CEST 2025
Query	INSERT INTO clientes (nombre, apellido, ciudad, fecha_registro) VALUES ('Ana', 'López', 'Madrid', '2024-02-01'), ('Luis', 'Gómez', 'Barcelona', '2024-03-15'), ('Marta', 'Ruiz', 'Valencia', '2024-01-10'), ('Carlos', 'Pérez', 'Madrid', '2024-04-05'), ('Elena', 'Martínez', 'Sevilla', '2023-12-30')

[Ver imagen a tamaño real](#)

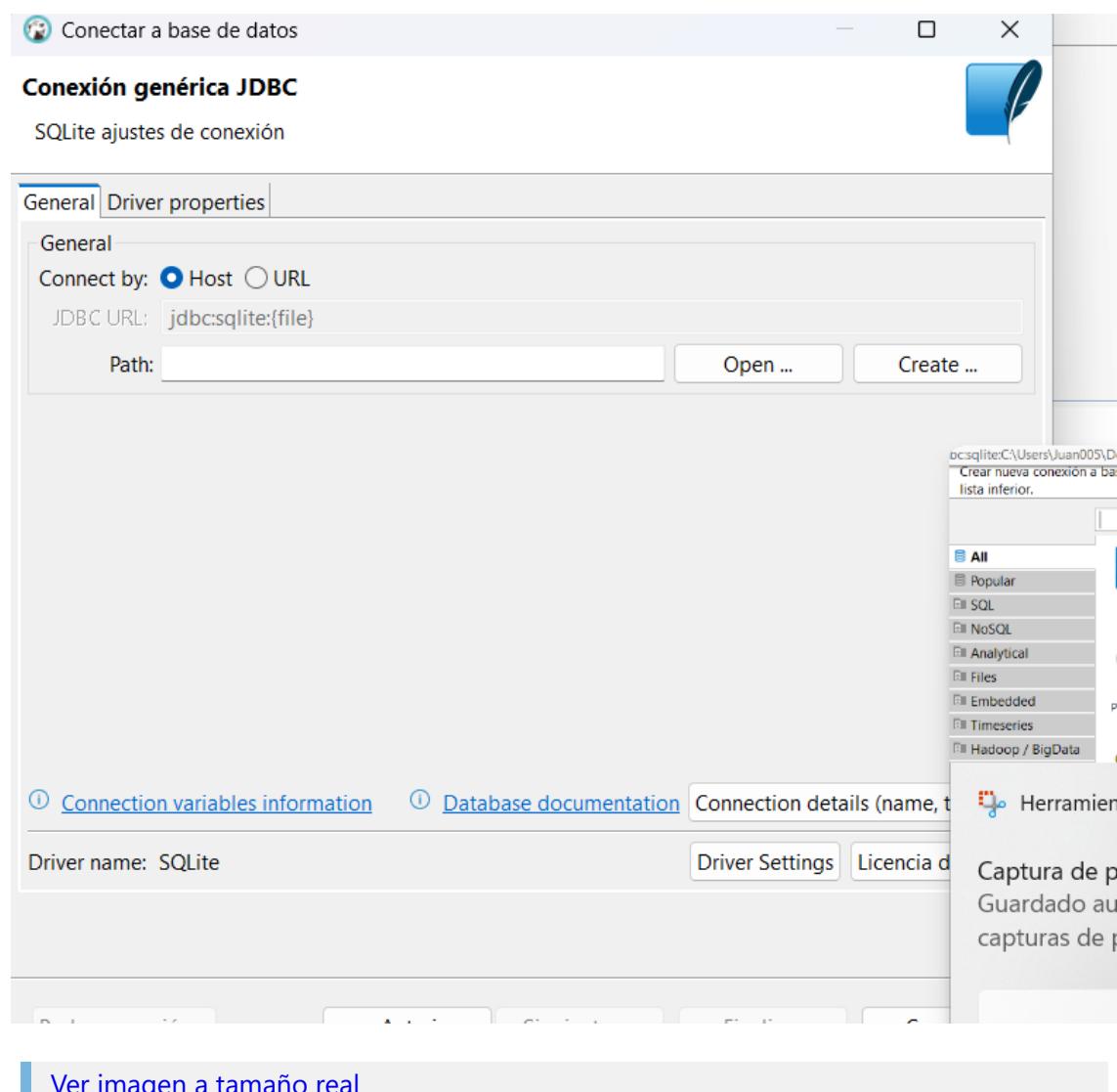
Parte 2 – SQLite: Creación e Inserción de Datos

6. Crear base de datos SQLite (nombre)



[Ver imagen a tamaño real](#)

7. Definir ruta del archivo SQLite



8. Crear base de datos SQLite

bc:sqlite:C:\Users\Juan005\Desktop\MySqlLite\mibasededatos.db

Crear nueva conexión a base de datos. Encuentre su driver de la base de datos en la lista inferior.

Sort by: Title Score

All

Popular

SQL

NoSQL

Analytical

Files

Embedded

Timeseries

Hadoop / BigData

Full-text search

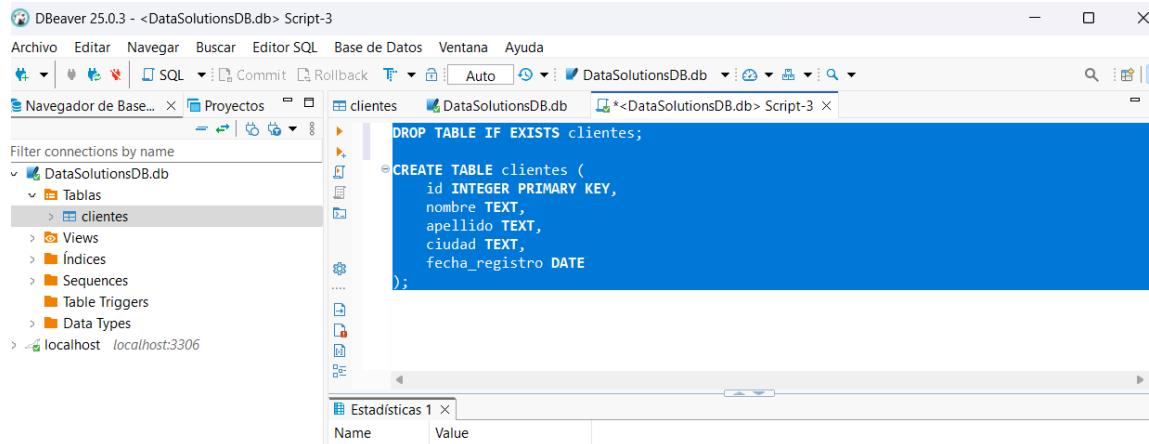
Graph databases

The screenshot shows a 'Create New Connection' dialog. At the top, there's a text input field containing 'bc:sqlite:C:\Users\Juan005\Desktop\MySqlLite\mibasededatos.db'. Below it is a note: 'Crear nueva conexión a base de datos. Encuentre su driver de la base de datos en la lista inferior.' A 'Sort by' dropdown is set to 'Score'. On the left, a sidebar lists categories like All, Popular, SQL, NoSQL, etc. The main area displays a grid of database icons and names:

Icon	Name
SQLite icon	SQLite
MariaDB icon	MariaDB
IBM DB2 icon	Db2 for LUW
MySQL icon	MySQL
Oracle icon	Oracle
PostgreSQL icon	PostgreSQL
Microsoft SQL Server icon	SQL Server
Altibase icon	Altibase
Apache Calcite Avatica icon	Apache Calcite Avatica
Apache Drill icon	Apache Drill
Apache Hive 2 icon	Apache Hive 2
Apache Hive 4+ icon	Apache Hive 4+
Apache Ignite icon	Apache Ignite
Apache Kylin icon	Apache Kylin
Apache Kyuubi icon	Apache Kyuubi
Apache Phoenix icon	Apache Phoenix
Apache Spark icon	Apache Spark
Athena icon	Athena
Azure icon	Azure
Microsoft SQL Server icon	Azure SQL

At the bottom are buttons: 'Probar conexión ...', '< Anterior', 'Siguiente >', 'Finalizar', and 'Cancelar'. A link 'Ver imagen a tamaño real' is at the bottom left.

9. Crear tabla clientes SQLite (con DROP IF EXISTS)



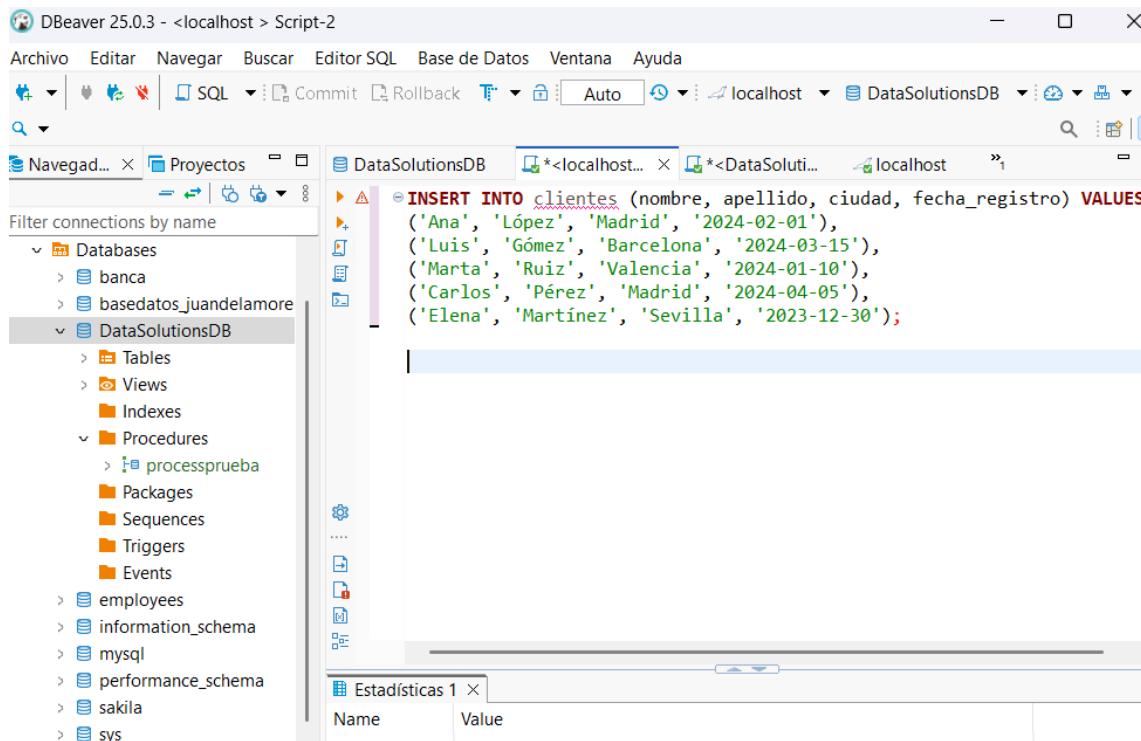
The screenshot shows the DBeaver 25.0.3 interface. The top menu bar includes Archivo, Editar, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, and Ayuda. The toolbar has icons for connection management, SQL editor, commit, rollback, and auto-commit. The main window has tabs for 'Navegador de Base...' and 'Proyectos'. A tree view under 'DataSolutionsDB.db' shows 'Tablas' with 'clientes' selected. The central SQL editor tab contains the following code:

```
DROP TABLE IF EXISTS clientes;
CREATE TABLE clientes (
    id INTEGER PRIMARY KEY,
    nombre TEXT,
    apellido TEXT,
    ciudad TEXT,
    fecha_registro DATE
);
```

Below the SQL editor is a statistics panel titled 'Estadísticas 1' with columns for Name and Value.

[Ver imagen a tamaño real](#)

10. Insertar registros de prueba



DBeaver 25.0.3 - <localhost> Script-2

Archivo Editar Navegar Buscar Editor SQL Base de Datos Ventana Ayuda

Navegad... Proyectos DataSolutionsDB *<localhost... <DataSoluti... localhost

Filter connections by name

Databases

- banca
- basedatos_juandelamore
- DataSolutionsDB
 - Tables
 - Views
 - Indexes
 - Procedures
 - processprueba
 - Packages
 - Sequences
 - Triggers
 - Events
- employees
- information_schema
- mysql
- performance_schema
- sakila
- sys

INSERT INTO clientes (nombre, apellido, ciudad, fecha_registro) VALUES ('Ana', 'López', 'Madrid', '2024-02-01'), ('Luis', 'Gómez', 'Barcelona', '2024-03-15'), ('Marta', 'Ruiz', 'Valencia', '2024-01-10'), ('Carlos', 'Pérez', 'Madrid', '2024-04-05'), ('Elena', 'Martínez', 'Sevilla', '2023-12-30');

Estadísticas 1

Name	Value
...	...

[Ver imagen a tamaño real](#)

Parte 3 – Automatización y Seguridad

11. Backup manual de MySQL

The screenshot shows the DBeaver interface with the following details:

- Toolbar:** Archivo, Editar, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, Ayuda.
- SQL Editor:** Shows the query: `CREATE TABLE clientes_backup AS SELECT * FROM clientes;`
- Database Navigator:** Shows the connection to `localhost` (`localhost:3306`) and the schema `datasolutionsdb`. The table `clientes` is selected, showing 32K rows.
- Statistics:** A table titled "Estadísticas 1" provides execution details:

Name	Value
Updated Rows	5
Execute time	0.019s
Start time	Fri Apr 25 14:08:48 CEST 2025
Finish time	Fri Apr 25 14:08:48 CEST 2025
Query	CREATE TABLE clientes_backup AS SELECT * FROM clientes
- Project View:** Shows the project structure with "Project - General" and "DataSources".
- Object Navigator:** Shows the objects in the `DataSolutionsDB.db` database, including Tables, Views, Indices, and Sequences.
- Bottom Bar:** Includes a link to "Ver imagen a tamaño real" (View image at full size).

12. Crear procedimiento almacenado (MySQL)

The screenshot shows the DBeaver interface with the following details:

- Toolbar:** Archivo, Editar, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, Ayuda.
- Connections:** localhost > Script-2, DataSolutionsDB.db, localhost:3306.
- Schemas:** DataSolutionsDB.db, localhost, DataSolutionsDB, DataSolutionsDB.db, localhost, DataSolutionsDB.
- Left Panel (Navegador de Bases de Datos):** Shows the database structure:
 - localhost
 - Databases
 - banca
 - basedatos_juandelamorena
 - datasolutionsdb
 - Tables
 - clientes
 - clientes_backup
 - Views
 - Indexes
 - Procedures
 - insertar_cliente
 - Packages
 - Triggers
 - Events
 - employees
 - information_schema
 - mysql
 - performance_schema
 - sakila
 - sys
 - tienda
- Right Panel (Editor SQL):** Script-2, DataSolutionsDB.db, Script-3.

```
DELIMITER //
CREATE PROCEDURE insertar_cliente (
    IN p_nombre VARCHAR(50),
    IN p_apellido VARCHAR(50),
    IN p_ciudad VARCHAR(50),
    IN p_fecha DATE
)
BEGIN
    IF p_fecha > CURDATE() THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La fecha no puede ser futura.';
    ELSE
        INSERT INTO clientes (nombre, apellido, ciudad, fecha_registro)
        VALUES (p_nombre, p_apellido, p_ciudad, p_fecha);
    END IF;
END //
DELIMITER ;
```

[Ver imagen a tamaño real](#)

13. Ejecutar procedimiento almacenado - correcto

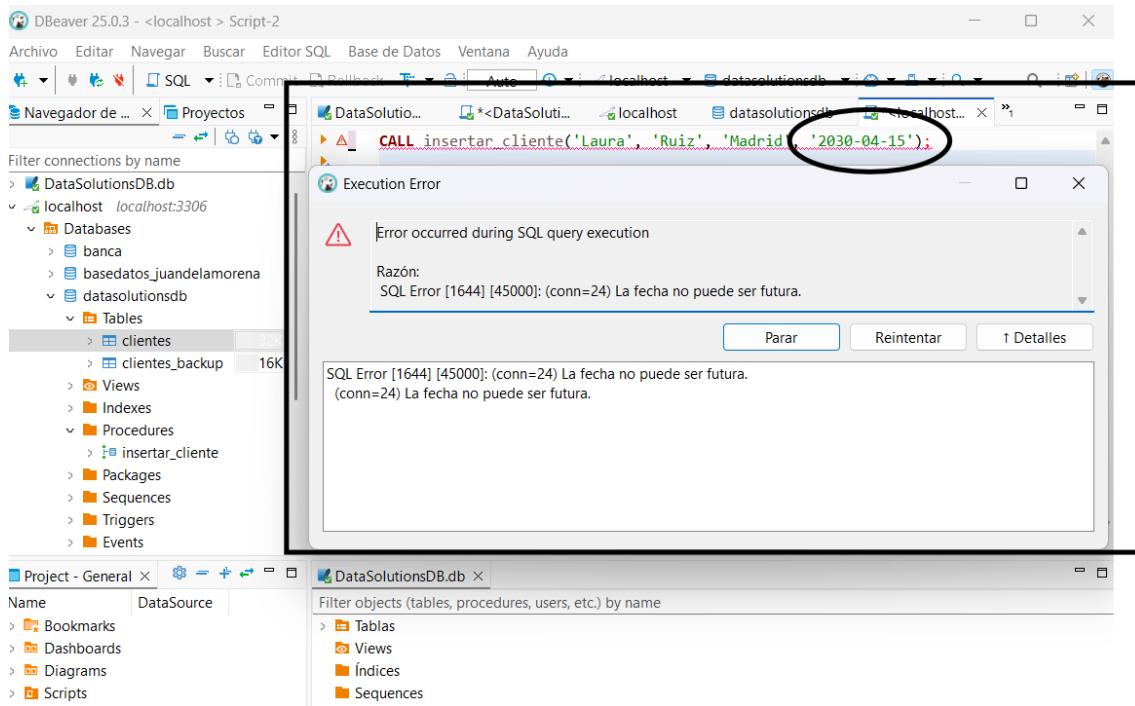
The screenshot shows the DBeaver interface with the following details:

- SQL Editor:** Contains the SQL command: `CALL insertar_cliente('Laura', 'Ruiz', 'Madrid', '2024-04-15');`
- Statistics Window:** Titled "Estadísticas 1", it displays the following information:

Name	Value
Updated Rows	1
Execute time	0.009s
Start time	Fri Apr 25 14:15:25 CEST 2025
Finish time	Fri Apr 25 14:15:25 CEST 2025
Query	CALL insertar_cliente('Laura', 'Ruiz', 'Madrid', '2024-04-15')

[Ver imagen a tamaño real](#)

14. Ejecutar procedimiento almacenado - error



[Ver imagen a tamaño real](#)

15. Mostrar procesos en ejecución

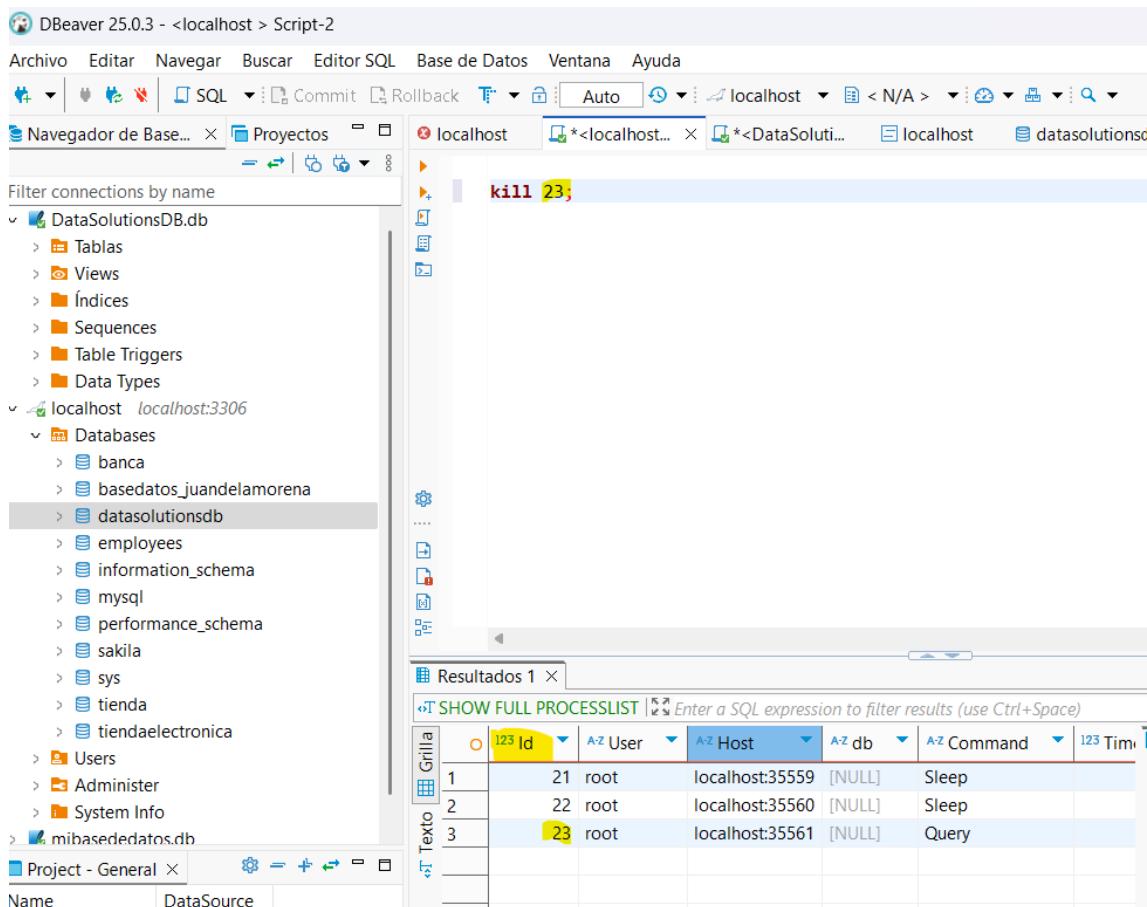
The screenshot shows the DBBeaver interface with the following details:

- Toolbar:** Archivo, Editor, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, Ayuda.
- Project:** DataSolutionsDB
- Database:** DataSolutionsDB
- SQL Editor:** Contains the query `SHOW FULL PROCESSLIST;`
- Results Grid:** Shows the output of the query:

Grilla	User	Host	db	Command	Time	State	Info	Progress	Valor
1	[User]	localhost:32784	(NULL)	Sleep	1.401	[NULL]	[NULL]	0	9
2	[User]	localhost:32785	(NULL)	Sleep	1.401	[NULL]	[NULL]	0	0
3	[User]	localhost:33419	datasolutionsdb	Query	0	starting	SHOW FULL PROCES	0	

[Ver imagen a tamaño real](#)

16. Kill de procesos bloqueados o pesados (slow_process)



DBeaver 25.0.3 - <localhost> Script-2

Archivo Editar Navegar Buscar Editor SQL Base de Datos Ventana Ayuda

Navegador de Base... Proyectos

localhost kill 23

Filter connections by name

DataSolutionsDB.db

- Tablas
- Views
- Índices
- Sequences
- Table Triggers
- Data Types

localhost localhost:3306

- Databases
 - banca
 - basedatos_juandelamorena
 - datasolutionsdb**
 - employees
 - information_schema
 - mysql
 - performance_schema
 - sakila
 - sys
 - tienda
 - tiendaelectronica
- Users
- Administer
- System Info

mibasededatos.db

Project - General

Resultados 1

SHOW FULL PROCESSLIST

Id	User	Host	db	Command	Time
1	21	root	localhost:35559	[NULL]	Sleep
2	22	root	localhost:35560	[NULL]	Sleep
3	23	root	localhost:35561	[NULL]	Query

[Ver imagen a tamaño real](#)

17. Crear Trigger de auditoría

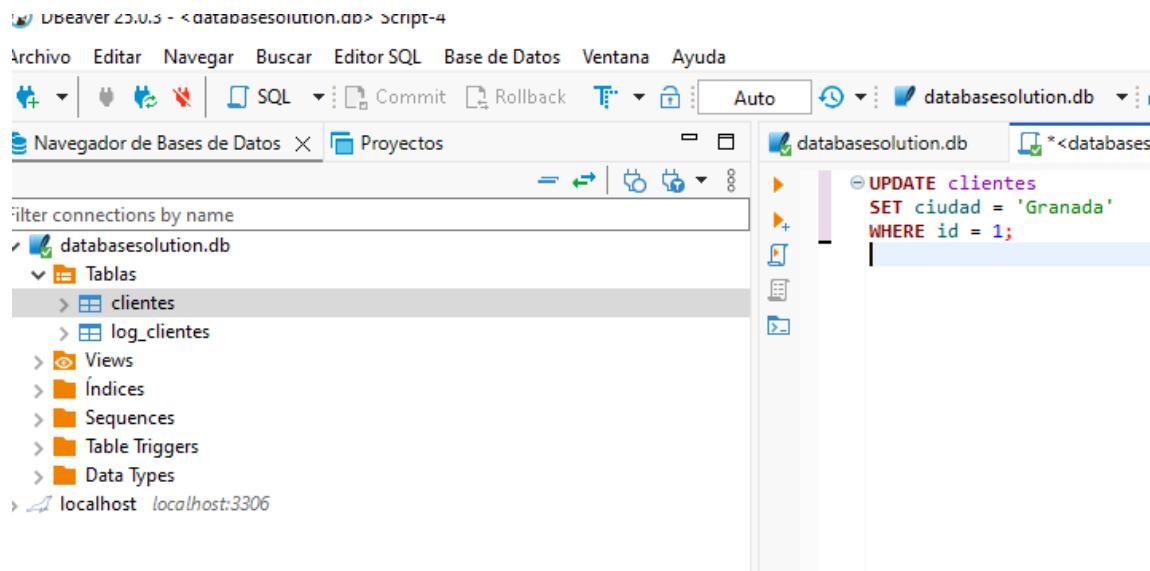
The screenshot shows a database interface with a left sidebar and a main panel. The sidebar lists connections and the current database 'databasesolution.db'. Under 'Tablas', it shows the 'clientes' table with its columns, keys, indices, and triggers. The 'Triggers' section contains one item: 'trigger_log_update'. The main panel is titled 'trigger_log_update' and shows the following details:

- Name: trigger_log_update
- Table: clientes
- Description: (empty)
- Source code:

```
CREATE TRIGGER trigger_log_update
AFTER UPDATE ON clientes
BEGIN
    INSERT INTO log_clientes (
        fecha_modificacion, usuario, operacion,
        datos_antiguos, datos_nuevos
    ) VALUES (
        datetime('now'),
        'usuario_windows',
        'UPDATE',
        json_object(
            'id', OLD.id,
            'nombre', OLD.nombre,
            'apellido', OLD.apellido,
            'ciudad', OLD.ciudad,
            'fecha_registro', OLD.fecha_registro
        ),
        json_object(
            'id', NEW.id,
            'nombre', NEW.nombre,
            'apellido', NEW.apellido,
            'ciudad', NEW.ciudad,
            'fecha_registro', NEW.fecha_registro
        )
    );
END;
```

[Ver imagen a tamaño real](#)

18. Ejecutar UPDATE para probar Trigger



[Ver imagen a tamaño real](#)

19. Visualizar registros en log_clientes

The screenshot shows a database management interface with the following details:

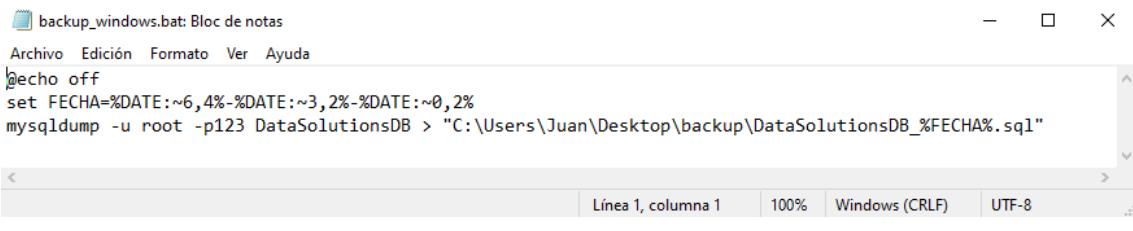
- Toolbar:** Archivo, Editar, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, Ayuda.
- Project Bar:** Navegador de Bases de Datos, Projects, databasesolution.db, databasesolution.db Script-4, trigger_log_update, clientes, log_clientes.
- Table Structure:** Shows the structure of the log_clientes table with columns: fecha_modificacion, usuario, operacion, datos_antiguos, and datos_nuevos.
- Data Grid:** Displays two rows of data from the log_clientes table.
- Left Sidebar:** Shows the database structure for databasesolution.db, including tables (clientes), triggers (trigger_log_update), and other objects like Views, Indices, Sequences, Table Triggers, and Data Types.
- Bottom Status:** localhost:3306.

	fecha_modificacion	usuario	operacion	datos_antiguos	datos_nuevos
1	2025-04-27 13:58:35	usuario_windows	UPDATE	{"id":3,"nombre":"Luis","apellido":"Martí"} {"id":3,"nombre": "Luis", "apellido": "Martí"}	2025-04-27 13:58:35
2	2025-04-27 13:59:07	usuario_windows	UPDATE	{"id":1,"nombre":"Juan","apellido":"Pérez"} {"id":1,"nombre": "Juan", "apellido": "Pérez"}	

[Ver imagen a tamaño real](#)

Parte 4 – Planificación de Tareas y Backup

20. Script .bat de backup Windows



backup_windows.bat: Bloc de notas

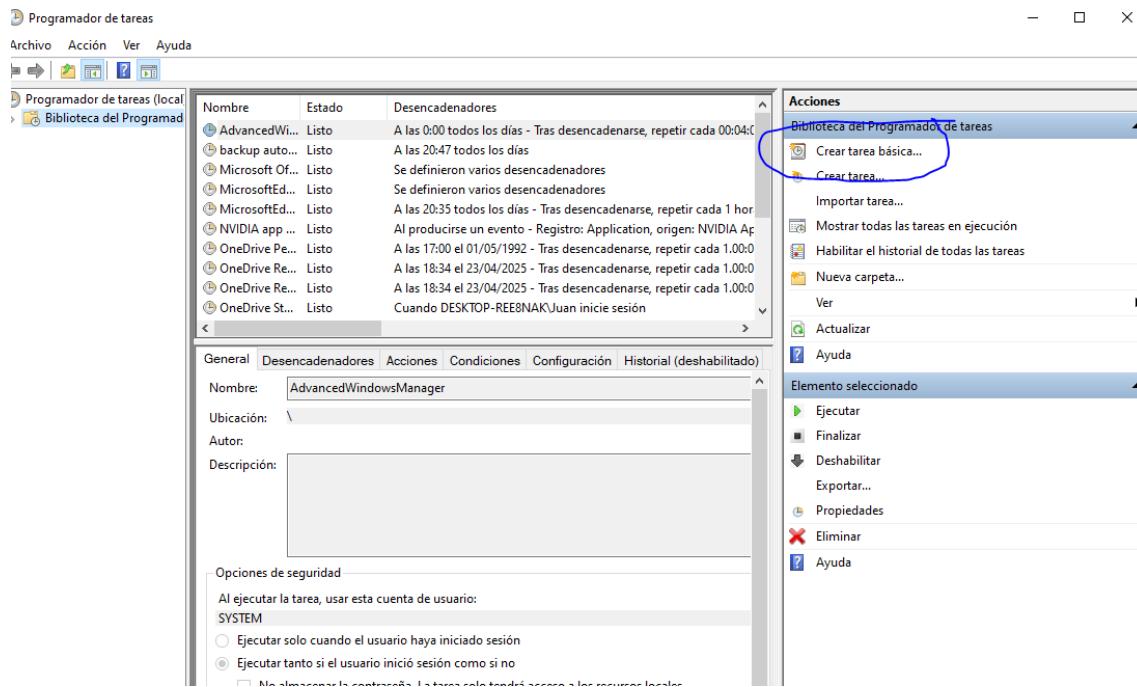
Archivo Edición Formato Ver Ayuda

```
@echo off
set FECHA=%DATE:~6,4%-%DATE:~3,2%-%DATE:~0,2%
mysqldump -u root -p123 DataSolutionsDB > "C:\Users\Juan\Desktop\backup\DataSolutionsDB_%FECHA%.sql"
```

Línea 1, columna 1 | 100% | Windows (CRLF) | UTF-8

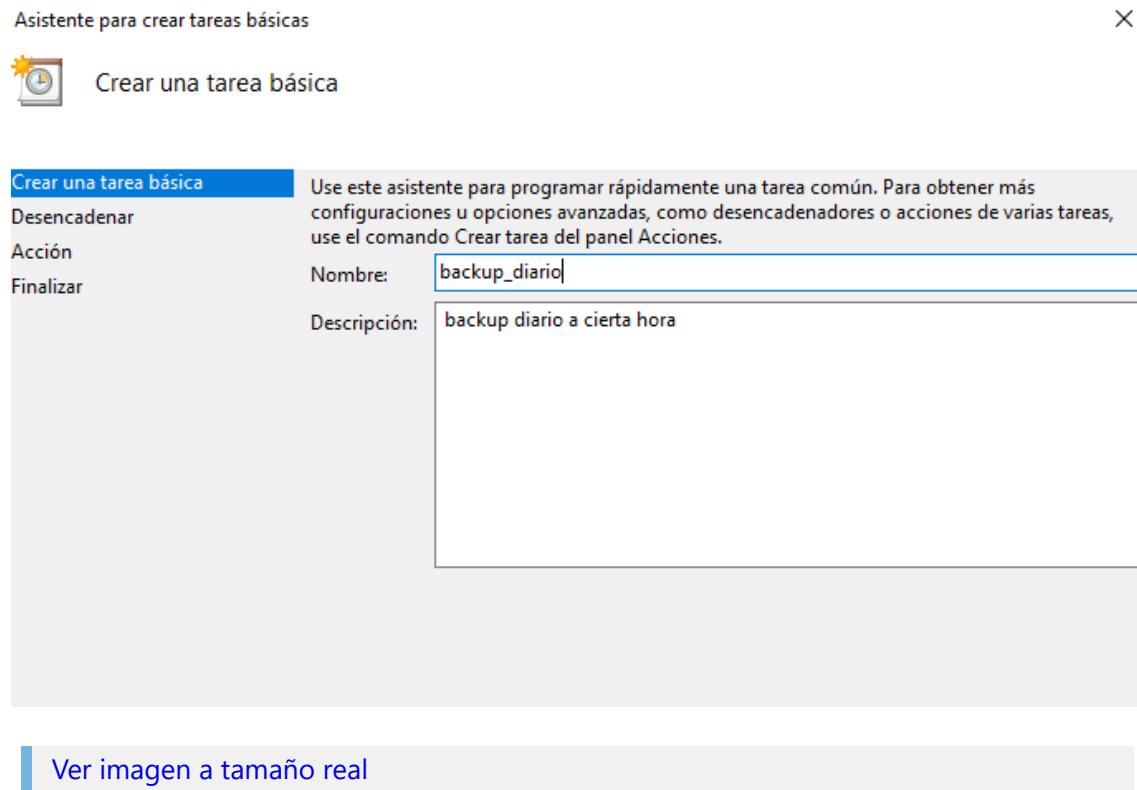
[Ver imagen a tamaño real](#)

21. Crear tarea programada en Windows



[Ver imagen a tamaño real](#)

22. Nombre y descripción de la tarea



23. Asignación de periodicidad diaria

Asistente para crear tareas básicas

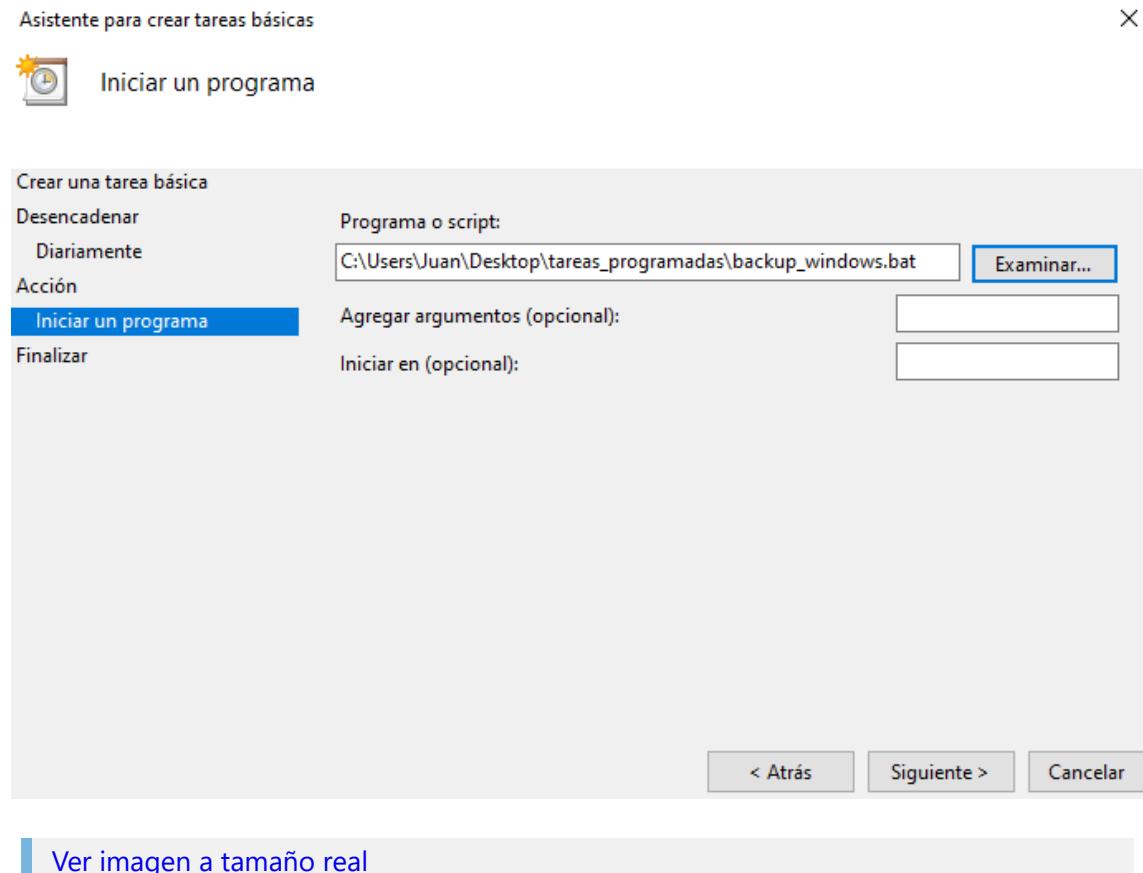


Desencadenador de tarea

Crear una tarea básica	¿Cuándo desea que se inicie la tarea?
Desencadenar	<input checked="" type="radio"/> Diariamente
Acción	<input type="radio"/> Semanalmente
Finalizar	<input type="radio"/> Mensualmente
	<input type="radio"/> Una vez
	<input type="radio"/> Al iniciarse el equipo
	<input type="radio"/> Al iniciar sesión
	<input type="radio"/> Cuando se registre un evento específico

[Ver imagen a tamaño real](#)

24. Seleccionar el script .bat



25. Opciones avanzadas: ejecutar con privilegios

Asistente para crear tareas básicas



Acción

Crear una tarea básica

Desencadenar

Diariamente

Acción

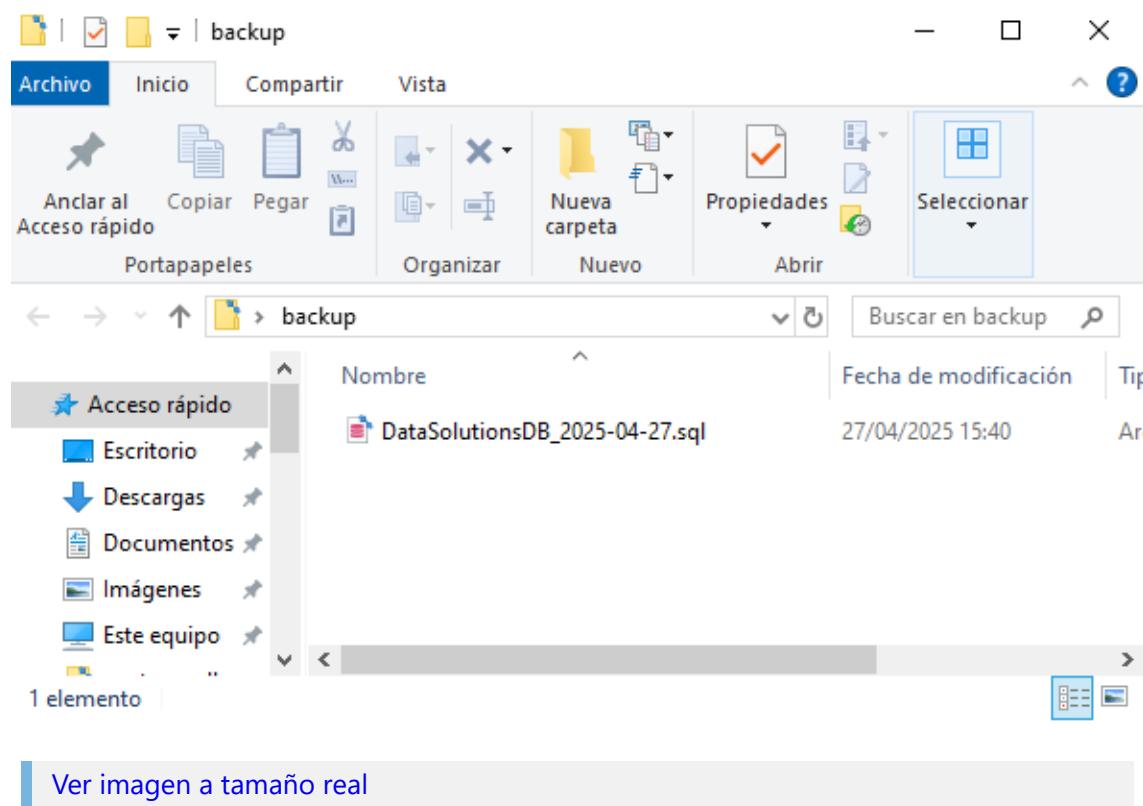
Finalizar

¿Qué acción desea que realice la tarea?

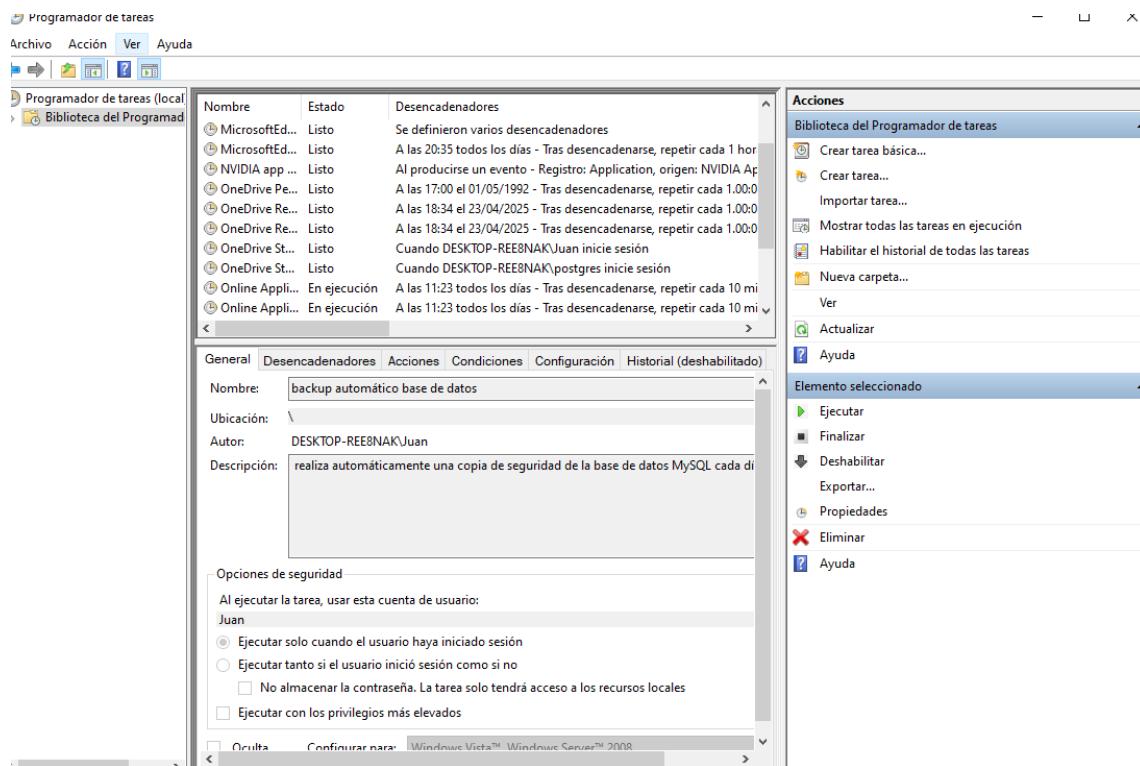
- Iniciar un programa
- Enviar un correo electrónico (desusado)
- Mostrar un mensaje (desusado)

[Ver imagen a tamaño real](#)

26. Resultado de la tarea progr



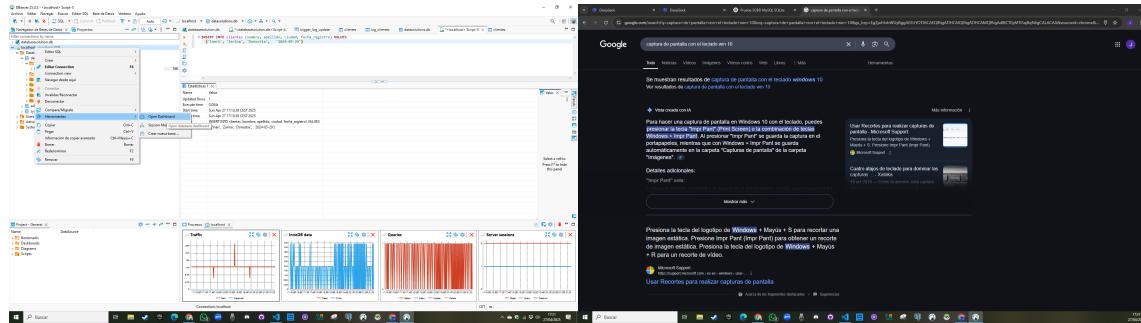
27. Backup generado automáticamente



[Ver imagen a tamaño real](#)

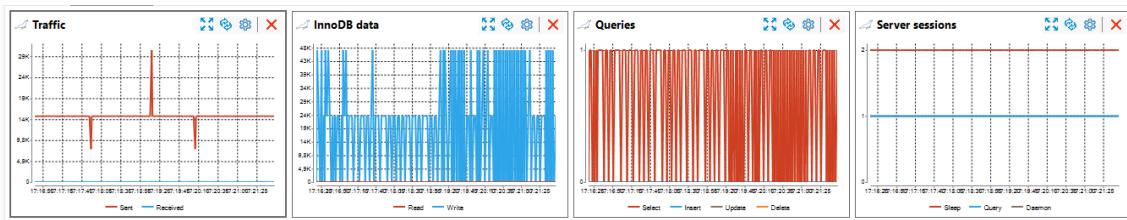
Parte 5 – Monitorización del Rendimiento

28. Acceso al Dashboard de DBeaver



[Ver imagen a tamaño real](#)

29. Vista general del Dashboard de Monitorización



[Ver imagen a tamaño real](#)

30. Estado general del servidor (Server Status)

The screenshot shows the DBBeaver interface with the following details:

- Toolbar:** Archivo, Editar, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, Ayuda.
- Connections:** localhost, datasolutions.db.
- SQL Editor:** Shows the query `SHOW STATUS LIKE '%slow_queries%';`.
- Result Grid:** A table with one row and two columns. The first column is labeled "Variable_name" and the second is "Value". The row contains "Slow_queries" and "0". This grid is circled in blue.
- Message Bar:** Shows "1 row(s) fetched - 0.042s, on 2025-04-27 at 16:12:33".
- Bottom Buttons:** CET, es, Editable, Insertión inteligente.

[Ver imagen a tamaño real](#)

31. Monitorizar procesos activos (SHOW PROCESSLIST)

The screenshot shows the DBeaver interface with the following details:

- Toolbar:** Archivo, Editar, Navegar, Buscar, Editor SQL, Base de Datos, Ventana, Ayuda.
- Connections:** localhost (localhost:3306), databasesolution.db, datasolutions.db (selected).
- SQL Editor:** Query: SHOW PROCESSLIST; Results: Salida X
- Resultados 1:** A grid showing the following data:

Id	User	Host	db	Command
1	event_scheduler	localhost	[NULL]	Daemon
25	root	localhost:53893	[NULL]	Sleep
26	root	localhost:53894	[NULL]	Sleep
28	root	localhost:53928	datasolutions	Query

Project - General: Shows Bookmarks, Dashboards, Diagrams, Scripts.

[Ver imagen a tamaño real](#)

32. EXPLAIN de consultas para optimización

The screenshot shows the DBeaver 25.0.3 interface. In the top navigation bar, the connection is set to 'localhost' and the database is 'datasolutions.db'. The main window displays the results of the EXPLAIN command for the query 'EXPLAIN SELECT * FROM clientes WHERE ciudad = 'Madrid''. The results are shown in a table:

id	select_type	table	partitions	type	possible_keys
1	SIMPLE	clientes	[NULL]	ALL	[NULL]

Below the table, a message indicates '1 row(s) fetched - 0.005s, on 2025-04-27 at 17:07:44'. The bottom status bar shows 'CET | es | Editable' and 'Inserción inteligente | 2:1'.

[Ver imagen a tamaño real](#)

Conclusiones

Este proyecto ha permitido:

- Aplicar de forma práctica los conceptos de diseño y administración de SGBD
- Automatizar tareas críticas de respaldo y validación
- Implementar auditoría y control de acceso
- Documentar profesionalmente un proyecto técnico en GitHub

Mejoras futuras: Uso de Docker para entornos reproducibles y Percona PMM para monitorización avanzada.

⚙️ Herramientas Utilizadas

- **DBeaver**
- **MySQL Server**
- **SQLite3**
- **Visual Studio Code + Markdown PDF**
- **PowerShell / Programador de Tareas (Windows)**
- **Git & GitHub**

🔗 Repositorio de consulta: >

https://github.com/JuanHoob/Trabajo_IFCT0310_1470.git