



ORION
SOFTWARE
CONSULTING

Orion MySQL Studio

Manual de usuario

1.2.0 · .NET 10 · Windows · Linux · Android

Contenido

1. Qué hace y qué necesita
2. Instalación y primer arranque
3. Cómo moverse por la ventana
4. Conexiones
5. Explorador
6. Comparar esquema
7. Aplicar al destino
8. Comparar datos
9. Multi-tenant
10. Snapshots
11. Editor SQL
12. Proyectos
13. Reportes y eventos al terminar
14. Línea de comandos
15. Android
16. Reglas de seguridad
17. Problemas frecuentes
18. Dónde guarda sus cosas

1. Qué hace y qué necesita

Orion MySQL Studio compara dos bases de datos —de **MySQL** o de **MariaDB**—, le enseña en qué se diferencian y escribe el script SQL que deja la segunda igual que la primera. Usted lo lee, lo edita si quiere, y decide si se aplica.

Compara dos cosas distintas, y conviene no confundirlas: el **esquema** —la estructura: tablas, columnas, índices, vistas, rutinas, permisos— y los **datos** —las filas de cada tabla—. Son dos pantallas y dos scripts diferentes.

Lo que no hace, y conviene saberlo desde el principio

- **No aplica nada sin que usted lo confirme** viendo antes el servidor, la base, el usuario y el script completo.
- **No copia contraseñas.** Al sincronizar cuentas de usuario, el `CREATE USER` sale con una contraseña marcador que usted debe cambiar.
- **No aplica cambios desde Android.** Allí puede comparar y generar el script; aplicarlo es solo de escritorio. Es una decisión de seguridad.
- **No necesita que instale nada más.** Ni cliente de MySQL, ni controladores, ni .NET.

Qué hace falta

En su equipo	Windows 10 u 11 de 64 bits, o Linux de 64 bits. Android 8 o superior para la aplicación móvil. Unos 250 MB de disco.
En el servidor	MySQL 5.7 o superior, o MariaDB 10.3 o superior, con su puerto accesible desde su equipo (3306 de fábrica).
Permisos para comparar	Un usuario con <code>SELECT</code> sobre <code>information_schema</code> y sobre las bases que vaya a comparar.
Permisos para aplicar	Los que exija cada cambio: <code>ALTER</code> , <code>CREATE</code> , <code>DROP</code> , <code>INDEX</code> ... El producto no pide más de lo que el script necesita.
Permisos para comparar cuentas	<code>SELECT</code> sobre <code>mysql.*</code> . Si no los tiene, la comparación sigue sin esa parte y lo avisa; no falla.

El producto se conecta como cualquier cliente de MySQL. **No instala nada en el servidor** ni le pide que active una extensión.

2. Instalación y primer arranque

Windows

Ejecute `orion-mysql-studio-1.2.0-win-x64-setup.msi`. Instala la aplicación y la consola `orion-mysql`, y crea los accesos directos. Para desinstalar, «Aplicaciones instaladas» de Windows.

Si prefiere no instalar nada, descargue el `.zip` portable y descomprímalo donde quiera. Son los mismos dos ejecutables.

Mientras los paquetes no estén firmados con certificado, Windows avisará de «editor desconocido» al abrirlos. Es normal y no indica un problema.

Linux

```
tar -xzf orion-mysql-studio-1.2.0-linux-x64.tar.gz
cd "Orion MySQL Studio 1.2.0"
./OrionMySqlStudio
```

El `.tar.gz` ya trae el bit de ejecución puesto. No hace falta ser administrador ni instalar dependencias.

Android

Instale el `.apk`. Android pedirá permiso para instalar desde esta fuente. La aplicación consulta, explora, compara y comparte el `.sql`, pero **no aplica cambios** a ninguna base.

La prueba gratuita y la licencia

Al abrirlo por primera vez, la barra superior dice «Sin activar». Pulse ahí —o **Ayuda > Licencia**— y rellene su nombre y su correo para pedir la **prueba gratuita de 30 días**. La activación necesita conexión a internet una sola vez; después funciona sin ella.

La licencia es **única y perpetua**, con todas las funciones, y vale para **dos equipos**; un teléfono Android cuenta como equipo. En la misma pantalla está «**Liberar este equipo**», que devuelve el cupo al servidor si va a cambiar de máquina.

Al vencer la prueba, el producto pasa a **solo lectura**: puede explorar, ver DDL y comparar, pero no generar ni aplicar scripts. No se pierde ninguna conexión ni ningún proyecto.

3. Cómo moverse por la ventana

La ventana tiene siempre las mismas cuatro piezas. Conviene reconocerlas antes de empezar, porque el resto del manual las nombra sin explicarlas.

Barra superior	Logotipo, menú —Archivo, Editar, Ver, Herramientas, Ayuda—, estado de la licencia y el selector de tema.
Barra lateral	Las ocho pantallas: Conexiones, Proyectos, Explorador, Comparar esquema, Comparar datos, Multi-tenant, Editor SQL y Snapshots.
Tira de ejecuciones	Aparece bajo la barra superior solo cuando hay proyectos ejecutándose , con una pestaña por cada uno.
Barra de estado	Abajo: el último mensaje, el proyecto abierto y el tema activo.

El menú, elemento por elemento

Elemento	Qué hace
Archivo › Nuevo proyecto	Vacía origen, destino y opciones. Pregunta antes si hay un proyecto abierto. Ctrl+N
Archivo › Abrir proyecto	Carga un <code>.omsproj</code> y vuelca sus dos lados y sus opciones en la pantalla que le corresponde. Ctrl+O
Archivo › Guardar proyecto	Guarda lo que hay en pantalla. La primera vez pregunta dónde. Ctrl+S
Editar › Buscar	Lleva el foco a la caja de búsqueda de la pantalla activa. Ctrl+F
Ver › Skin	Los cinco temas, con su color de acento. Se aplica en caliente y se recuerda.
Ver › Alternar tema	Cambia entre «Noche Orion» y «Claro».
Herramientas › Opciones de comparación	Las opciones de esquema y de datos, con buscador y perfiles.
Ayuda › Manual y ayuda	La ayuda de la pantalla en la que esté, con su sección del manual. F1
Ayuda › Licencia	Activar, revalidar o liberar el equipo.
Ayuda › Acerca de	Versión, licencias de terceros y avisos de marcas.

Los cinco temas

Noche Orion (el de fábrica), **Claro**, **Grafito**, **Terminal** y **Alto contraste**. El cambio es inmediato y se guarda. En todos ellos el estado de una diferencia se escribe además con palabras: el color nunca es el único canal, de modo que el producto se puede usar sin distinguir tonos.

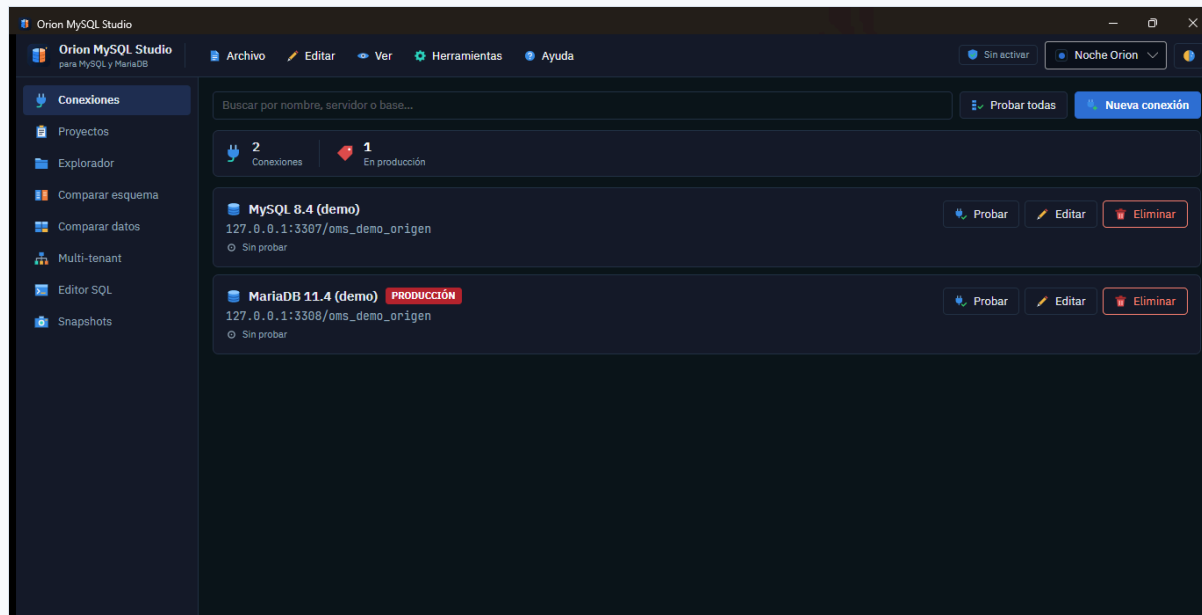
En un teléfono o una tableta

Por debajo de unos 1000 píxeles de ancho, la barra lateral se sustituye por una **navegación inferior** con las mismas pantallas; por debajo de 600, desaparecen también el menú y la barra de estado. Es la misma aplicación, no una versión recortada.

4. Conexiones

Una conexión es un servidor guardado. Se crean una vez y se usan desde todas las pantallas.

CAPTURA 01



La pantalla de Conexiones: una tarjeta por servidor, con su resumen arriba y la etiqueta PRODUCCIÓN donde corresponde.

Crear una conexión

1. Pulse «Nueva conexión».
2. Rellene nombre, servidor, puerto, usuario y contraseña. La **base por omisión** es opcional: solo decide qué se propone al abrir el árbol.
3. Si el servidor exige cifrado, elija el **modo TLS** y, si hace falta, el certificado de la autoridad.
4. Si el servidor no está expuesto a internet, active el **túnel SSH** y ponga su equipo, usuario y llave privada.
5. Marque **PRODUCCIÓN** si la base no se puede romper. Lea abajo lo que eso cambia.
6. Pulse «Probar» y, si contesta, «Guardar».

Qué significa marcar PRODUCCIÓN

- La conexión lleva una **etiqueta roja permanente** en todas las pantallas donde aparece.
- Para aplicar cualquier cosa contra ella hay que **escribir el nombre de la base** en la confirmación.

- En el editor SQL, las sentencias que modifican datos o estructura quedan **bloqueadas** salvo que active «Permitir modificaciones».

Cuando «Probar» falla

El mensaje distingue la causa en vez de decir «no se pudo conectar». Estas son las que más aparecen:

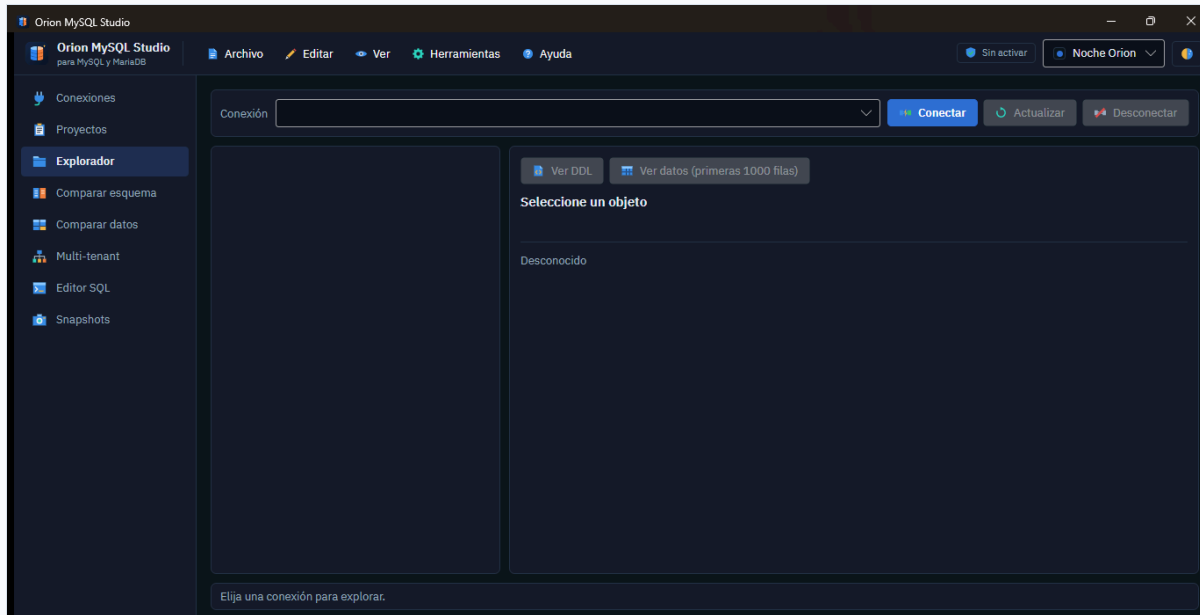
Lo que dice	Qué suele ser
El nombre del servidor no se resuelve	Un error de escritura en el host, o falta el sufijo del dominio.
El puerto está cerrado	El servidor no escucha ahí, o hay un cortafuegos en medio.
Usuario o contraseña incorrectos (1045)	Credenciales. Pruebe el mismo usuario desde otro cliente.
El usuario no tiene acceso a esa base (1044)	Las credenciales son buenas; faltan permisos sobre la base.
La base no existe (1049)	El nombre de la base por omisión está mal escrito.
Este equipo no está autorizado (1130)	El GRANT del usuario está acotado a otro host.
Se agotó la espera	Red lenta, VPN caída o el servidor no responde.

Las contraseñas **no se guardan en el archivo de conexiones**. Van cifradas en el almacén de secretos del sistema: DPAPI en Windows, un archivo con permisos 600 en Linux y el almacén de claves en Android. Al borrar una conexión se borran también sus secretos.

5. Explorador

El explorador recorre un servidor sin tener que escribir SQL. Sirve para mirar antes de comparar.

CAPTURA 02



El árbol del explorador, el panel de propiedades y la rejilla de datos de la tabla seleccionada.

1. Elija la conexión en el desplegable y pulse «**Conectar**».
2. Abra el nodo de la base. Las ramas son Tablas, Vistas, Procedimientos, Funciones, Triggers, Eventos y —en MariaDB— Secuencias, cada una con su número de objetos.
3. Al seleccionar un objeto, el panel de la derecha enseña sus propiedades: motor, filas aproximadas, tamaño, collation y fechas.

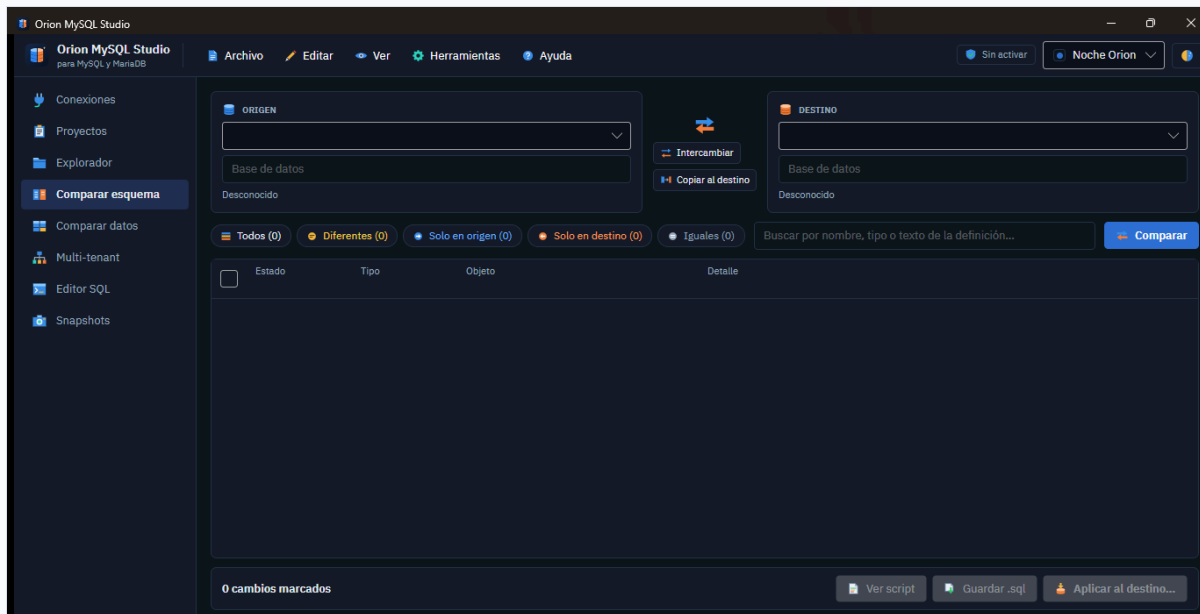
Con un objeto seleccionado, los dos botones de arriba hacen lo que más se usa: «**Ver DDL**» enseña su `CREATE` tal como lo devuelve el servidor, y «**Ver datos**» trae las **primeras 1000 filas**.

El árbol se carga **rama a rama**, según la abre. Una base con mil tablas aparece igual de rápido que una con diez, porque no se lee lo que no se mira.

6. Comparar esquema

Es la pantalla principal del producto. Compara la estructura de dos bases y escribe el script que lleva el destino al estado del origen.

CAPTURA 03



Comparar esquema: las tarjetas de origen y destino, los filtros por estado con sus conteos y la barra de acciones.

Paso a paso

1. Elija **origen** y **destino**: la conexión en el desplegable y el nombre de la base debajo. El origen es el que manda; el destino es el que cambiará.
2. Pulse «**Comparar**». El producto lee los dos catálogos y los compara.
3. Revise la lista. Use las **píldoras** para filtrar por estado y la caja de búsqueda para encontrar un objeto por nombre, por tipo o por **texto de su definición**.
4. Seleccione una fila para ver el **diff lado a lado**, con el origen a la izquierda y el destino a la derecha.
5. Marque o desmarque las casillas de lo que quiera llevar. La casilla de la cabecera marca y desmarca **todo lo visible**, no todo.
6. Pulse «**Ver script**» para leerlo, «**Guardar .sql**» para archivarlo o «**Aplicar al destino...**» para ejecutarlo.

Los cuatro estados

Estado	Qué significa	Qué hace el script
Diferente	El objeto está en los dos lados y no coincide.	<code>ALTER</code> con solo lo que cambió.
Solo en origen	Falta en el destino.	Lo crea.
Solo en destino	Sobra en el destino.	Lo borra. Destructivo .
Igual	Coincide. Se enseña para que conste.	Nada.

Dos iconos acompañan al nombre cuando hace falta: el de **destructivo** —aplicar eso puede perder datos— y el de **no compatible** —el objeto no se puede llevar al motor del destino, y queda fuera del script con su motivo escrito—.

Intercambiar y copiar

Entre las dos tarjetas hay dos botones que ahorran mucho clic. «**Intercambiar**» cambia origen por destino —y **descarta el resultado anterior** a propósito: dejarlo en pantalla sería enseñar algo que ya no corresponde a las tarjetas—. «**Copiar al destino**» pone en el destino la misma conexión y base del origen, que es por donde se empieza cuando se comparan dos bases del mismo servidor.

MySQL contra MariaDB

Si los dos lados son de motores distintos, sale un aviso amarillo y el script se escribe en el dialecto del **destino**. Lo que no es portable —el `JSON` nativo frente al `LONGTEXT` con `json_valid`, `CREATE SEQUENCE`, `WITH SYSTEM VERSIONING`, las columnas `INVISIBLE`— se marca «No compatible» y no entra en el script.

Las opciones de comparación

En **Herramientas > Opciones de comparación**. Deciden qué cuenta como diferencia, y es lo primero que hay que mirar cuando salen diferencias que a usted no le parecen diferencias. Vienen puestas con lo razonable:

Opción	De fábrica
Ignorar el contador <code>AUTO_INCREMENT</code>	Activada: cambia con cada inserción.
Ignorar <code>DEFINER</code> y <code>SQL SECURITY</code>	Activada: casi siempre difieren sin significar nada.
Ignorar los anchos de enteros — <code>INT(11)</code> frente a <code>INT</code> —	Activada. <code>tinyint(1)</code> se conserva: es un booleano.
Ignorar comentarios	Desactivada.
Ignorar espacios y mayúsculas en los cuerpos	Activada.
Ignorar collation / mapear equivalencias entre motores	Según el caso; mire esta primero al comparar MySQL con MariaDB.
Ignorar el orden de las columnas	Desactivada.
Ignorar permisos	Activada: leer <code>mysql.*</code> exige permisos que la mayoría de las cuentas no tienen.

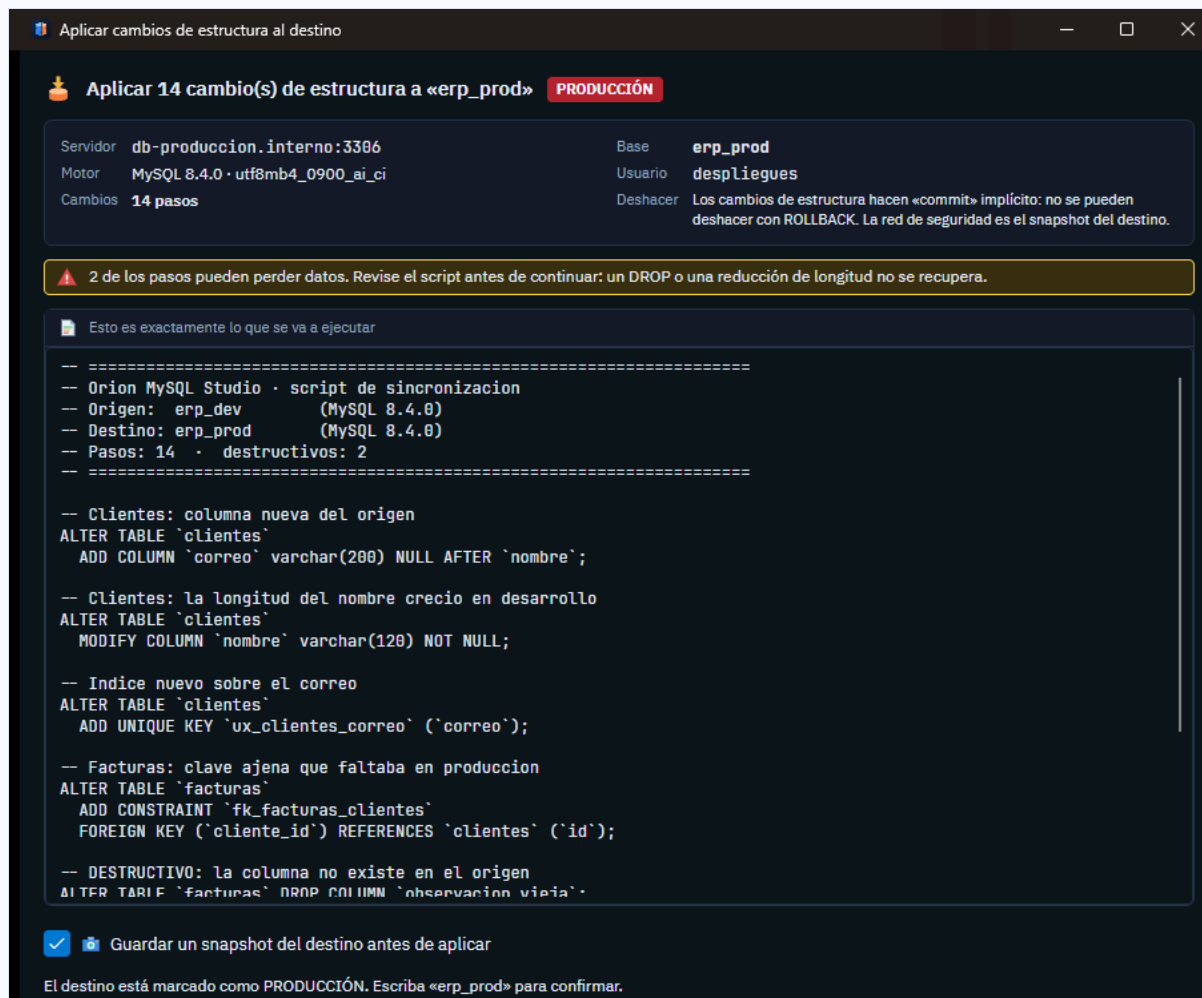
La caja de búsqueda del diálogo entra también en la **descripción** de cada opción, no solo en su nombre: quien busca «mayúsculas» no tiene por qué saber cómo se llama la opción.

Las opciones se guardan **con el proyecto**, no en el producto. Así dos comparaciones distintas pueden tener criterios distintos sin pisarse.

7. Aplicar al destino

Aplicar es el único momento en que el producto escribe en una base. Está rodeado de puertas a propósito.

CAPTURA 04



La confirmación: las señas del destino, el aviso de los pasos destructivos, el script completo, el snapshot previo y —por ser PRODUCCIÓN— la casilla donde hay que escribir el nombre de la base.

Qué pasa cuando pulsa «Aplicar al destino...»

1. Se abre la confirmación con **servidor**, **base**, **motor**, **usuario**, cuántos pasos son, cuántos pueden perder datos y el script entero.
2. Si marcó «**Guardar un snapshot del destino**» —viene marcado—, antes de tocar nada se guarda una foto del esquema actual en la biblioteca de snapshots.

3. Si el destino es **PRODUCCIÓN**, el botón nace deshabilitado y solo se activa al escribir el nombre exacto de la base.
4. El script se ejecuta **paso a paso**. Si uno falla, se detiene ahí y le dice en qué objeto.
5. Al terminar bien, el producto **vuelve a comparar solo**, para que vea el resultado real y no el de antes.

Los cambios de estructura no se pueden deshacer. En MySQL y MariaDB toda sentencia DDL hace «commit» implícito, así que un **ROLLBACK** no revierte un **ALTER TABLE**. La red de seguridad es el snapshot, no la transacción. Acéptelo cuando se lo ofrezca.

El script es editable

Lo que se ejecuta es **el texto que usted tiene delante**, no otro. Si lo edita en el visor, se aplica lo editado; y lo que parezca destructivo —un **DROP**, un **TRUNCATE**, un **DELETE**, un **REVOKE**— se cuenta como tal en la confirmación aunque lo haya escrito usted.

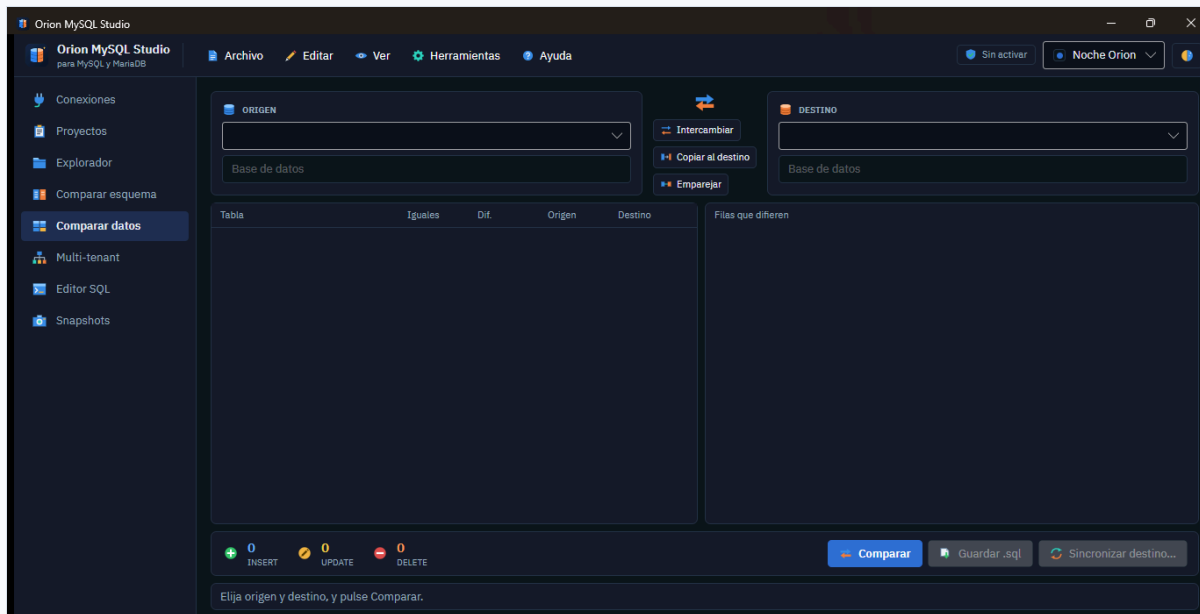
Si algo falla a mitad

- Lo aplicado **sigue aplicado**: no hay vuelta atrás automática.
- El mensaje dice el objeto y el error del servidor.
- Lo normal es arreglar la causa y volver a comparar: el script nuevo tendrá solo lo que falte.

8. Comparar datos

Compara las **filas** de las tablas, no su estructura. Sirve para igualar catálogos, parámetros y tablas de configuración entre dos entornos.

CAPTURA 05



Comparar datos: el progreso cancelable, el resumen tabla por tabla y los contadores de INSERT, UPDATE y DELETE.

Cómo empareja las filas

Por su **llave**: la clave primaria si la hay, y si no un índice único. Si una tabla no tiene ninguno de los dos, se omite y se dice por qué; puede elegir usted las columnas que hagan de llave en las opciones.

Paso a paso

1. Elija origen y destino, igual que en comparar esquema.
2. Pulse «**Comparar**». La barra de progreso dice en qué bloque va y se puede **cancelar** en cualquier momento.
3. Revise la lista: una fila por tabla, con cuántas filas son iguales, diferentes, solo del origen y solo del destino.
4. Seleccione una tabla para ver el detalle: cada fila que difiere, con sus celdas en formato **origen** → **destino**.
5. Abajo, los tres contadores dicen cuántos **INSERT**, **UPDATE** y **DELETE** saldrían.

6. «**Guardar .sql**» archiva el script; «**Sincronizar destino...**» lo aplica, con la misma confirmación del capítulo anterior.

Tolerancias: qué cuenta como «igual»

Una base migrada llena la pantalla de diferencias que no lo son. Las tolerancias —en **Herramientas > Opciones de comparación**, sección de datos— se aplican por proyecto y se pueden sobrescribir por tabla:

- Ignorar mayúsculas y minúsculas, también en las llaves.
- Recortar los espacios al final o al principio del texto.
- Tratar `NULL` y la cadena vacía como iguales.
- Normalizar los saltos de línea —el archivo que pasó por Windows—.
- Ignorar las fracciones de segundo en fechas y horas.
- Tolerancia numérica, para los decimales que difieren en el último dígito.

Normalizar al comparar es una cosa; cambiar los datos del cliente, otra. **El script escribe siempre los valores originales**, no los normalizados.

Emparejar tablas que se llaman distinto

El botón «**Emparejar**» propone parejas entre las tablas que solo están en un lado y se parecen a una del otro: `Clientes` con `clientes`, `orden-detalle` con `orden_detalle`. Lo que es idéntico salvo mayúsculas, espacios o guiones viene **marcado**; lo dudoso, desmarcado. Usted acepta o corrige, y lo aceptado se guarda como mapeo en el proyecto.

No confunde `factura1` con `factura2`: los números no se ignoran nunca al comparar nombres.

Tablas grandes

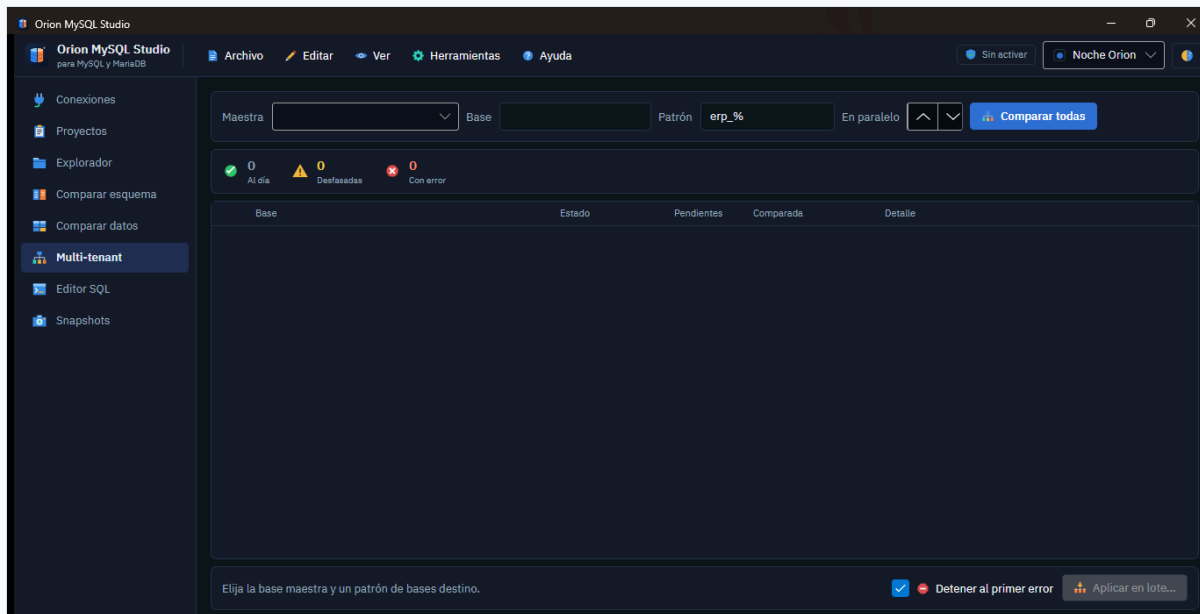
Primero se pide al servidor una **huella por bloques de filas** y se descartan los bloques idénticos sin bajar una sola fila; de los que difieren se leen los dos lados **en flujo** y se cruzan por la llave. Una tabla de un millón de filas se compara sin cargarla en memoria.

El script de datos sale **por lotes y dentro de una transacción**, respetando el orden de las claves ajenas. Si un paso falla, no se confirma nada.

9. Multi-tenant

Para quien tiene la misma base repetida —una por cliente— y necesita saber cuáles están al día. Compara una **base maestra**, o un snapshot, contra muchas a la vez.

CAPTURA 06



Multi-tenant: la maestra, la lista de bases destino y la matriz de resultados.

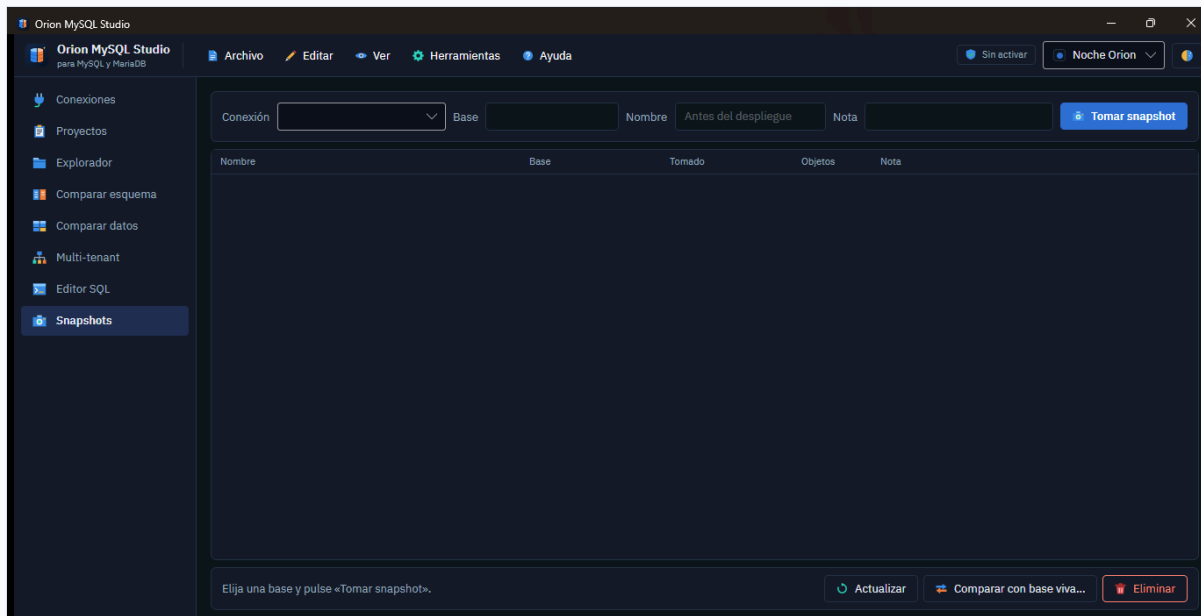
1. Elija la **maestra**: una conexión y una base, o un snapshot.
2. Elija las bases destino: márquelas de la lista o escriba un **patrón** como `erp_%`, que el producto resuelve contra el servidor.
3. Ajuste cuántas comparaciones corren **a la vez**. Más no siempre es más rápido: todas pegan al mismo servidor.
4. Pulse «**Comparar**» y espere a la matriz.

La matriz pone **delante lo que necesita atención**: primero las que fallaron, después las desfasadas y al final las que están al día. Desde ahí se generan los scripts por base y se pueden aplicar en lote, con registro por base y la opción de **detenerse al primer error**.

10. Snapshots

Un snapshot es el esquema completo de una base guardado en un archivo JSON. Sirve para comparar contra «cómo estaba antes» sin necesidad de tener aquella base viva.

CAPTURA 07



La biblioteca de snapshots, con su fecha, su motor y la nota de cada uno.

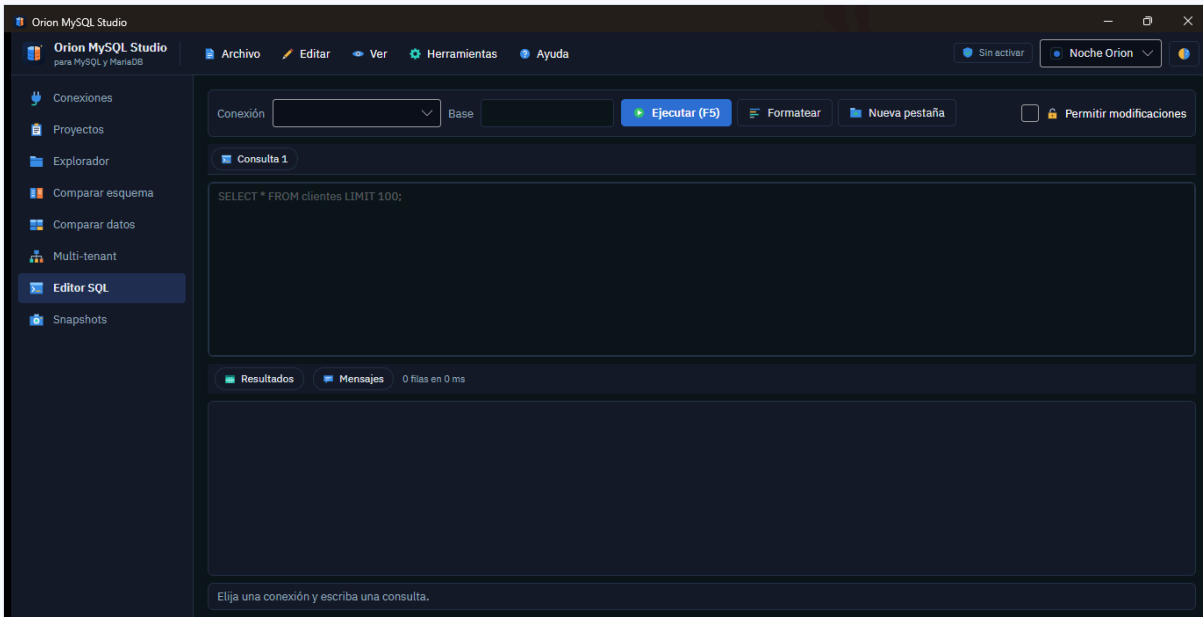
- **Capturar:** elija conexión y base, póngale una nota y guárdelo.
- **Comparar contra él:** un snapshot puede ser el origen o el destino de una comparación de esquema.
- **Volver a una versión:** comparar la base viva contra el snapshot genera el script que la devuelve a aquel estado.

El archivo se escribe **canonizado** —claves ordenadas y sin datos volátiles—, así que dos capturas de la misma base sin cambios dan archivos idénticos. Eso permite guardarlos junto al código y que el historial enseñe solo los cambios de verdad.

11. Editor SQL

Para lo que no es comparar: una consulta, un recuento, un ajuste puntual.

CAPTURA 08



El editor SQL: pestañas de consulta, la barra con «Ejecutar», «Formatear» y la casilla «Permitir modificaciones».

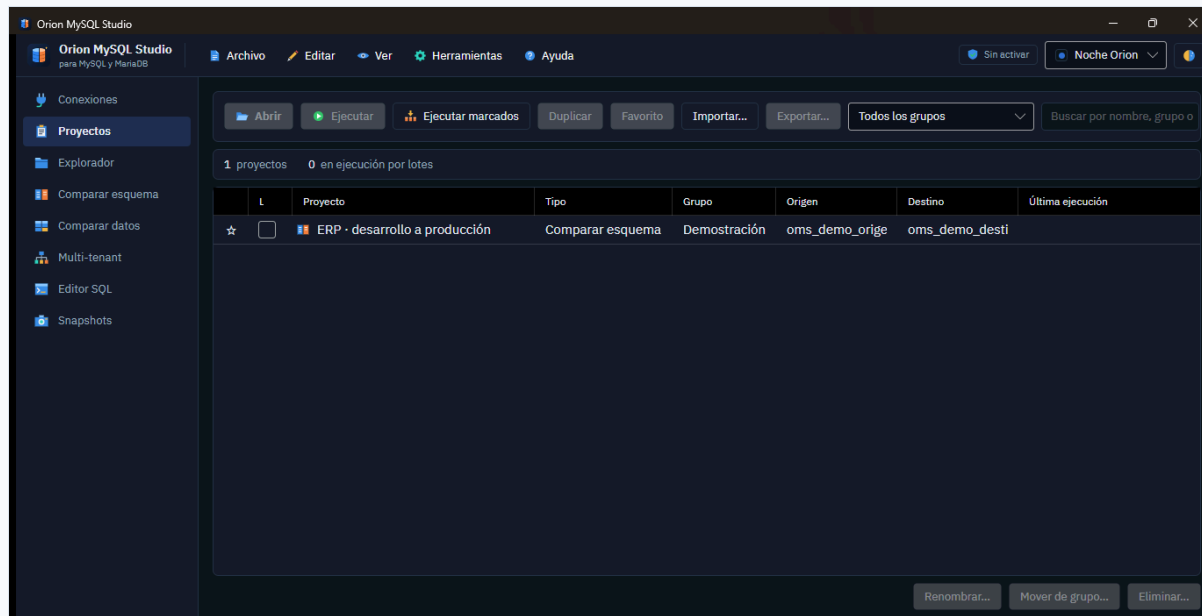
Ejecutar	F5 ejecuta todo lo de la pestaña; si hay texto seleccionado, ejecuta solo eso.
Varias sentencias	Se ejecutan en orden y cada una informa de sus filas afectadas y su tiempo.
DELIMITER	Soportado: puede pegar el cuerpo de una rutina con BEGIN ... END sin partirlo.
Resultados	Rejilla virtualizada —aguanta muchas filas— con exportación a CSV y Excel.
Mensajes	La otra pestaña: lo que devolvió el servidor, sentencia por sentencia.

En conexiones marcadas como **PRODUCCIÓN**, las sentencias que modifican están bloqueadas. Para ejecutarlas hay que marcar «Permitir modificaciones» expresamente, y el producto lo avisa antes.

12. Proyectos

Un proyecto guarda los dos lados, las opciones, los filtros, los mapeos y lo que debe pasar al terminar. Es lo que convierte «la comparación de cada lunes» en un clic.

CAPTURA 09



El administrador de proyectos: grupos, favoritos, la columna de ejecución en lote y el historial de la última ejecución.

Guardar y abrir

Archivo > **Guardar proyecto** escribe un `.omsproj` —JSON indentado, sin ninguna contraseña dentro— y lo registra en el administrador. **Archivo** > **Abrir proyecto** lo carga y lleva sus dos lados a la pantalla del tipo que sea.

La pantalla Proyectos

Acción	Qué hace
Abrir	Lleva el proyecto a su pantalla para mirarlo y decidir.
Ejecutar	Lo lanza tal como está guardado, sin abrirlo.
Ejecutar marcados	Lanza a la vez todos los marcados en la columna Lote .
Duplicar	Copia con nombre nuevo; puede cambiar el tipo y conservar las conexiones.
Favorito	Lo sube arriba del todo, por encima de los grupos.
Importar... / Exportar...	Traer el proyecto de un compañero, o pasárselo. El historial no viaja.
Renombrar... / Mover de grupo...	El nombre es con el que lo ejecuta la consola; el grupo es una etiqueta.
Eliminar...	Borra el proyecto y su archivo . Pregunta antes, diciendo el nombre.

La columna **Última ejecución** no dice solo cuándo: avisa cuando un proyecto **viene fallando varias veces seguidas**, que es lo que de verdad se quiere ver de un vistazo.

Varias comparaciones a la vez

Al lanzar proyectos aparece la **tira de ejecuciones** bajo la barra superior, con una pestaña por cada uno: su estado, su mensaje de progreso y su botón de cancelar, que cancela **solo esa**. De fábrica corren tres a la vez —cada comparación abre dos conexiones y lee catálogos enteros, así que lanzarlas todas no va más rápido— y en Android, una. «Limpiar» quita de la tira las que ya terminaron.

Lanzar no es aplicar. Ejecutar un proyecto compara y hace lo que el proyecto tenga configurado en sus eventos; para que llegue a tocar un destino hace falta haber activado «Permitir ejecución desatendida» en ese proyecto.

Perfiles de opciones

En el diálogo de opciones, el desplegable de arriba aplica un criterio completo de una vez. Vienen tres de fábrica —**Valores de fábrica**, **Entre motores (MySQL ↔ MariaDB)** y **Datos migrados (tolerante)**— y con «Guardar perfil...» añade los suyos, que quedan en la carpeta de datos. Los de fábrica no se pueden sobrescribir: son el suelo al que volver cuando ya no sabe qué tocó.

Aplicar un perfil **copia sus valores** al proyecto; no deja una referencia viva. Así, cambiar un perfil no altera en silencio el comportamiento de veinte proyectos guardados hace meses.

13. Reportes y eventos al terminar

Reportes HTML

Un archivo **autocontenido**: se abre en cualquier navegador, se envía por correo y no depende de ningún recurso externo. Hay reporte de comparación de esquema —con cuatro niveles de detalle—, de comparación de datos, de multi-tenant y de documentación completa de una base.

Los colores salen del tema activo, con una versión clara para imprimir, y el estado de cada objeto se escribe **con palabras** además del color.

Eventos al terminar

Un proyecto puede tener configurado qué hacer cuando acabe: mostrar el script, guardarlo, ejecutarlo, generar un reporte, enviar un correo, guardar un snapshot, ejecutar un `.sql`, encadenar otro proyecto, cerrar la pestaña o cerrar la aplicación.

Para que un proyecto pueda **ejecutar** cambios sin que nadie mire, hay que activar expresamente «Permitir ejecución desatendida». La confirmación no desaparece: **se adelanta**. Usted la da una vez, y queda registrada con fecha.

14. Línea de comandos

`orion-mysql` se instala junto a la aplicación y sirve para lo mismo, sin ventana: comparar de madrugada, desde una tarea programada o desde integración continua.

Comandos

<code>orion-mysql run <proyecto></code>	Ejecuta un proyecto con sus opciones y eventos
<code>orion-mysql list projects</code>	Enumera los proyectos del administrador
<code>orion-mysql list connections</code>	Enumera las conexiones guardadas
<code>orion-mysql encrypt-password</code>	Cifra una contraseña para la línea de comandos

El proyecto se nombra por su **nombre en el administrador** o por la ruta de un `.omsproj`. Un archivo que no esté registrado se trata como plantilla de solo lectura y acepta el reemplazo de conexión.

Opciones de `run`

Opción	Qué hace
<code>--all</code>	Ejecuta todos los proyectos marcados en el administrador.
<code>--log</code>	Escribe el log en la carpeta de registros aunque el proyecto no lo pida.
<code>--quiet</code>	Sin ventana ni detalle de progreso; solo el resultado.
<code>--deploy</code>	Al terminar, genera el script y lo aplica. Exige <code>--yes</code>.
<code>--yes</code>	Confirma el despliegue. Obligatoria junto con <code>--deploy</code> .
<code>--allow-production</code>	Permite desplegar contra una conexión marcada PRODUCCIÓN.
<code>--source-server, --source-database, ...</code>	Reemplazo de conexión del origen (solo plantillas).
<code>--target-server, --target-database, ...</code>	Lo mismo para el destino.

Códigos de salida

Código	Significa
0	Sin diferencias.
1	Hay diferencias.
2	Error de ejecución.
3	Error al aplicar el script. La comparación fue bien; falló el despliegue.

Con `--all`, el código es el **peor** de todos: si falla un proyecto de veinte, la tarea programada tiene que enterarse.

Contraseñas

`encrypt-password` lee la contraseña de la **entrada estándar** —no de un argumento, para que no quede en el historial ni en la lista de procesos— y devuelve una cadena cifrada con prefijo `!`, ligada a este usuario y a este equipo. Esa cadena es la que se pasa en `--source-password` o `--target-password`.

La consola **no aplica nada por su cuenta**. Sin `--deploy --yes` solo compara y genera; contra una conexión de producción hace falta además `--allow-production`.

15. Android

La aplicación de Android sirve para mirar y comparar desde el teléfono: conexiones, explorador, editor de consultas y las dos comparaciones. El resultado se puede **compartir como .sql** con el selector del sistema —correo, mensajería, la nube—.

No aplica cambios, y los botones de aplicar no aparecen. No están deshabilitados: no existen. Es una decisión de seguridad, no una limitación técnica.

Las comparaciones se ejecutan **de una en una**: dos comparaciones grandes a la vez agotan la memoria del teléfono y el sistema cierra la aplicación sin avisar.

16. Reglas de seguridad

Resumidas en un sitio, porque son lo que separa esta herramienta de un generador de SQL.

1. **Nada se ejecuta contra un destino** sin una pantalla que enseñe servidor, base, motor, usuario, cuántos cambios son y el script completo.
2. Las conexiones **PRODUCCIÓN** llevan etiqueta roja permanente y exigen escribir el nombre de la base para confirmar.
3. Los cambios **destructivos** se cuentan aparte y se marcan.
4. Antes de aplicar se ofrece **guardar un snapshot** del destino.
5. El script de **datos** va en transacción; el de **estructura** no puede ir, y el producto lo dice en lugar de fingir lo contrario.
6. **Android no aplica** cambios.
7. Las **contraseñas y llaves SSH** se guardan cifradas en el almacén del sistema y nunca aparecen en los registros.
8. Los valores van siempre como **parámetros** y los nombres de objeto, escapados y validados: no se concatena SQL con lo que usted escriba.

17. Problemas frecuentes

Salen diferencias que no son diferencias

Mire Herramientas > Opciones de comparación. Lo más habitual es el contador `AUTO_INCREMENT`, el `DEFINER` o la collation entre motores distintos. En comparación de datos, las **tolerancias**.

La comparación no trae usuarios ni permisos

La opción «Ignorar permisos» viene **activada**. Desactívela. Si aun así no aparecen, es que la cuenta no puede leer `mysql.*`: el producto lo avisa y sigue sin esa parte.

«Probar» falla pero el servidor responde desde otro cliente

Revise el **host del GRANT**: un usuario creado como `'app'@'localhost'` no entra desde otro equipo. El error 1130 dice exactamente eso.

El script se detuvo a mitad

Lo aplicado sigue aplicado. El mensaje dice el objeto y el error del servidor; arregle la causa y vuelva a comparar: el script nuevo traerá solo lo que falte.

Una tabla no se compara en datos

No tiene clave primaria ni índice único. Elija usted las columnas llave para esa tabla en las opciones de datos.

El script de datos es enorme y no se puede aplicar desde la ventana

Por encima de 128 MB el producto no lo carga en memoria: lo dice y le pide que lo guarde y lo aplique desde la consola, que lo ejecuta sin cargarlo entero.

La prueba no se concede

Necesita conexión a internet una sola vez. Si el servidor de licencias contesta que no, compruebe que el correo es válido y que no ha gastado ya los dos equipos de la licencia; **«Liberar este equipo»** devuelve el cupo.

18. Dónde guarda sus cosas

Todo vive en la carpeta de datos del usuario: `%APPDATA%\OrionMySQLStudio` en Windows y `~/.config/OrionMySQLStudio` en Linux.

Archivo o carpeta	Qué guarda
<code>conexiones.json</code>	Las conexiones, sin contraseñas .
<code>proyectos.json</code> y <code>proyectos\</code>	El índice del administrador y los <code>.omsproj</code> registrados.
<code>perfiles.json</code>	Los perfiles de opciones que haya creado.
<code>snapshots\</code>	La biblioteca de snapshots.
<code>registros\</code>	El registro de la aplicación, un archivo por día, catorce días.
Almacén del sistema	Las contraseñas y las llaves SSH, cifradas. No están en ningún archivo de los de arriba.

Para llevarse la configuración a otro equipo basta copiar esa carpeta, **salvo las contraseñas**: están cifradas contra el usuario y la máquina, así que hay que volver a escribirlas. Es a propósito.



ORION
SOFTWARE
CONSULTING

Orion MySQL Studio

User manual

1.2.0 · .NET 10 · Windows · Linux · Android

Contents

1. What it does and what you need
2. Installing and first run
3. Finding your way around the window
4. Connections
5. Browser
6. Compare schema
7. Applying to the target
8. Compare data
9. Multi-tenant
10. Snapshots
11. SQL editor
12. Projects
13. Reports and on-completion events
14. Command line
15. Android
16. Safety rules
17. Common problems
18. Where it keeps your things

1. What it does and what you need

Orion MySQL Studio compares two databases —MySQL or MariaDB—, shows you how they differ, and writes the SQL script that makes the second one match the first. You read it, edit it if you want, and decide whether it runs.

It compares two different things, and they are worth keeping apart: the **schema** —the structure: tables, columns, indexes, views, routines, privileges— and the **data** —the rows in each table—. Two screens, two scripts.

What it does not do, and you should know from the start

- It applies nothing without your confirmation, and the confirmation shows the server, the database, the user, and the full script first.
- It does not copy passwords. When synchronising user accounts, the `CREATE USER` comes out with a placeholder password you must change.
- It does not apply changes from Android. There you can compare and generate the script; applying it is desktop only. That is a security decision.
- It needs nothing else installed. No MySQL client, no drivers, no .NET.

What you need

On your machine	Windows 10 or 11 64-bit, or 64-bit Linux. Android 8 or later for the mobile app. About 250 MB of disk.
On the server	MySQL 5.7 or later, or MariaDB 10.3 or later, with its port reachable from your machine (3306 by default).
To compare	A user with <code>SELECT</code> on <code>information_schema</code> and on the databases you are going to compare.
To apply	Whatever each change requires: <code>ALTER</code> , <code>CREATE</code> , <code>DROP</code> , <code>INDEX</code> ... The product never asks for more than the script needs.
To compare accounts	<code>SELECT</code> on <code>mysql.*</code> . Without it the comparison carries on without that part and says so; it does not fail.

The product connects like any other MySQL client. It installs nothing on the server and does not ask you to enable an extension.

2. Installing and first run

Windows

Run `orion-mysql-studio-1.2.0-win-x64-setup.msi`. It installs the application and the `orion-mysql` command line, and creates the shortcuts. To remove it, use «Installed apps» in Windows.

If you would rather not install anything, download the portable `.zip` and unpack it wherever you like. Same two executables.

Until the packages are signed with a certificate, Windows will warn about an «unknown publisher». That is normal and does not indicate a problem.

Linux

```
tar -xzf orion-mysql-studio-1.2.0-linux-x64.tar.gz
cd "Orion MySQL Studio 1.2.0"
./OrionMySqlStudio
```

The `.tar.gz` already carries the execute bit. No root, no dependencies.

Android

Install the `.apk`. Android will ask for permission to install from this source. The app queries, browses, compares, and shares the `.sql`, but **never applies changes** to any database.

The free trial and the licence

The first time you open it, the top bar reads «**Not activated**». Click there —or **Help > Licence**— and fill in your name and e-mail to request the **30-day free trial**. Activation needs an internet connection once; after that it works offline.

The licence is **single and perpetual**, with every feature, and covers **two machines**; an Android phone counts as a machine. On the same screen there is «**Release this machine**», which returns the slot to the server if you are moving to another computer.

When the trial expires, the product goes **read-only**: you can browse, view DDL, and compare, but not generate or apply scripts. No connection and no project is lost.

3. Finding your way around the window

The window always has the same four pieces. It is worth recognising them before you start, because the rest of this manual names them without explaining them again.

Top bar	Logo, menu —File, Edit, View, Tools, Help—, licence status, and the theme selector.
Sidebar	The eight screens: Connections, Projects, Browser, Compare schema, Compare data, Multi-tenant, SQL editor, and Snapshots.
Run strip	Appears under the top bar only while projects are running , with one tab per project.
Status bar	At the bottom: the last message, the open project, and the active theme.

The menu, item by item

Item	What it does
File › New project	Clears source, target, and options. Asks first if a project is open. Ctrl+N
File › Open project	Loads an <code>.omspoj</code> and pours its two sides and its options into the matching screen. Ctrl+O
File › Save project	Saves what is on screen. The first time it asks where. Ctrl+S
Edit › Find	Moves the focus to the search box of the active screen. Ctrl+F
View › Skin	The five themes, each with its accent colour. Applied live and remembered.
View › Toggle theme	Switches between «Noche Orion» and «Claro».
Tools › Comparison options	Schema and data options, with a search box and profiles.
Help › Manual and help	Help for the screen you are on, with its section of the manual. F1
Help › Licence	Activate, revalidate, or release the machine.
Help › About	Version, third-party licences, and trademark notices.

The five themes

Noche Orion (the default), **Claro**, **Grafito**, **Terminal**, and **Alto contraste**. The change is immediate and is remembered. In all of them the state of a difference is also written in words: colour is never the only channel, so the product can be used without telling hues apart.

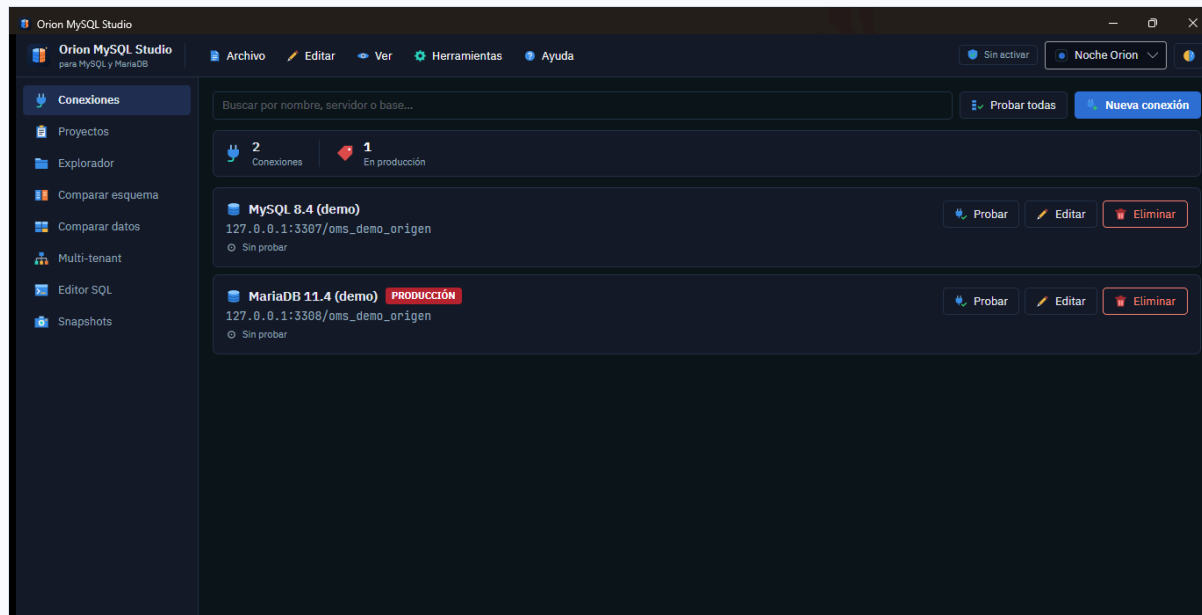
On a phone or a tablet

Below about 1000 pixels wide, the sidebar is replaced by a **bottom navigation bar** with the same screens; below 600, the menu and the status bar go away too. It is the same application, not a cut-down version.

4. Connections

A connection is a saved server. You create them once and use them from every screen.

FIGURE 01



The Connections screen: one card per server, the summary on top, and the PRODUCTION tag where it applies.

Creating a connection

1. Click «**New connection**».
2. Fill in name, server, port, user, and password. The **default database** is optional: it only decides what is proposed when you open the tree.
3. If the server requires encryption, pick the **TLS mode** and, if needed, the certificate authority file.
4. If the server is not exposed to the internet, turn on the **SSH tunnel** and set its host, user, and private key.
5. Tick **PRODUCTION** if the database must not break. See below what that changes.
6. Click «**Test**» and, if it answers, «**Save**».

What marking PRODUCTION means

- The connection carries a **permanent red tag** on every screen it appears on.
- To apply anything against it you must **type the database name** in the confirmation.

- In the SQL editor, statements that modify data or structure are **blocked** unless you tick «Allow modifications».

When «Test» fails

The message tells the cause apart instead of saying «could not connect». These are the common ones:

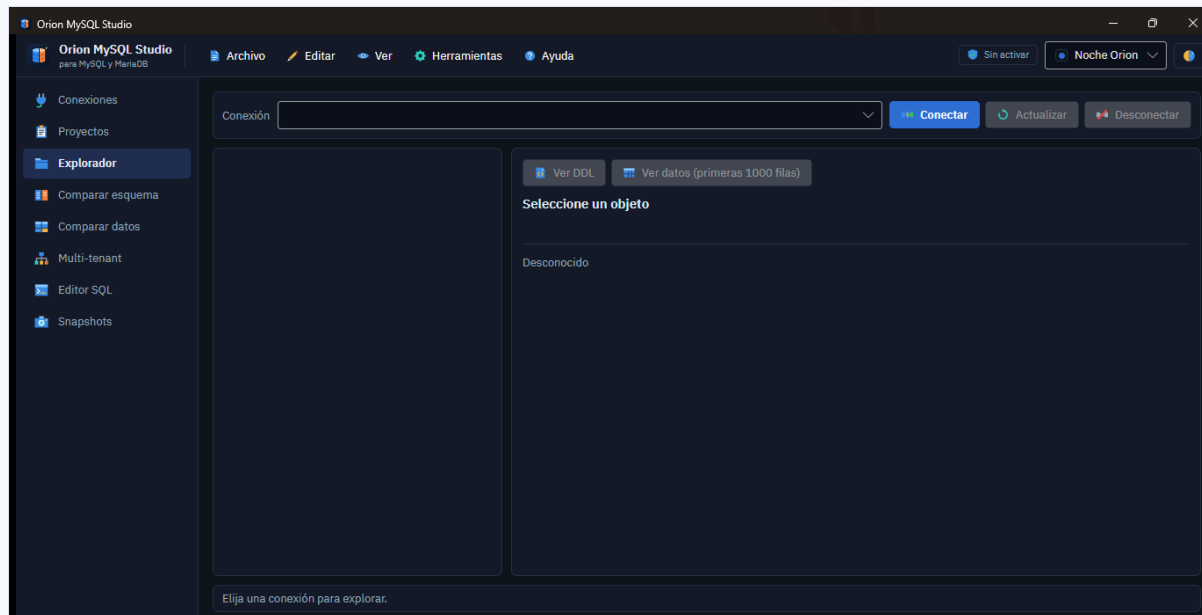
What it says	What it usually is
The server name does not resolve	A typo in the host, or a missing domain suffix.
The port is closed	The server is not listening there, or a firewall is in the way.
Wrong user or password (1045)	Credentials. Try the same user from another client.
The user has no access to that database (1044)	Credentials are fine; privileges on the database are missing.
The database does not exist (1049)	The default database name is misspelled.
This host is not allowed (1130)	The user's GRANT is tied to another host.
Timed out	Slow network, a VPN down, or the server not answering.

Passwords are **not stored in the connections file**. They go encrypted into the system secret store: DPAPI on Windows, a file with 600 permissions on Linux, and the key store on Android. Deleting a connection deletes its secrets too.

5. Browser

The browser walks a server without typing SQL. It is for looking before comparing.

FIGURE 02



The browser tree, the properties panel, and the data grid of the selected table.

1. Pick the connection in the drop-down and click «**Connect**».
2. Open the database node. The branches are Tables, Views, Procedures, Functions, Triggers, Events, and —on MariaDB— Sequences, each with its object count.
3. Selecting an object fills the panel on the right with its properties: engine, approximate rows, size, collation, and dates.

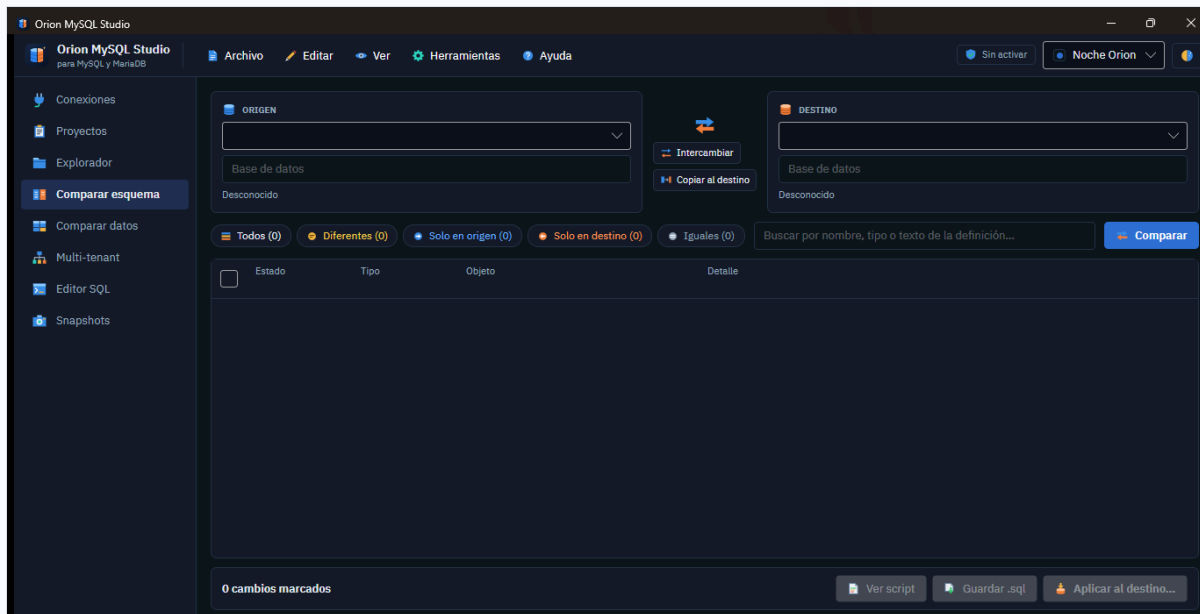
With an object selected, the two buttons on top do what gets used most: «**View DDL**» shows its `CREATE` exactly as the server returns it, and «**View data**» brings the **first 1000 rows**.

The tree loads **branch by branch**, as you open it. A database with a thousand tables shows up as fast as one with ten, because what you do not look at is not read.

6. Compare schema

This is the main screen. It compares the structure of two databases and writes the script that brings the target to the state of the source.

FIGURE 03



Compare schema: the source and target cards, the state filters with their counts, and the action bar.

Step by step

1. Pick **source** and **target**: the connection in the drop-down and the database name underneath. The source rules; the target is what changes.
2. Click «**Compare**». The product reads both catalogues and compares them.
3. Go through the list. Use the **pills** to filter by state and the search box to find an object by name, by type, or by **text in its definition**.
4. Select a row to see the **side-by-side diff**, source on the left and target on the right.
5. Tick or untick what you want to carry over. The header checkbox ticks and unticks **everything visible**, not everything.
6. Click «**View script**» to read it, «**Save .sql**» to file it, or «**Apply to target...**» to run it.

The four states

State	What it means	What the script does
Different	The object is on both sides and does not match.	<code>ALTER</code> with only what changed.
Only in source	Missing on the target.	Creates it.
Only in target	Left over on the target.	Drops it. Destructive.
Equal	They match. Shown so it is on the record.	Nothing.

Two icons sit next to the name when they apply: **destructive** —applying this may lose data— and **not compatible** —the object cannot be carried over to the target engine, and stays out of the script with its reason written down—.

Swap and copy

Between the two cards there are two buttons that save a lot of clicking. «**Swap**» exchanges source and target —and **discards the previous result** on purpose: leaving it on screen would show something that no longer matches the cards—. «**Copy to target**» puts the source connection and database into the target, which is where you start when comparing two databases on the same server.

MySQL against MariaDB

If the two sides are different engines, a yellow warning appears and the script is written in the **target** dialect. Whatever is not portable —native `JSON` versus `LONGTEXT` with `json_valid`, `CREATE SEQUENCE`, `WITH SYSTEM VERSIONING`, `INVISIBLE` columns— is flagged «Not compatible» and left out of the script.

The comparison options

Under **Tools > Comparison options**. They decide what counts as a difference, and they are the first thing to check when differences show up that do not look like differences to you. They ship sensibly set:

Option	Default
Ignore the <code>AUTO_INCREMENT</code> counter	On: it changes with every insert.
Ignore <code>DEFINER</code> and <code>SQL SECURITY</code>	On: they almost always differ without meaning anything.
Ignore integer display widths — <code>INT(11)</code> versus <code>INT</code> —	On. <code>tinyint(1)</code> is kept: it is a boolean.
Ignore comments	Off.
Ignore whitespace and keyword case in bodies	On.
Ignore collation / map equivalents across engines	Depends on the case; check this one first when comparing MySQL with MariaDB.
Ignore column order	Off.
Ignore privileges	On: reading <code>mysql.*</code> needs rights most accounts do not have.

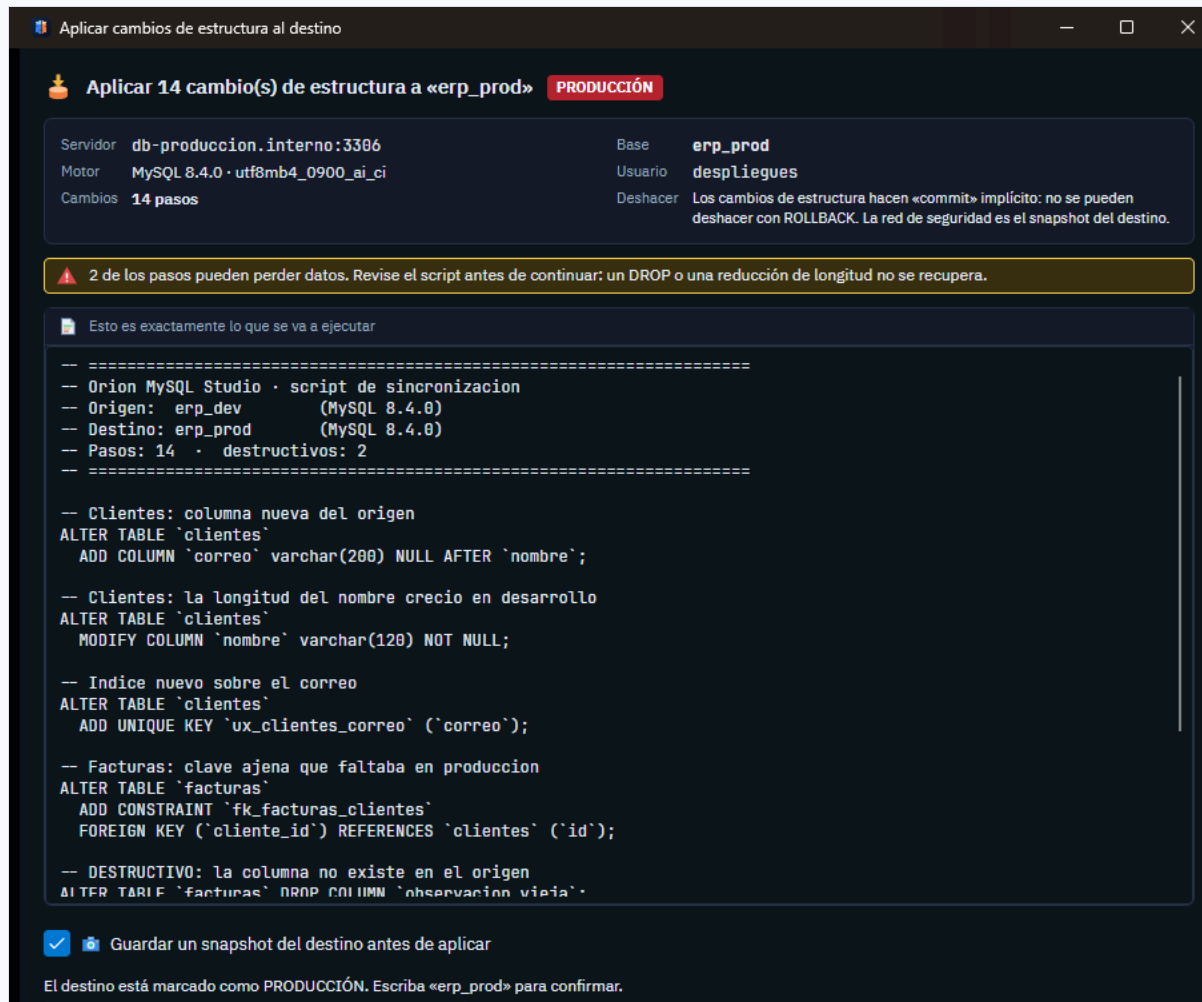
The search box in the dialog also looks inside each option's **description**, not just its name: someone searching for «case» does not have to know what the option is called.

Options are saved **with the project**, not with the product. That way two different comparisons can use different criteria without stepping on each other.

7. Applying to the target

Applying is the only moment the product writes to a database. It is surrounded by gates on purpose.

FIGURE 04



The confirmation: the target details, the warning about destructive steps, the full script, the snapshot offer, and —because it is PRODUCTION— the box where the database name must be typed.

What happens when you click «Apply to target...»

1. The confirmation opens with **server**, **database**, **engine**, **user**, how many steps there are, how many may lose data, and **the whole script**.
2. If «**Save a snapshot of the target**» is ticked —it is by default—, a picture of the current schema is saved to the snapshot library before anything is touched.
3. If the target is **PRODUCTION**, the button starts disabled and only becomes active when you type the exact database name.

4. The script runs **step by step**. If one fails, it stops there and tells you which object it was.
5. When it finishes cleanly, the product **compares again on its own**, so you see the real result and not the old one.

Structure changes cannot be undone. In MySQL and MariaDB every DDL statement commits implicitly, so a `ROLLBACK` does not revert an `ALTER TABLE`. The safety net is the snapshot, not the transaction. Accept it when it is offered.

The script is editable

What runs is **the text in front of you**, not another one. If you edit it in the viewer, the edited version is what gets applied; and whatever looks destructive —a `DROP`, a `TRUNCATE`, a `DELETE`, a `REVOKE`— is counted as such in the confirmation even if you typed it yourself.

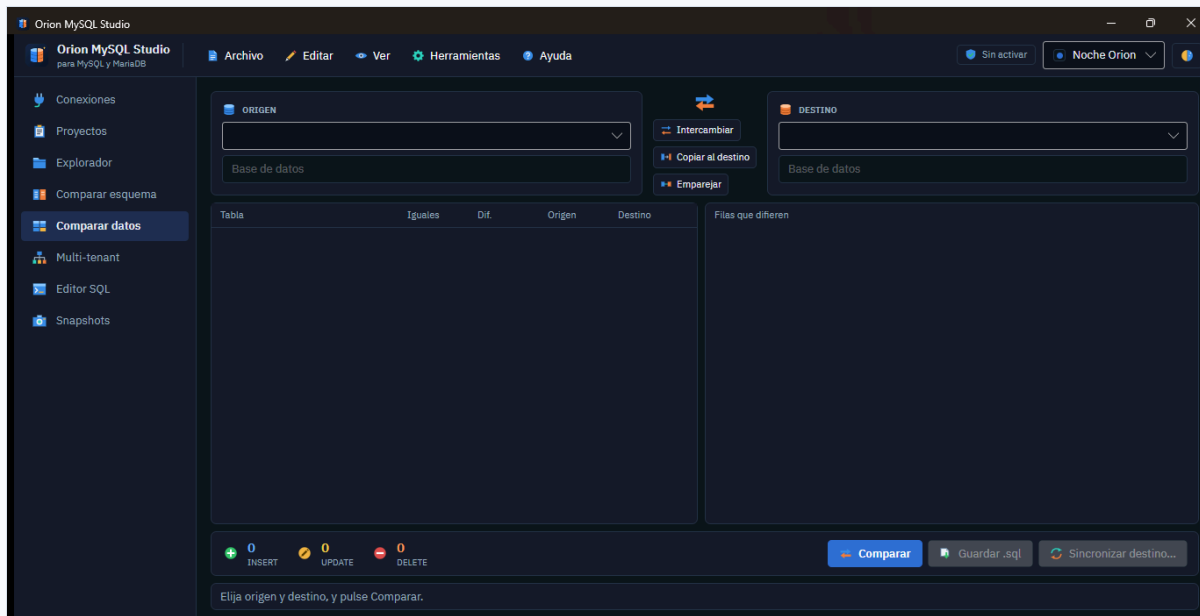
If something fails halfway

- What was applied **stays applied**: there is no automatic rollback.
- The message names the object and the server error.
- The usual move is to fix the cause and compare again: the new script will carry only what is still missing.

8. Compare data

This compares the **rows** of the tables, not their structure. It is for levelling catalogues, parameters, and configuration tables between two environments.

FIGURE 05



Compare data: the cancellable progress, the table-by-table summary, and the INSERT, UPDATE, and DELETE counters.

How rows are matched

By their **key**: the primary key if there is one, otherwise a unique index. A table with neither is skipped with the reason given; you can pick the key columns yourself in the options.

Step by step

1. Pick source and target, same as in compare schema.
2. Click «**Compare**». The progress bar says which block it is on and can be **cancelled** at any time.
3. Go through the list: one row per table, with how many rows are equal, different, only in the source, and only in the target.
4. Select a table to see the detail: every differing row, with its cells shown as **source** → **target**.
5. At the bottom, the three counters say how many **INSERT**, **UPDATE**, and **DELETE** statements would come out.

6. «Save .sql» files the script; «Synchronise target...» applies it, with the same confirmation as the previous chapter.

Tolerances: what counts as «equal»

A migrated database fills the screen with differences that are not differences. Tolerances —under **Tools > Comparison options**, data section— apply per project and can be overridden per table:

- Ignore letter case, in the keys too.
- Trim trailing or leading whitespace.
- Treat `NULL` and the empty string as equal.
- Normalise line endings —the file that went through Windows—.
- Ignore fractional seconds in dates and times.
- Numeric tolerance, for decimals that differ in the last digit.

Normalising for comparison is one thing; changing the customer's data is another. **The script always writes the original values**, never the normalised ones.

Matching tables with different names

The «Match» button proposes pairs among the tables that exist on only one side and resemble one on the other: `Clientes` with `clientes`, `orden-detalle` with `orden_detalle`. Anything identical but for case, spaces, or dashes comes **ticked**; anything doubtful comes unticked. You accept or correct, and what you accept is saved as a mapping in the project.

It never confuses `factura1` with `factura2`: digits are never ignored when comparing names.

Large tables

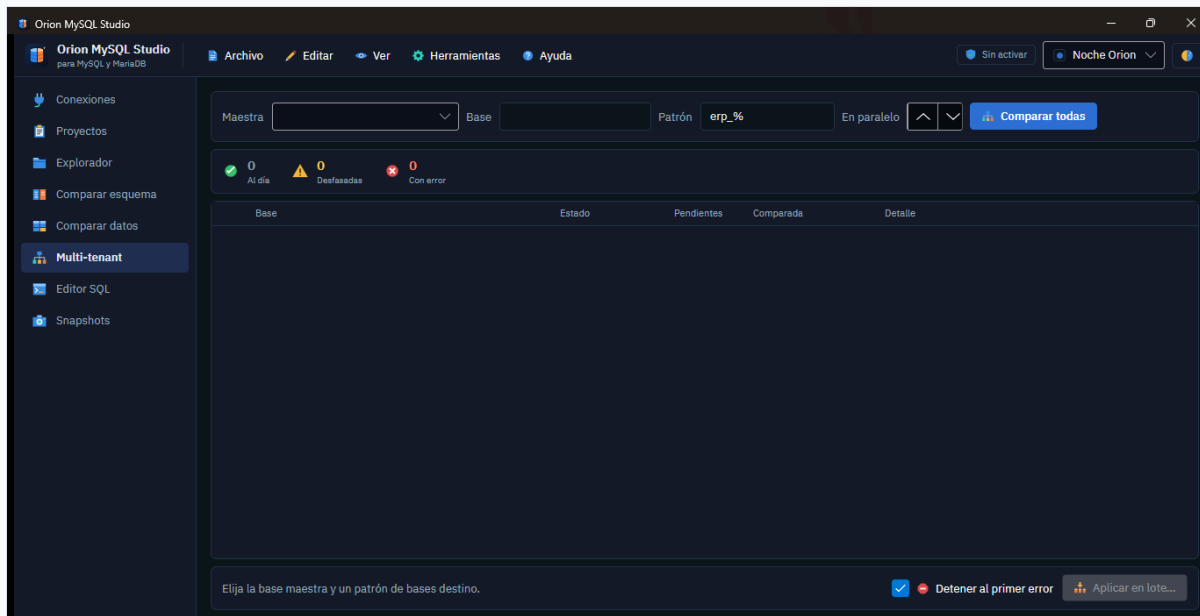
First the server is asked for a **per-block hash of rows**, and identical blocks are discarded without pulling a single row; for the blocks that differ, both sides are read **streamed** and merged by key. A million-row table is compared without loading it into memory.

The data script comes out **batched and inside a transaction**, respecting foreign key order. If a step fails, nothing is committed.

9. Multi-tenant

For whoever has the same database repeated —one per customer— and needs to know which ones are up to date. It compares one **master database**, or a snapshot, against many at once.

FIGURE 06



Multi-tenant: the master, the list of target databases, and the result matrix.

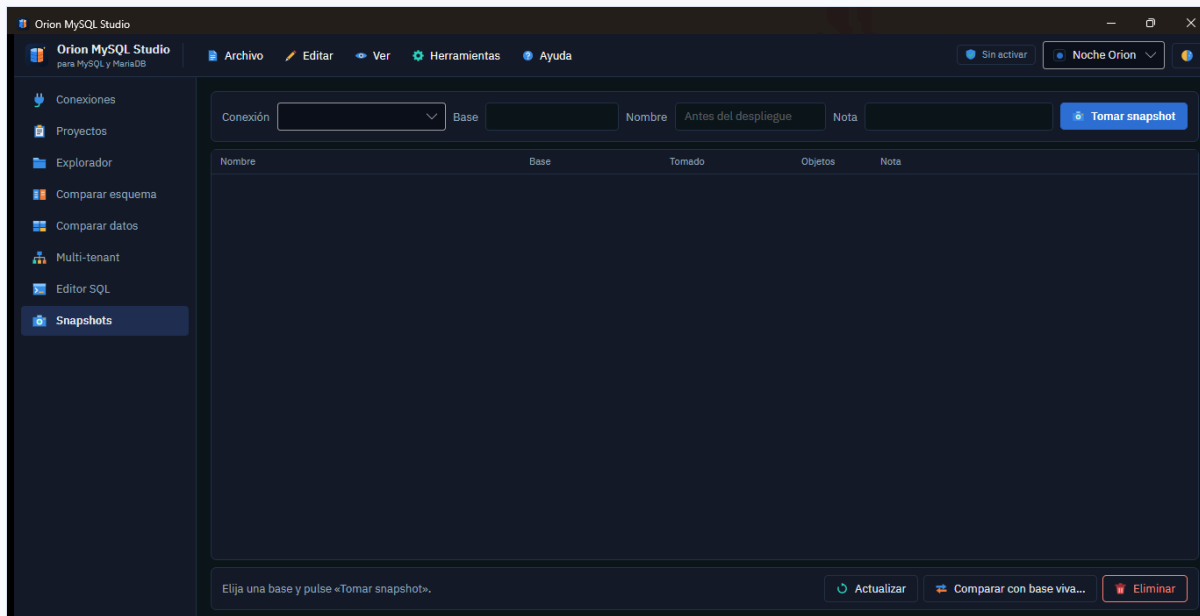
1. Pick the **master**: a connection and a database, or a snapshot.
2. Pick the targets: tick them from the list or type a **pattern** such as `erp_%`, which the product resolves against the server.
3. Set how many comparisons run **at once**. More is not always faster: they all hit the same server.
4. Click «**Compare**» and wait for the matrix.

The matrix puts **what needs attention first**: the ones that failed, then the ones out of date, and finally the ones already current. From there the per-database scripts can be generated and applied in batch, with a log per database and the option to **stop at the first error**.

10. Snapshots

A snapshot is the full schema of a database saved to a JSON file. It lets you compare against «how it used to be» without keeping that database alive.

FIGURE 07



The snapshot library, with the date, engine, and note of each one.

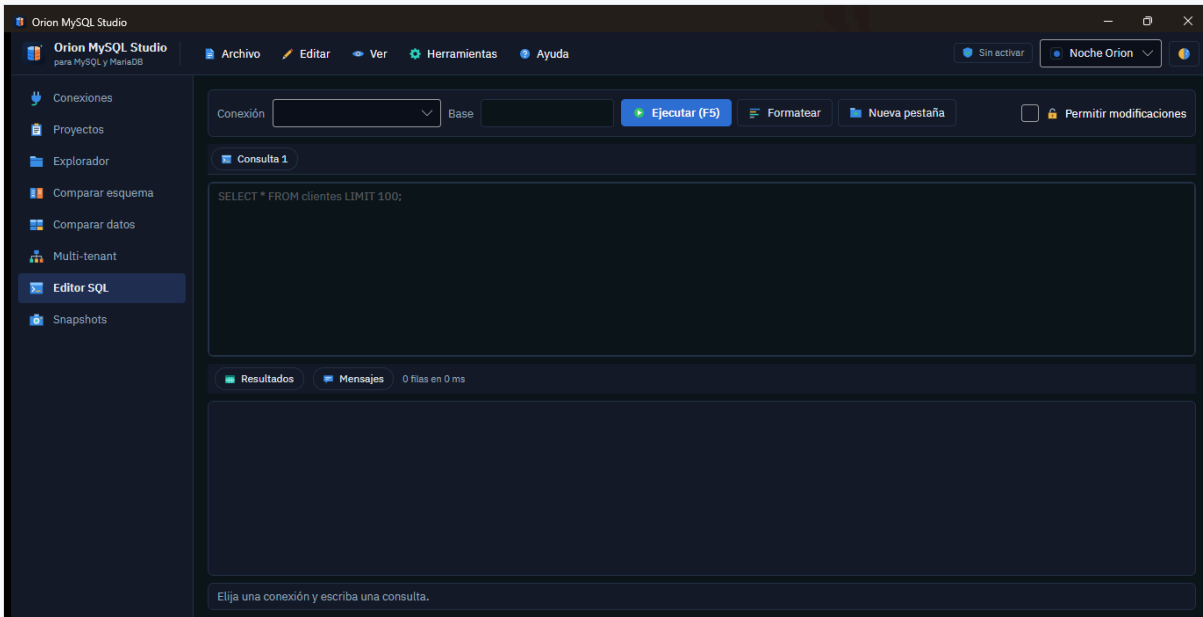
- **Capture:** pick a connection and a database, add a note, and save it.
- **Compare against it:** a snapshot can be the source **or** the target of a schema comparison.
- **Go back to a version:** comparing the live database against the snapshot produces the script that returns it to that state.

The file is written **canonicalised** —sorted keys, no volatile data—, so two captures of an unchanged database produce identical files. That makes them worth keeping next to your code, where the history shows only real changes.

11. SQL editor

For what is not comparing: a query, a count, a one-off fix.

FIGURE 08



The SQL editor: query tabs, the bar with «Run» and «Format», and the «Allow modifications» checkbox.

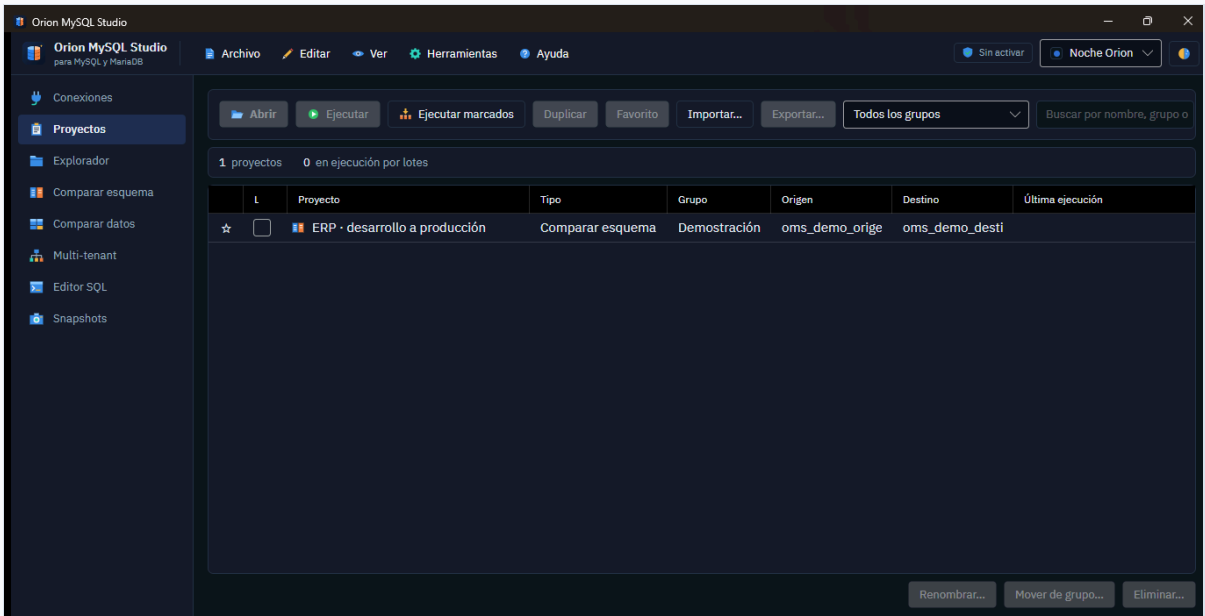
Run	F5 runs everything in the tab; if text is selected, it runs only that.
Several statements	They run in order, each reporting its affected rows and its time.
DELIMITER	Supported: you can paste a routine body with BEGIN ... END without splitting it.
Results	A virtualised grid —it holds many rows— with export to CSV and Excel.
Messages	The other tab: what the server returned, statement by statement.

On connections marked **PRODUCTION**, modifying statements are blocked. To run them you have to tick «**Allow modifications**» explicitly, and the product warns you first.

12. Projects

A project saves both sides, the options, the filters, the mappings, and what should happen when it finishes. It is what turns «the Monday comparison» into one click.

FIGURE 09



The project manager: groups, favourites, the batch column, and the history of the last run.

Saving and opening

File > Save project writes an `.omsproj` —indented JSON, with no password inside— and registers it in the manager. File > Open project loads it and takes its two sides to the screen for its type.

The Projects screen

Action	What it does
Open	Takes the project to its screen so you can look and decide.
Run	Launches it as saved, without opening it.
Run ticked	Launches at once everything ticked in the Batch column.
Duplicate	A copy with a new name; you can change its type and keep the connections.
Favourite	Moves it to the very top, above the groups.
Import... / Export...	Bring in a colleague's project, or pass yours on. History does not travel.
Rename... / Move to group...	The name is what the command line uses; the group is just a label.
Delete...	Removes the project and its file . It asks first, naming it.

The **Last run** column does not only say when: it flags a project that has been **failing several times in a row**, which is what you actually want to see at a glance.

Several comparisons at once

When you launch projects, the **run strip** appears under the top bar with one tab each: its state, its progress message, and its cancel button, which cancels **only that one**. Three run at a time by default—each comparison opens two connections and reads whole catalogues, so launching them all is not faster—and one on Android. «Clear» removes the finished ones from the strip.

Launching is not applying. Running a project compares and does whatever the project has configured in its events; for it to touch a target, «Allow unattended execution» must have been turned on for that project.

Option profiles

In the options dialog, the drop-down at the top applies a whole set of criteria at once. Three ship with the product—**Factory values**, **Across engines (MySQL ↔ MariaDB)**, and **Migrated data (tolerant)**—and «**Save profile...**» adds your own, which live in the data folder. The factory ones cannot be overwritten: they are the floor to come back to when you no longer know what you changed.

Applying a profile **copies its values** into the project; it does not leave a live reference. That way changing a profile does not silently alter the behaviour of twenty projects saved months ago.

13. Reports and on-completion events

HTML reports

A **self-contained** file: it opens in any browser, travels by e-mail, and depends on no external resource. There is a schema comparison report —with four levels of detail—, a data comparison report, a multi-tenant one, and full documentation of a database.

Colours come from the active theme, with a light version for printing, and each object's state is written **in words** as well as in colour.

On-completion events

A project can be told what to do when it finishes: show the script, save it, run it, generate a report, send an e-mail, save a snapshot, run a `.sql`, chain another project, close the tab, or close the application.

For a project to **run** changes with nobody watching, «Allow unattended execution» has to be turned on explicitly. The confirmation does not disappear: **it moves earlier**. You give it once, and it is recorded with a date.

14. Command line

`orion-mysql` installs next to the application and does the same job without a window: comparing overnight, from a scheduled task, or from continuous integration.

Commands

<code>orion-mysql run <project></code>	Runs a project with its options and events
<code>orion-mysql list projects</code>	Lists the projects in the manager
<code>orion-mysql list connections</code>	Lists the saved connections
<code>orion-mysql encrypt-password</code>	Encrypts a password for the command line

The project is named either by its **name in the manager** or by the **path of an `.omsproj`**. A file that is not registered is treated as a read-only template and accepts connection overrides.

run options

Option	What it does
<code>--all</code>	Runs every project ticked in the manager.
<code>--log</code>	Writes the log to the logs folder even if the project does not ask for it.
<code>--quiet</code>	No window and no progress detail; just the result.
<code>--deploy</code>	When the comparison ends, generates the script and applies it. Requires <code>--yes</code> .
<code>--yes</code>	Confirms the deployment. Mandatory together with <code>--deploy</code> .
<code>--allow-production</code>	Allows deploying against a connection marked PRODUCTION.
<code>--source-server</code> , <code>--source-database</code> , ...	Source connection override (templates only).
<code>--target-server</code> , <code>--target-database</code> , ...	The same for the target.

Exit codes

Code	Meaning
0	No differences.
1	There are differences.
2	Execution error.
3	Error applying the script. The comparison was fine; the deployment failed.

With `--all`, the code is the **worst** of them all: if one project out of twenty fails, the scheduled task has to find out.

Passwords

`encrypt-password` reads the password from **standard input** —not from an argument, so it does not end up in the shell history or the process list— and returns an encrypted string prefixed with `!`, tied to this user and this machine. That string is what you pass in `--source-password` or `--target-password`.

The command line **applies nothing on its own**. Without `--deploy --yes` it only compares and generates; against a production connection it also needs `--allow-production`.

15. Android

The Android app is for looking and comparing from your phone: connections, browser, query editor, and both comparisons. The result can be **shared as a .sql** through the system picker —mail, messaging, the cloud—.

It does not apply changes, and the apply buttons are not there. They are not disabled: they do not exist. A security decision, not a technical limitation.

Comparisons run **one at a time**: two large comparisons at once exhaust the phone's memory and the system kills the app without warning.

16. Safety rules

Gathered in one place, because they are what separates this tool from a SQL generator.

1. **Nothing runs against a target** without a screen showing the server, the database, the engine, the user, how many changes there are, and the full script.
2. **PRODUCTION** connections carry a permanent red tag and require typing the database name to confirm.
3. **Destructive** changes are counted separately and flagged.
4. Before applying, the product offers to **save a snapshot** of the target.
5. The **data** script runs inside a transaction; the **structure** one cannot, and the product says so instead of pretending otherwise.
6. **Android does not apply** changes.
7. **Passwords and SSH keys** are stored encrypted in the system store and never appear in the logs.
8. Values always travel as **parameters** and object names are escaped and validated: no SQL is concatenated from what you type.

17. Common problems

Differences show up that are not differences

Check **Tools > Comparison options**. The usual suspects are the `AUTO_INCREMENT` counter, the `DEFINER`, or the collation between different engines. For data comparison, the **tolerances**.

The comparison brings no users or privileges

The «Ignore privileges» option is **on** by default. Turn it off. If they still do not appear, the account cannot read `mysql.*`: the product says so and carries on without that part.

«Test» fails but the server answers from another client

Check the **host in the GRANT**: a user created as `'app'@'localhost'` cannot come in from another machine. Error 1130 says exactly that.

The script stopped halfway

What was applied stays applied. The message names the object and the server error; fix the cause and compare again: the new script will carry only what is still missing.

A table is not compared in data mode

It has neither a primary key nor a unique index. Pick the key columns for that table yourself in the data options.

The data script is huge and cannot be applied from the window

Above 128 MB the product does not load it into memory: it says so and asks you to save it and apply it from the command line, which runs it without loading it whole.

The trial is not granted

It needs an internet connection once. If the licence server says no, check that the e-mail is valid and that you have not already used the two machines on the licence; «**Release this machine**» returns the slot.

18. Where it keeps your things

Everything lives in the user data folder: `%APPDATA%\OrionMySQLStudio` on Windows and `~/.config/OrionMySQLStudio` on Linux.

File or folder	What it holds
<code>conexiones.json</code>	The connections, without passwords .
<code>proyectos.json</code> and <code>proyectos\</code>	The manager index and the registered <code>.omsproj</code> files.
<code>perfiles.json</code>	The option profiles you have created.
<code>snapshots\</code>	The snapshot library.
<code>registros\</code>	The application log, one file per day, fourteen days.
System store	Passwords and SSH keys, encrypted. They are in none of the files above.

To move your setup to another machine, copying that folder is enough **except for the passwords**: they are encrypted against the user and the machine, so they have to be typed again. That is on purpose.