DB CONSOLE

CONSOLA DE BASES DE DATOS

Tus bases de datos: exploradas, consultadas y cambiadas solo cuando tú lo dices, con una copia de seguridad debajo.

CÓMO FUNCIONA

MANUAL DE USUARIO · OCTUBRE DE 2026

777 SOFTWARE

777 ADMIN TOOLS

1 — QUÉ ES DB CONSOLE

Una base de datos es donde un programa guarda sus hechos: los usuarios, los pedidos, los ajustes, el historial. Una consola de administración de bases de datos es el banco de trabajo para mirar dentro de una: ver sus tablas, leer sus filas, hacerle preguntas en su propio idioma (SQL) y, cuando el administrador lo decide, cambiar lo que hay allí.

DB Console es ese banco de trabajo como un programa que vive en tu propia computadora. Empezó como una pequeña herramienta construida para el trabajo de administración de esta misma casa; la versión independiente conserva exactamente el alma de aquella herramienta — encuentra tus bases de datos por ti, hace copia de seguridad antes de escribir y bloquea todo cambio hasta que el administrador lo dice — y la hace crecer hasta un programa de escritorio que cualquier administrador puede apuntar a sus propias bases de datos SQLite y PostgreSQL.

Toda la personalidad de DB Console es proteger al administrador del dedo gordo. Toda herramienta de bases de datos del planeta ejecutará con gusto la peor consulta de tu vida a las 3 de la mañana. Esta trata tu propio peor momento como un caso de diseño: empieza en solo lectura, te hace revisar cada cambio antes de que se ejecute, copia la base de datos antes de que aterrice cualquier escritura y hace que las sentencias verdaderamente peligrosas sean difíciles de disparar por accidente. Las secciones 5–7 son ese sistema, pieza por pieza.

Lo que no es: no es un servidor, y no es un servicio que corre cuando no estás mirando. Se abre cuando tú la abres, se conecta solo adonde tú la apuntas, y cuando la cierras, está cerrada. No hay cuentas, nada informa a nadie, y el programa sale sabiendo hablar con bases de datos, nunca dónde viven las tuyas (§10).

2 — CONECTAR CON TUS BASES DE DATOS

DB Console habla hoy con dos tipos de base de datos. Se conectan de manera distinta, porque viven de manera distinta:

- **SQLite — una base de datos que es un archivo** — Una base de datos SQLite es un solo archivo en un disco. Abre una directamente, o deja que la consola las encuentre por ti: tú

apruebas las carpetas en las que puede mirar, ella revisa la cabecera de identidad de cada archivo candidato para asegurarse de que de verdad es una base de datos SQLite, y enumera cada una que encuentra — con su tamaño y su conteo de tablas — una sola vez, aunque un archivo aparezca en dos sitios. Mira solo en las carpetas que aprobaste. No hay ninguna lista de la casa integrada: el programa llega sin conocer ninguna carpeta.

- **PostgreSQL — una base de datos que es un servidor** — Una base de datos PostgreSQL vive en un servidor, así que rellenas un formulario breve: la dirección del servidor, su puerto, el nombre de la base de datos y tu nombre de usuario; una dirección como db.example.local. El campo de dirección sale vacío: la primera dirección que contenga será una que tú escribiste. La conexión que hace es real, con el servidor que nombraste: la consola nunca finge estar conectada a algo que no ha alcanzado de verdad.

Conexiones guardadas

- Las conexiones que conservas se guardan por nombre, en tus propias palabras: «Shop Postgres», «Laptop data store» («Almacén de datos de la portátil»). Una conexión guardada recuerda la dirección, el puerto, el nombre de usuario y las rutas de archivos: las direcciones, no las llaves.
- Las contraseñas nunca se escriben en esa lista guardada. El propio código que guarda elimina todo lo que tenga forma de contraseña antes de que la lista toque el disco. Las contraseñas viven en un solo lugar: la bóveda cifrada (§8).
- Toda conexión nueva empieza en solo lectura (§5), incluidas las que guardaste hace mucho y simplemente no has cambiado. Leer siempre está permitido; cambiar algo requiere una decisión que tomas a propósito.

3 — MIRAR ALREDEDOR: ÁRBOL, CUADRÍCULA, ESTRUCTURA

Una vez conectado, la ventana te da tres maneras de mirar la misma base de datos. Juntas responden a «¿qué hay aquí, y qué aspecto tiene?»

- **The schema tree** («El árbol de esquema») — La base de datos como un esquema: sus tablas y vistas, cada una abriéndose para mostrar sus columnas. Este es el mapa: qué hay aquí, antes de tocar nada.
- **The data grid** («La cuadrícula de datos») — Las filas de una tabla, 50 por página, con controles de página. Haz clic en el encabezado de una columna para ordenar por esa columna. Un cuadro de búsqueda caza tus palabras en todas las columnas a la vez. Las celdas vacías se muestran como NULL: la palabra de la base de datos para «nada guardado aquí», y los datos binarios se muestran como una etiqueta sencilla como [blob 12B]: mostrados como lo que son, nunca adivinados, nunca un fallo.
- **The structure view** («La vista de estructura») — Una tabla desarmada: cada columna con su tipo y su valor por defecto, la clave primaria, las claves foráneas que la enlazan con otras tablas (en ambas direcciones), sus índices y la sentencia CREATE real de la tabla: las palabras exactas que la construyeron, legibles y copiables.

Mirar no cambia nada. Puedes explorar una base de datos de producción todo el día en esta ventana y la base de datos nunca sabrá que estuviste allí: eso es lo que significa solo lectura, y es el estado en el que empiezas (§5).

4 — EL EDITOR SQL: PREGÚNTALE TÚ MISMO

El editor es donde escribes SQL — el propio idioma de la base de datos — cuando el árbol y la cuadrícula no son la pregunta que necesitas responder. Escribe una sentencia, ejecuta todo o solo la parte que has seleccionado, y la respuesta aterriza en una cuadrícula de resultados bajo el editor, en la misma tabla limpia que el explorador.

- **El límite de 500 filas, dicho en voz alta.** Una pregunta puede coincidir con un millón de filas; la cuadrícula muestra las primeras 500 y lo dice. Una respuesta limitada se etiqueta como limitada: nunca recortada en silencio y presentada como toda la verdad.
- **Llévate la respuesta contigo.** Los resultados se exportan a un archivo CSV o JSON en tu propio disco: tuyos, en formatos abiertos, para abrirlos en cualquier cosa.
- **Las sentencias de lectura se ejecutan.** Una pregunta (un SELECT) simplemente se ejecuta: leer es para lo que está la consola.
- **Las sentencias de cambio no se ejecutan: todavía no.** Si tu sentencia cambiaría algo (INSERT, UPDATE, DELETE, DROP y su familia), no se ejecuta cuando pulsas ejecutar. Va a la bandeja de cambios pendientes (§6), y el sistema de seguridad se encarga desde ahí. Ese desvío no es una negativa; es el producto funcionando como fue diseñado.

Una frase para recordar: en DB Console, ejecutar un cambio y aterrizar un cambio son dos actos distintos. Puedes escribir lo que sea. Nada aterriza hasta que ha pasado la bandeja, las advertencias y tu Commit («Confirmar»).

5 — SOLO LECTURA POR DEFECTO: EL MARCO DORADO

Toda conexión — un archivo que acabas de abrir, un servidor que acabas de guardar, uno que guardaste hace un año — empieza en solo lectura, y se queda así hasta que tú personalmente cambies esa conexión a escribible. El cambio es por conexión y se recuerda: tu base de datos de pruebas puede vivir escribible mientras la de producción nunca sale del dorado.

Y siempre puedes ver en qué mundo estás. El propio marco de la ventana lleva el modo, en color, en todo momento:

- **GOLD — READ-ONLY** («DORADO — SOLO LECTURA») — Puedes mirarlo todo y no cambiar nada. Las sentencias de cambio no tienen dónde aterrizar: ni siquiera pueden entrar en la bandeja. Este es el estado inicial de toda conexión.
- **RED — WRITABLE** («ROJO — ESCRIBIBLE») — Tú cambiaste esta conexión a propósito. Los cambios son posibles: a través de la bandeja, la copia de seguridad y las guardas (§6–§7), nunca rodeándolas. El borde rojo está ahí para que veas tu nivel de peligro desde el otro lado de la habitación.

¿Por qué dorado por defecto? Porque la mayor parte del trabajo de administración es leer: revisar un valor, cazar una fila, responder una pregunta, y la única vez entre cien en que estás cambiando algo es exactamente cuando deberías tener que decirlo en voz alta primero. Solo lectura por defecto significa que el accidente tiene que ser deliberado antes de poder ser real.

6 — LA BANDEJA DE CAMBIOS PENDIENTES: REVISAR ANTES DE QUE SE EJECUTE

En una conexión escribible, una sentencia de cambio empieza un viaje con cuatro paradas, y puede desecharse en cualquiera de ellas:

- 1. **Se escribe.** Escribes la sentencia en el editor y la ejecutas. No le ha pasado nada a la base de datos.
- 2. **Espera en la bandeja.** La sentencia se reúne en la bandeja de cambios pendientes, que te muestra — en SQL sencillo, exactamente como se ejecutará — cada sentencia a punto de ejecutarse. Un contador en el marco de la ventana mantiene el conteo frente a ti: «3 changes pending» («3 cambios pendientes»).
- 3. **Recibe advertencia, si merece una advertencia.** Algunas sentencias activan las guardas (§7) y se detienen aquí para una conversación en palabras claras antes de poder seguir.
- 4. **Tú decides. Commit** («Confirmar») ejecuta toda la bandeja, después de que la copia de seguridad (§7) se haya hecho con seguridad. **Discard** («Descartar») la tira, y la base de datos nunca se entera.

La bandeja es la diferencia entre una consola que hace lo que escribiste y una que hace lo que quisiste decir. Una errata en una sentencia es una curiosidad mientras está en la bandeja, legible a tu ritmo. La misma errata, ejecutada al clic, es un incidente. DB Console mantiene todo cambio como una curiosidad hasta que tú personalmente lo promueves.

Cierra la ventana con cambios pendientes y la consola te pregunta, por nombre, qué hacer con ellos. Nada pendiente se evapora jamás en silencio, y nada pendiente se cuela jamás tampoco.

7 — COPIA DE SEGURIDAD ANTES DE ESCRIBIR, Y LAS GUARDAS

La copia de seguridad

Cuando pulsas **Commit** («Confirmar»), el primer acto de la consola no es el cambio: es la copia. Para una base de datos SQLite, el archivo entero se copia a una copia de seguridad con marca de tiempo (el mismo nombre, con .bak y la hora añadidos), y solo entonces se ejecuta la bandeja. Se conservan las diez copias más recientes. Dos detalles importan:

- La copia se hace antes de que nada se ejecute: así, si una escritura falla luego a mitad de camino, la copia sigue existiendo. La red se teje antes del salto, no después de la caída.
- Una mala escritura es por tanto siempre deshacible: devuelve el archivo de copia al lugar de la base de datos y estarás donde estabas antes del cambio: cada fila, cada tabla.

Para un servidor PostgreSQL no hay un solo archivo que copiar, y este manual no fingirá lo contrario. Allí, tu red es el resto del sistema, y es real: solo lectura por defecto (§5), cada cambio revisado en la bandeja (§6) y las guardas de abajo, más las copias que el propio servidor conserve, que siguen siendo la buena práctica del administrador del servidor.

Las guardas: palabras claras antes de daños claros

- **DELETE o UPDATE sin WHERE** — Una cláusula WHERE es la parte que dice qué filas. Sin ella, la sentencia significa todas las filas: así que la consola dice exactamente eso, con palabras claras, antes de que el cambio pueda confirmarse: esto cambiará todas las filas de la tabla. Probablemente no es lo que querías decir, y si es lo que querías decir, puedes decirlo y seguir, con los ojos abiertos.
- **DROP o TRUNCATE** — Estos no cambian una tabla; la terminan. Permanecen desarmados hasta que escribes el nombre exacto de la tabla para armar el **Commit** («Confirmar»): ninguna casilla que marcar por accidente, ningún botón que un clic perdido pueda disparar. Y esta guarda vive en el núcleo de la consola, no solo en sus botones: un DROP que no ha sido armado con el nombre escrito no puede confirmarse, por ningún camino.

Todo lo demás que cambia datos recibe la revisión de la bandeja como su examen. Nada con aspecto destructivo recibe jamás una ejecución silenciosa.

8 — LA BÓVEDA: DONDE VIVEN LAS CONTRASEÑAS

Una conexión a un servidor necesita una contraseña. La regla de DB Console es absoluta: las contraseñas nunca están en texto claro, en ninguna parte: no en la lista de conexiones guardadas (§2), no en un registro, no en una pantalla después de que las hayas escrito. Viven en una bóveda cifrada: un archivo, en tu propia computadora, sellado con una contraseña maestra que tú eliges.

- **La contraseña maestra nunca se guarda.** Ni en la bóveda, ni en ninguna parte. Si la olvidas, la bóveda se queda sellada: eso no es un defecto del diseño; eso es el diseño. Elige una que recordarás.
- **Cómo funciona el sellado, en palabras claras:** tu contraseña maestra pasa por un molino de un solo sentido — una derivación llamada PBKDF2 — 600.000 veces, con un ingrediente aleatorio (una sal, *salt*) mezclado, para fabricar la llave. Los secretos en sí se sellan con AES-GCM, el cifrado autenticado estándar: cambia un bit del archivo sellado y no se abrirá, en vez de abrirse mal.
- **El archivo de la bóveda está bloqueado solo para tu cuenta** (permisos 0600): otros usuarios de la misma máquina no pueden leerlo en absoluto, y sin la contraseña maestra tampoco puede nadie más.
- **Una contraseña equivocada falla en voz alta.** Escribe la contraseña maestra equivocada y la bóveda se niega, con palabras claras. Nunca se abre a medias, nunca adivina, nunca deja una llave equivocada traqueteando dentro.

- **Hay una reserva integrada, y probada.** En un sistema al que le falta la biblioteca de cifrado estándar, la consola sella la bóveda con un método integrado construido sobre los mismos principios — la misma derivación de 600.000 rondas, el mismo bloqueo solo para tu cuenta — en vez de caer jamás al texto claro. No hay ningún camino en este programa que termine con una contraseña legible en un disco.

Toda la historia de las credenciales en una línea: las direcciones a la vista, las contraseñas en la bóveda, y la llave de la bóveda existe solo en tu cabeza.

9 — LOS LÍMITES HONESTOS

Una herramienta en la que puedes confiar te dice lo que no puede hacer, y lo que simplemente todavía no hace. Esta es la lista completa para esta versión:

- **Dos motores hoy.** Esta versión habla SQLite y PostgreSQL. MySQL y MariaDB están en el plan para una fase posterior: no están en esta versión, y nada en este manual debe leerse como si lo estuvieran.
- **Los cambios se escriben, no se hacen clic.** En esta fase cambias datos escribiendo SQL en el editor; los cambios pasan luego por la bandeja, la copia de seguridad y las guardas como cualquier otra escritura. Un formulario de edición de filas a punta de clic es trabajo de una fase posterior: el sistema de seguridad se construyó primero, a propósito.
- **La copia de seguridad es la historia de un archivo.** La copia .bak con marca de tiempo (§7) protege las bases de datos SQLite, que son archivos. La seguridad de un servidor PostgreSQL aquí es el modo solo lectura por defecto, la bandeja de revisión y las guardas, dicho con claridad, porque fingir que existe una copia de archivo para un servidor sería una mentira con una barra de progreso.
- **Dos pequeños asuntos cosméticos, reconocidos.** En una ventana estrecha (800 píxeles de ancho) los botones de página de la cuadrícula pueden cortarse en el borde derecho, y en una ventana baja (600 píxeles de alto) el árbol de esquema se aprieta. Ambos son conocidos, ambos son cosméticos, y ninguno toca un solo byte de tus datos.
- **Probado en bases de datos de prueba; la prueba del fundador es la puerta.** Toda protección de este manual se probó contra bases de datos de prueba construidas a propósito: 37 comprobaciones centrales y 15 comprobaciones de ventana, todas superadas. El programa nunca se ha apuntado a una base de datos real de trabajo en las pruebas, a propósito. Sale solo después de que el fundador lo haya usado con sus propias manos.

10 — TU MÁQUINA, TUS ASUNTOS

La historia de la privacidad es lo bastante corta para imprimirla entera:

- Sin cuentas. Sin telemetría. Sin comprobaciones de actualizaciones. La consola nunca informa a nadie, porque no hay nadie a quien informar.
- Su único tráfico de red son las conexiones que tú defines: con tus propios servidores de bases de datos, cuando te conectas a ellos. Antes de que a esta versión se le permitiera salir,

su interior se escaneó cadena por cadena: no contiene ninguna dirección integrada en absoluto. Ni las nuestras, ni las de nadie. Toda dirección que contenga alguna vez será una que tú escribiste.

- Todo lo que guarda vive contigo: las conexiones guardadas, la bóveda, tus archivos CSV y JSON exportados: archivos locales sencillos en tu propia máquina, exportables y borrables, tuyos.
- Solo puede dañar aquello a lo que la apuntas, y todo el sistema de seguridad (§5–§7) existe para que, incluso allí, dañar algo requiera un acto revisado, respaldado y deliberado. Una herramienta de administración debería hacer que lo cuidadoso sea lo fácil. Esta se construyó para eso.

Eso es todo. Abre tus bases de datos, léelas en el marco dorado, y cuando un cambio sea de verdad necesario: cambia la conexión, revisa la bandeja, deja que la copia aterrice y confirma con los ojos abiertos. La promesa de la consola no es que nada pueda salir mal. Es que nada sale mal por accidente.

DB Console — Enclave DB Console · 777 Software · 777 Admin Tools. Este manual viaja con el programa. Tus bases de datos son tuyas, a tu manera; la forma de esta herramienta — primero solo lectura, revisar antes de ejecutar, una copia bajo cada escritura — no cambia.