



ORION
SOFTWARE
CONSULTING

Orion DB2 Studio

Manual de usuario

Versión 1.1.0 • .NET 10

Contenido

1. Qué es y qué necesitas
2. Instalación
3. Licencia y prueba de 30 días
4. Conectarse a un IBM i
5. El explorador de objetos
6. El diseñador de tablas
7. Comparar bibliotecas
8. Comparar un archivo físico
9. Comparar datos
10. Análisis de impacto
11. Análisis de aplicación
12. El editor SQL
13. Seguridad: lo que no se hace sin confirmar
14. Reportes y línea de mandatos
15. Solución de problemas
16. Apéndice: lista de capturas

1. Qué es y qué necesitas

Orion DB2 Studio es un administrador y diseñador de tablas para **DB2 for i**, la base de datos que llevan dentro los sistemas IBM i (AS/400, iSeries, IBM Power). Sirve para ver qué hay en una biblioteca, crear y modificar tablas sin escribir el SQL a mano, comparar dos bibliotecas y saber qué programas se rompen antes de tocar un archivo.

Se conecta **directamente a los host servers del IBM i**. No hace falta instalar Java, ni IBM i Access Client Solutions, ni controladores ODBC: el controlador viaja dentro del programa.

Plataformas

Escritorio	Windows 10 y 11, y Linux, de 64 bits
Móvil	Android 7.0 o posterior, en modo diseño: sin conexión al IBM i
Plataforma	.NET 10, con la interfaz en Avalonia
Idiomas	Español e inglés, con cambio en caliente

Qué necesita del lado del sistema

Un IBM i 7.3 o posterior accesible por red, con los host servers activos y un perfil de usuario habilitado. La aplicación **no instala nada en el IBM i**: se apoya en los servicios que ya usa IBM i Access Client Solutions.

Versión mínima	IBM i 7.3
Servicios	Host servers arrancados con <code>STRHOSTSVR SERVER(*ALL)</code>
Puertos	8470, 8471 y 8476 sin cifrar; 9470, 9471 y 9476 con TLS
Autorización	El perfil necesita al menos <code>*USE</code> sobre las bibliotecas y objetos que vaya a consultar

La versión del sistema se detecta sola al conectar, leyendo `SYSIBMADM.ENV_SYS_INFO`. Con ella la aplicación desactiva lo que esa versión no soporta: el tipo `BOOLEAN`, por ejemplo, solo se ofrece a partir de 7.5.

Qué hace, en una lista

- **Explorador:** bibliotecas, tablas, vistas, índices, procedimientos, funciones, triggers y secuencias, con sus propiedades, sus columnas, su DDL y sus datos.
- **Diseñador de tablas:** crear y modificar tablas con el DDL generándose al lado y la validación en vivo.
- **Ingeniería inversa:** el DDL real de cualquier objeto con `QSYS2.GENERATE_SQL`, también de los archivos creados con DDS.
- **Comparar bibliotecas:** desarrollo contra producción, con el diff del DDL y el script de sincronización.
- **Comparar archivo físico:** dos archivos campo por campo, leídos del objeto compilado.
- **Comparar datos:** los registros de dos bibliotecas, comparados dentro del propio IBM i.
- **Análisis de impacto:** qué programas, lógicos, vistas y consultas dejan de funcionar al cambiar una tabla.
- **Análisis de aplicación:** el tablero de una biblioteca, «¿dónde se usa?» sobre los fuentes de RPG, CL y COBOL, las relaciones, el diagrama y el diccionario de datos.
- **Editor SQL:** consultas y mandatos CL, con los resultados en una rejilla y exportación a Excel y CSV.
- **Editor de registros:** ver y editar los datos de un archivo, con control de concurrencia y bitácora de auditoría.
- **Reportes y línea de mandatos:** HTML, PDF y Excel, y el ejecutable `orion-db2` para automatizar.

Lo que entró en la versión 1.1.0 —el análisis de aplicación y el editor de registros— se describe en el capítulo 11. El editor de registros, la bitácora, la recuperación de borrados y las herramientas de diagnóstico tienen pantalla propia y todavía **no tienen capítulo en este manual**: se documentarán en la edición siguiente.

2. Instalación

Requisitos

Sistema	Windows 10 u 11 de 64 bits, o Linux de 64 bits
Espacio	Unos 150 MB tras instalar
Permisos	Los de administrador del equipo: el instalador escribe en Archivos de programa
Otros programas	Ninguno. Ni Java, ni ACS, ni ODBC

El tamaño viene del controlador: la aplicación lleva dentro JTOpen —el cliente de IBM para hablar con el IBM i— compilado a .NET. Es el precio de no pedirle al cliente que instale Java.

Descarga

Entre en orionsoftwareconsulting.com/Descargas y baje el instalador de su sistema. Para Windows se ofrecen dos formatos con el mismo contenido:

.MSI	El instalador. Doble clic, y queda en el menú Inicio
Portable	La aplicación suelta en un .zip. No necesita ser administrador: se descomprime y se ejecuta

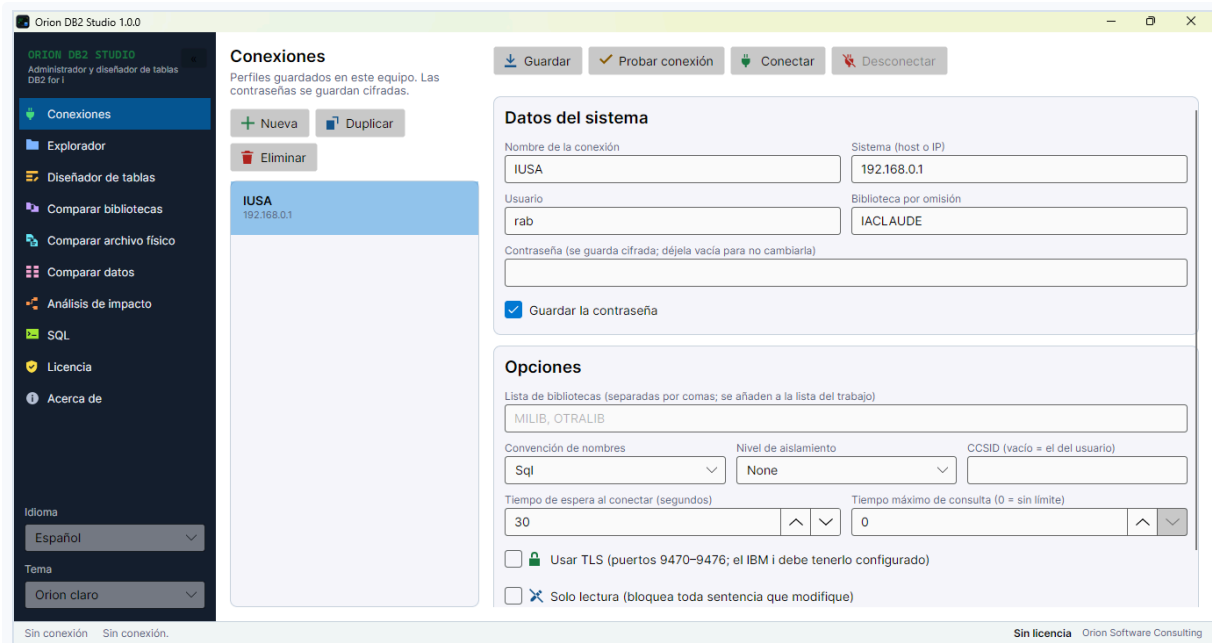
Para Linux hay tres paquetes: [.deb](#) para Debian, Ubuntu, Kali y Mint; [.rpm](#) para Fedora, RHEL, Rocky, Alma y openSUSE; y un [.tar.gz](#) portable que no necesita permisos de root.

Windows avisará de «editor desconocido» al abrir el instalador: todavía no está firmado con un certificado. No es señal de que el archivo venga alterado. Para comprobarlo, en la página de descargas está publicada la huella **SHA-256** de cada archivo; compárela con la del archivo que acaba de bajar antes de ejecutarlo.

Primer arranque

La aplicación abre en la pantalla de **Conexiones**, sin ninguna configurada, y con la barra de estado avisando de que no hay licencia. Las dos cosas se arreglan en los dos capítulos siguientes.

CAPTURA 01



La pantalla de Conexiones, que es por donde se empieza. A la izquierda la barra lateral con las nueve secciones; abajo, los selectores de idioma y de tema.

Idioma y tema

Al final de la barra lateral hay dos listas. **Idioma** cambia toda la interfaz entre español e inglés sin reiniciar. **Tema** ofrece cinco: Orion oscuro, Orion claro, Pantalla verde, Ámbar y Alto contraste. Los cinco están medidos para que el texto se lea con el contraste que pide la norma de accesibilidad WCAG 2.1 AA, y el que elija se recuerda para la próxima vez.

Las figuras de este manual están tomadas con el tema **Orion claro**, que es el que mejor se imprime. Lo que ve en su pantalla puede tener otros colores sin que nada haya cambiado de sitio.

3. Licencia y prueba de 30 días

Orion DB2 Studio se vende con **licencia única y perpetua**: se paga una vez y no caduca. Hay una sola edición, con todas las funciones, y cada licencia sirve para **dos equipos**.

Precio de lanzamiento	22,99 USD
Equipos	Hasta 2 por licencia
Prueba	30 días con todas las funciones
Al vencer	Modo solo lectura: explorar, ver el DDL y generar reportes

Dónde se ve el estado

En la sección **Licencia**, y de un vistazo en la esquina derecha de la barra de estado, que acompaña a la aplicación en todas las pantallas.

CAPTURA 02



La pantalla de Licencia: el estado actual, los datos del titular, el botón de la prueba gratuita, el cuadro para activar una clave comprada y el identificador de este equipo.

Pedir la prueba

1. Rellene **Correo y Nombre**. El correo es donde llegará la clave si después compra.

2. Pulse **Solicitar la prueba gratuita**.

3. La aplicación habla con el servidor de licencias de Orion y deja la prueba activada en este equipo.

La prueba se concede **una sola vez por equipo** y **necesita conexión a internet** en ese momento. Después se puede trabajar sin red: la aplicación revalida de vez en cuando, no en cada arranque.

Activar una clave comprada

Escriba la clave en el cuadro de **Activar** y pulse el botón. La activación queda ligada a un identificador de este equipo que se calcula del propio sistema operativo y **no contiene datos personales**. Puede verlo, y guardarlo en un archivo, al final de la misma pantalla.

Para mover la licencia a otro equipo —un portátil nuevo, un cambio de disco— use **Liberar este equipo**: suelta la activación y deja el hueco libre para otra máquina.

Qué pasa al terminar la prueba. La aplicación sigue abriendo y no se pierde nada: puede explorar objetos, ver su DDL y generar reportes. Lo que queda bloqueado es **ejecutar cambios** y **lanzar mandatos CL**. La barra de estado lo dice y cada pantalla que lo necesita explica por qué.

4. Conectarse a un IBM i

Una **conexión** es un perfil guardado en su equipo: el sistema, el usuario, la contraseña cifrada y las opciones de la sesión. Puede tener tantos como sistemas administre, y la contraseña nunca se guarda en texto plano.

Crear el primer perfil

1. Pulse **Nueva**.
2. Rellene **Nombre de la conexión** —el que verá usted, por ejemplo «Producción»—, **Sistema (host o IP)**, **Usuario** y **Contraseña**.
3. Indique la **Biblioteca por omisión**, que es con la que empiezan las pantallas.
4. Pulse **Guardar**.

Campos del perfil

Nombre de la conexión	Cómo la llama usted. No viaja al sistema
Sistema (host o IP)	El nombre de red del IBM i, o su dirección IP
Usuario y contraseña	El perfil de usuario del IBM i. La contraseña se cifra con el sistema operativo antes de guardarla
Biblioteca por omisión	Con la que arrancan el explorador y el editor SQL
Guardar la contraseña	Si la desmarca, la aplicación la pedirá cada vez que conecte

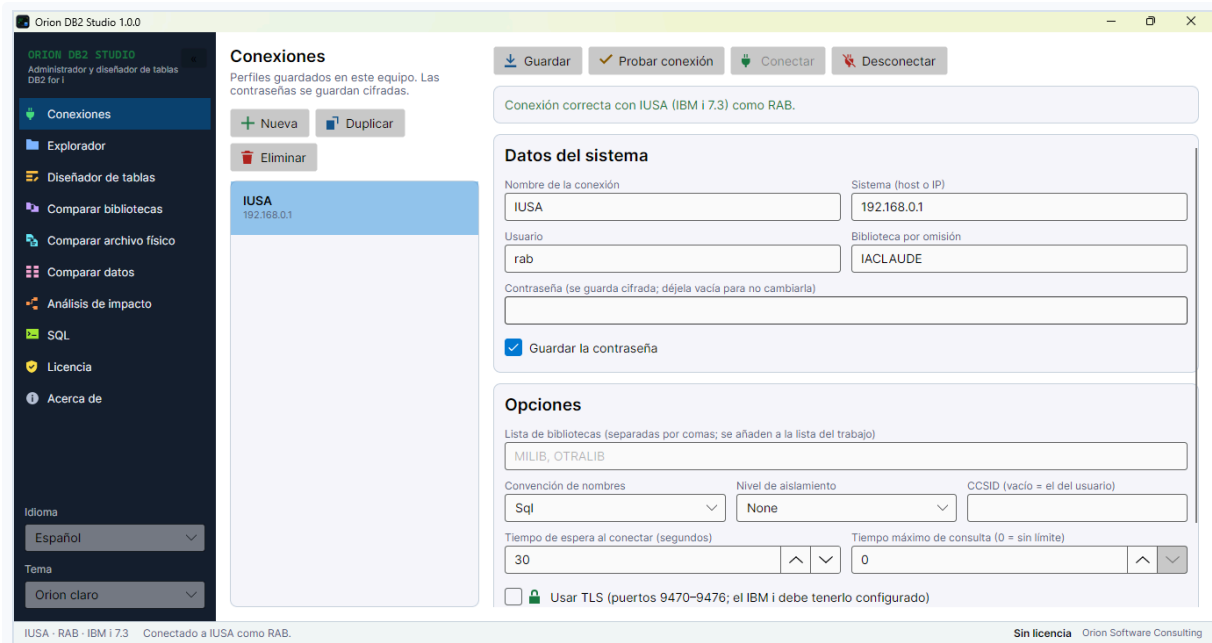
Opciones

Lista de bibliotecas	Se añaden a la lista del trabajo, separadas por comas
Convención de nombres	<code>Sql</code> usa el punto (<code>BIBLIOTECA.TABLA</code>); <code>System</code> usa la barra (<code>BIBLIOTECA/TABLA</code>)
Nivel de aislamiento	El control de compromiso de la sesión
CCSID	Juego de caracteres. Vacío = el del perfil de usuario
Tiempo de espera al conectar	Segundos antes de darse por vencido
Tiempo máximo de consulta	Segundos por sentencia. 0 = sin límite
Usar TLS	Cifra la conexión. El IBM i debe tener SSL configurado
Solo lectura	Bloquea toda sentencia que modifique y todo mandato CL, en la capa de datos
PRODUCCIÓN	Franja roja permanente y confirmación escrita antes de cada cambio

Probar antes de conectar

Probar conexión comprueba que se puede llegar al sistema y contesta en un aviso con el resultado. Si funciona, dice con qué sistema habló, qué versión de IBM i tiene y con qué usuario entró.

CAPTURA 03



«Probar conexión» contra un IBM i 7.3: el aviso con el resultado y el mismo mensaje en la franja de arriba.

Si falla, el mensaje **dice por qué y qué hacer**, en vez de un error genérico. Distingue estos casos:

Qué pasó	Qué le dirá
El nombre no resuelve	Que compruebe el nombre o use la dirección IP
El sistema no contesta	Que puede estar apagado o que un cortafuegos descarta el tráfico
El sistema rechaza la conexión	Que ejecute <code>STRHOSTSVR SERVER(*ALL)</code> , porque los host servers no están
Pidió TLS y el sistema no lo tiene	Que desactive TLS, o que configure SSL en el IBM i. No le manda ejecutar STRHOSTSVR, porque el servicio sí está
El sistema solo acepta TLS	Que active la casilla TLS
Usuario o contraseña incorrectos	Que lo revise, y le recuerda que el IBM i deshabilita el perfil tras varios intentos
Perfil deshabilitado	Que pida <code>CHGUSRPRF STATUS(*ENABLED)</code>
Contraseña vencida	Que entre por 5250 o ACS a cambiarla

Antes de llamar al controlador, la aplicación **sondea el puerto** de base de datos. Por eso un sistema apagado se detecta en unos segundos en vez de esperar el plazo completo.

Conectar

Conectar abre la sesión y la deja activa para el resto de las pantallas. La barra de estado pasa a enseñar el sistema, el usuario y la versión.

CAPTURA 04

The screenshot shows the Orion DB2 Studio 1.0.0 application window. On the left is a dark sidebar with a menu containing 'Conexiones', 'Explorador', 'Diseñador de tablas', 'Comparar bibliotecas', 'Comparar archivo físico', 'Comparar datos', 'Análisis de impacto', 'SQL', 'Licencia', and 'Acerca de'. Below the menu are settings for 'Idioma' (set to 'Español') and 'Tema' (set to 'Orion claro'). The main window has a title bar 'Orion DB2 Studio 1.0.0'. Below it is a 'Conexiones' section with a subtitle 'Perfiles guardados en este equipo. Las contraseñas se guardan cifradas.' and buttons for '+ Nueva', 'Duplicar', and 'Eliminar'. A list of connections shows 'IUSA' with host '192.168.0.1' selected. To the right of this list are buttons: 'Guardar', 'Probar conexión', 'Conectar', and 'Desconectar'. Below these is a status bar that says 'Conectado a IUSA como RAB.'. The main configuration area is titled 'Datos del sistema' and contains fields for 'Nombre de la conexión' (IUSA), 'Sistema (host o IP)' (192.168.0.1), 'Usuario' (rab), and 'Biblioteca por omisión' (IACLAUDE). There is a password field and a checkbox 'Guardar la contraseña' which is checked. Below this is the 'Opciones' section with a text area for 'Lista de bibliotecas' containing 'MILIB, OTRALIB'. It also has dropdowns for 'Convención de nombres' (Sql) and 'Nivel de aislamiento' (None), and a text field for 'CCSID'. At the bottom of the options are sliders for 'Tiempo de espera al conectar (segundos)' (set to 30) and 'Tiempo máximo de consulta (0 = sin límite)' (set to 0). A checkbox 'Usar TLS' is unchecked. The status bar at the very bottom shows 'IUSA - RAB - IBM i 7.3' on the left and 'Sin licencia Orion Software Consulting' on the right.

Conectado: el mensaje en la franja superior y, abajo a la izquierda, el sistema, el usuario y la versión de IBM i.

Si la conexión está marcada como **PRODUCCIÓN**, la aplicación pide confirmación al conectar y después enseña una franja roja permanente en la parte de arriba, con el sistema y la biblioteca, que no se puede quitar mientras esa sesión esté abierta.

Si la conexión se cae sola

Cada veinte segundos la aplicación comprueba que la sesión sigue viva. Si el IBM i, la VPN o el cortafuegos la cortaron —cosa habitual tras un rato sin actividad—, lo avisa en un diálogo, cierra la sesión y lo deja en claro en la barra de estado. Vuelva a pulsar **Conectar**.

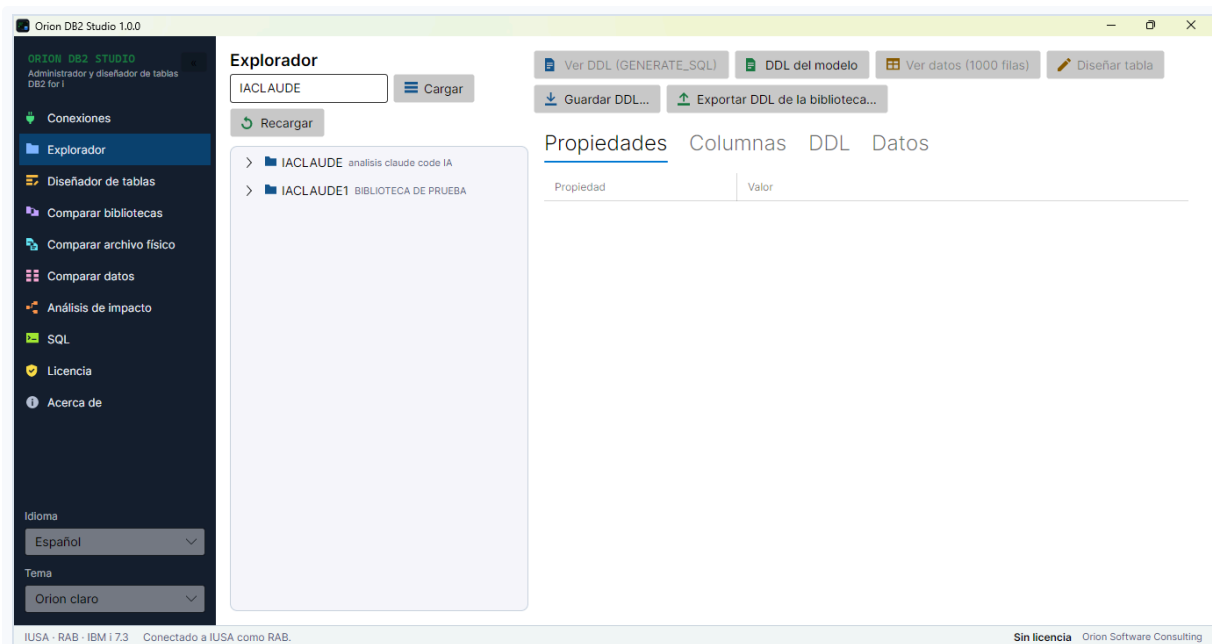
5. El explorador de objetos

El **Explorador** es el árbol de lo que hay en el sistema: bibliotecas y, dentro de cada una, tablas, vistas, índices, procedimientos, funciones, triggers y secuencias. Se carga por partes, cuando abre cada nodo, para no traerse una biblioteca entera que a lo mejor no necesita.

Cargar una biblioteca

Escriba el nombre —o parte de él— en el cuadro de arriba y pulse **Cargar**. Puede usar un nombre genérico: **IA** trae todas las que empiecen por IA.

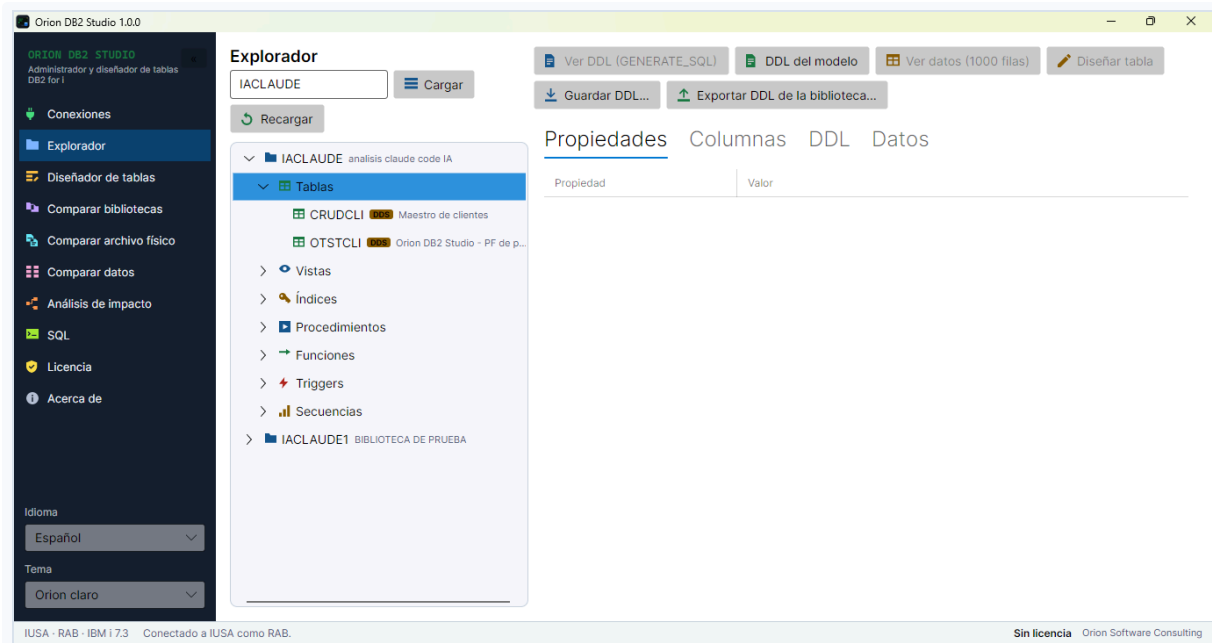
CAPTURA 05



El explorador tras cargar el filtro: las bibliotecas que coinciden, cada una con su texto descriptivo.

Abra una biblioteca y verá las siete familias de objetos. Abra una familia y verá sus objetos, cada uno con su nombre, su texto y —cuando difiere del nombre SQL— su nombre de sistema.

CAPTURA 06



El árbol abierto: biblioteca, familias de objetos y, dentro de Tablas, los archivos con su distintivo DDS.

El distintivo **DDS** marca los archivos que se crearon con DDS en vez de con SQL. No es un defecto ni una advertencia: es información que cambia lo que se puede hacer con ellos, y la aplicación los trata igual que a los demás.

Las cuatro pestañas

Al seleccionar un objeto, el panel de la derecha lo describe en cuatro pestañas.

Propiedades

Nombre SQL y nombre de sistema, biblioteca, tipo, texto, número de columnas y de filas, tamaño, si está registrado por diario (**journaling**), la llave primaria, la fecha de creación y si el objeto se creó con SQL o con DDS.

CAPTURA 07

The screenshot shows the Orion DB2 Studio 1.0.0 interface. On the left is a dark sidebar with navigation options like 'Explorador', 'Diseñador de tablas', and 'Comparar bibliotecas'. The main area is divided into three panes. The 'Explorador' pane on the left shows a tree structure with 'IACLAUDE' as the selected database. The central pane shows the 'Propiedades' (Properties) tab for the table 'OTSTCLI'. It lists various attributes such as 'Nombre SQL', 'Nombre de sistema', 'Biblioteca', 'Tipo', 'Texto', 'Origen', 'Columnas', 'Filas', 'Tamaño', 'Journaling', 'Llave primaria', and 'Creada'. The right pane shows the 'Columnas' (Columns) tab, which is currently empty. At the bottom, a status bar indicates 'IUSA - RAB - IBM i 7.3' and 'Conectado a IUSA como RAB.'.

Propiedad	Valor
Nombre SQL	OTSTCLI
Nombre de sistema	OTSTCLI
Biblioteca	IACLAUDE
Tipo	Tabla
Texto	Orion DB2 Studio - PF de prueba
Origen	Creada con DDS
Columnas	10
Filas	50,000
Tamaño	9.3 MB
Journaling	No
Llave primaria	CLICOD (vía de acceso DDS)
Creada	19/10/2026 06:12

Propiedades de una tabla: lo que hay que mirar antes de tocarla.

Columnas

Una fila por columna, con el nombre SQL, el nombre de sistema, el tipo, la longitud, los decimales, el CCSID y si admite nulos. La llave primaria se marca con un icono a la izquierda.

CAPTURA 08

The screenshot shows the Orion DB2 Studio 1.0.0 interface. On the left is a dark sidebar with navigation options like 'Conexiones', 'Explorador', 'Diseñador de tablas', etc. The main area is divided into 'Explorador' (left) and 'Propiedades' (right). The 'Explorador' shows a tree structure with 'IACLAUDE' selected. The 'Propiedades' tab is set to 'Columnas', displaying a table of columns for the selected object 'OTSTCLI'.

	Columna	Sistema	Tipo	Longit	D	CCS	Nu
1	CLICOD	CLICOD	DECIMAL	6	0		☐
2	CLINOM	CLINOM	CHAR	40		284	☐
3	CLIDIR	CLIDIR	CHAR	60		284	☐
4	CLICIU	CLICIU	CHAR	30		284	☐
5	CLIPAI	CLIPAI	CHAR	3		284	☐
6	CLITEL	CLITEL	CHAR	20		284	☐
7	CLILIM	CLILIM	DECIMAL	11	2		☐
8	CLISLD	CLISLD	DECIMAL	11	2		☐
9	CLIFAL	CLIFAL	DATE				☐
10	CLIACT	CLIACT	CHAR	1		284	☐

Las columnas de un archivo físico creado con DDS, con la llave señalada en la primera fila.

DDL

El DDL **real** del objeto, el que devuelve el propio sistema con `QSYS2.GENERATE_SQL`. Funciona también con los archivos creados con DDS: es la forma de recuperar la definición de un archivo del que ya nadie tiene el fuente.

CAPTURA 09

The screenshot shows the Orion DB2 Studio 1.0.0 interface. On the left is a dark sidebar with navigation options like 'Conexiones', 'Explorador', 'Diseñador de tablas', and 'Comparar bibliotecas'. The main area is divided into two panes. The left pane, titled 'Explorador', shows a tree view of the database structure with 'IACLAUDE' selected. The right pane, titled 'Propiedades', has tabs for 'Columnas', 'DDL', and 'Datos'. The 'Columnas' tab is active, displaying a table with 10 columns. The table has columns for 'Columna', 'Sistema', 'Tipo', 'Longit', 'D', 'CCS', and 'Nu'. The data rows show columns like CLICOD, CLINOM, CLIDIR, CLICIU, CLIPAI, CLITEL, CLILIM, CLISLD, CLIFAL, and CLIACT with their respective data types and lengths.

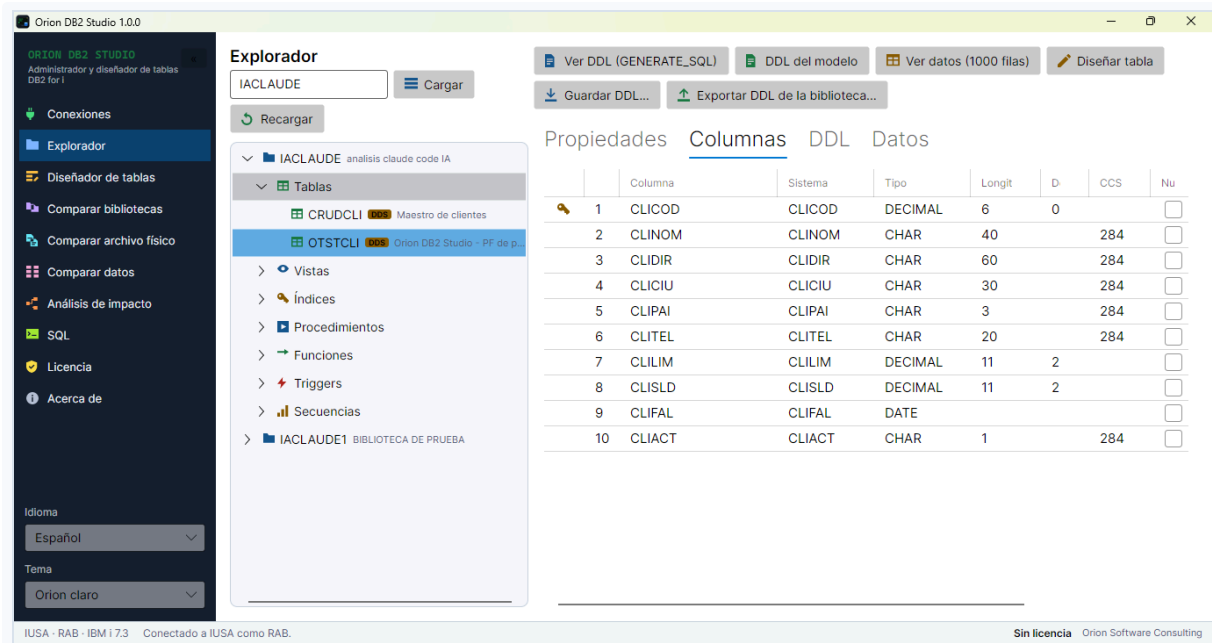
	Columna	Sistema	Tipo	Longit	D	CCS	Nu
1	CLICOD	CLICOD	DECIMAL	6	0		☐
2	CLINOM	CLINOM	CHAR	40		284	☐
3	CLIDIR	CLIDIR	CHAR	60		284	☐
4	CLICIU	CLICIU	CHAR	30		284	☐
5	CLIPAI	CLIPAI	CHAR	3		284	☐
6	CLITEL	CLITEL	CHAR	20		284	☐
7	CLILIM	CLILIM	DECIMAL	11	2		☐
8	CLISLD	CLISLD	DECIMAL	11	2		☐
9	CLIFAL	CLIFAL	DATE				☐
10	CLIACT	CLIACT	CHAR	1		284	☐

El panel de la derecha, con las pestañas Propiedades, Columnas, DDL y Datos.

Datos

Las primeras **1.000 filas** del archivo, para mirar el contenido sin salir a otra herramienta. Está disponible para tablas y vistas; en un índice, la aplicación indica cuál es la tabla base.

CAPTURA 10



Las primeras filas de un archivo, dentro del propio explorador.

Los botones de arriba

Ver DDL (GENERATE_SQL)	El DDL real del objeto, tal como está hoy
DDL del modelo	El DDL que generaría Orion a partir de lo que leyó del catálogo. Es la vía para modernizar un archivo DDS a SQL
Ver datos (1000 filas)	Las primeras mil filas
Diseñar tabla	Carga la tabla en el diseñador
Guardar DDL...	Guarda lo que muestra la pestaña DDL en un <code>.sql</code>
Exportar DDL de la biblioteca...	El DDL de todos los objetos de la biblioteca, un archivo por objeto, en la carpeta que elija

Diferencia importante entre los dos DDL. «Ver DDL» es lo que dice el sistema. «DDL del modelo» es lo que Orion reconstruye, y hoy **no convierte la llave** de un archivo DDS en `PRIMARY KEY` ni en `CREATE INDEX`: revísela antes de usar ese script para crear el objeto en otro sitio.

6. El diseñador de tablas

El **Diseñador** arma el DDL por usted. Funciona en dos modos: **crear** una tabla nueva, y **modificar** una que ya existe, donde genera el `ALTER TABLE` mínimo comparando lo que había con lo que usted dejó.

Nada se ejecuta hasta que lo confirme. Mientras edita, el DDL se va escribiendo al lado y el panel de validación avisa de los errores.

La cabecera

Biblioteca	Dónde se creará la tabla
Nombre SQL de la tabla	El nombre largo, sin el límite de 10 caracteres
Nombre de sistema (máx. 10)	El nombre corto del objeto. Si lo deja vacío, el sistema lo inventa
Texto de la tabla	<code>LABEL ON TABLE</code> , hasta 50 caracteres
Comentario	<code>COMMENT ON TABLE</code> , para la descripción larga

Columnas

La rejilla tiene una fila por columna: nombre SQL, nombre de sistema, tipo, longitud, decimales, CCSID, si admite nulos, valor por omisión, encabezado de columna y texto.

Los botones de arriba añaden y quitan columnas y las mueven de sitio. Dos merecen explicación:

Columnas de auditoría	Añade de un clic usuario y fecha de creación y de modificación, con sus valores por omisión ya puestos
Sugerir nombres de sistema	Propone el nombre corto de cada columna imitando el algoritmo del propio DB2 for i, y evita los duplicados

CAPTURA 11

Orion DB2 Studio 1.0.0

ORION DB2 STUDIO
Administrador y diseñador de tablas DB2 for i

Conexiones
Explorador
Diseñador de tablas
Comparar bibliotecas
Comparar archivo físico
Comparar datos
Análisis de impacto
SQL
Licencia
Acerca de

Idioma
Español
Tema
Orion claro

Crear tabla
Diseñe la tabla y vea el DDL generarse en tiempo real. Nada se ejecuta hasta que lo confirme.

Nueva tabla Añadir columna Quitar columna ↑ ↓ Columnas de auditoría Sugerir nombres de sistema

Biblioteca Nombre SQL de la tabla Nombre de sistema (máx. 10)
Texto de la tabla (LABEL ON TABLE, máx. 50) Comentario (COMMENT ON TABLE)

Columnas Llave primaria Únicas Foráneas Checks Índices

	Nombre SQL	Sistema	Tipo	Lo	I	CC:	Por defecto	Encabezado (60)	Texto (50)
1	ID		INTEGER				☒		
2	USUARIO_CREACION	USRCRE	VARCHAR	18			☐	USER	Usuario de creación
3	FECHA_CREACION	FECCRE	TIMESTAMP				☐	CURRENT	Fecha de creación
4	USUARIO_MODIFICACION	USRMOD	VARCHAR	18			☐	USER	Usuario de modificación
5	FECHA_MODIFICACION	FECMOD	TIMESTAMP				☐		Fecha de modificación Última modificación de la

Guardar .sql Ejecutar ☐ 80 col. ☐ DROP antes

IUSA - RAB - IBM i 7.3 Conectado a IUSA como RAB. Sin licencia Orion Software Consulting

El diseñador con las columnas de auditoría recién añadidas: usuario y fecha de creación y de modificación, con sus valores por omisión.

Llaves, restricciones e índices

Las pestañas de la rejilla recogen el resto de la definición:

Llave primaria	Marque las columnas y el orden de cada una
Únicas	Restricciones de unicidad sobre una o varias columnas
Foráneas	Con su biblioteca, tabla y columnas de destino, y qué hacer al borrar
Checks	Condiciones que cada fila debe cumplir
Índices	Con su propio nombre de sistema

CAPTURA 12

Orion DB2 Studio 1.0.0

ORION DB2 STUDIO
Administrador y diseñador de tablas DB2 for i

Conexiones
Explorador
Diseñador de tablas
Comparar bibliotecas
Comparar archivo físico
Comparar datos
Análisis de impacto
SQL
Licencia
Acerca de

Idioma
Español
Tema
Orion claro

Crear tabla
Diseñe la tabla y vea el DDL generarse en tiempo real. Nada se ejecuta hasta que lo confirme.

Nueva tabla Añadir columna Quitar columna ↑ ↓ Columnas de auditoría Sugerir nombres de sistema

Biblioteca Nombre SQL de la tabla Nombre de sistema (máx. 10)
Texto de la tabla (LABEL ON TABLE, máx. 50) Comentario (COMMENT ON TABLE)

Columnas Llave primaria Únicas Foráneas Checks Índices

Marque las columnas que forman la llave primaria. Todas deben ser NOT NULL.

Llave	Columna	Descendente
☒	ID	☐
☐	USUARIO_CREACION	☐
☐	FECHA_CREACION	☐
☐	USUARIO_MODIF	☐
☐	FECHA_MODIF	☐

Guardar .sql Ejecutar ☐ 80 col. ☐ DROP antes

IUSA - RAB - IBM i 7.3 Conectado a IUSA como RAB. Sin licencia Orion Software Consulting

La pestaña de llave primaria: se marcan las columnas que la forman y el orden de cada una.

Validación y vista previa

El panel de validación no espera a que pulse nada: avisa mientras escribe de los nombres de sistema repetidos o inválidos, de los decimales mayores que la longitud, de los tipos que su versión de IBM i no soporta y de las tablas sin llave primaria. El resumen dice cuántos errores y cuántas advertencias hay.

Guardar o ejecutar

Guardar .sql	Deja el script listo para <code>RUNSQLSTM</code>
Ejecutar	Lo aplica contra el IBM i, tras la pantalla de confirmación
80 col.	Parte las líneas a 80 columnas, como pide <code>RUNSQLSTM</code>
DROP antes	Añade el <code>DROP TABLE</code> previo, solo en modo crear

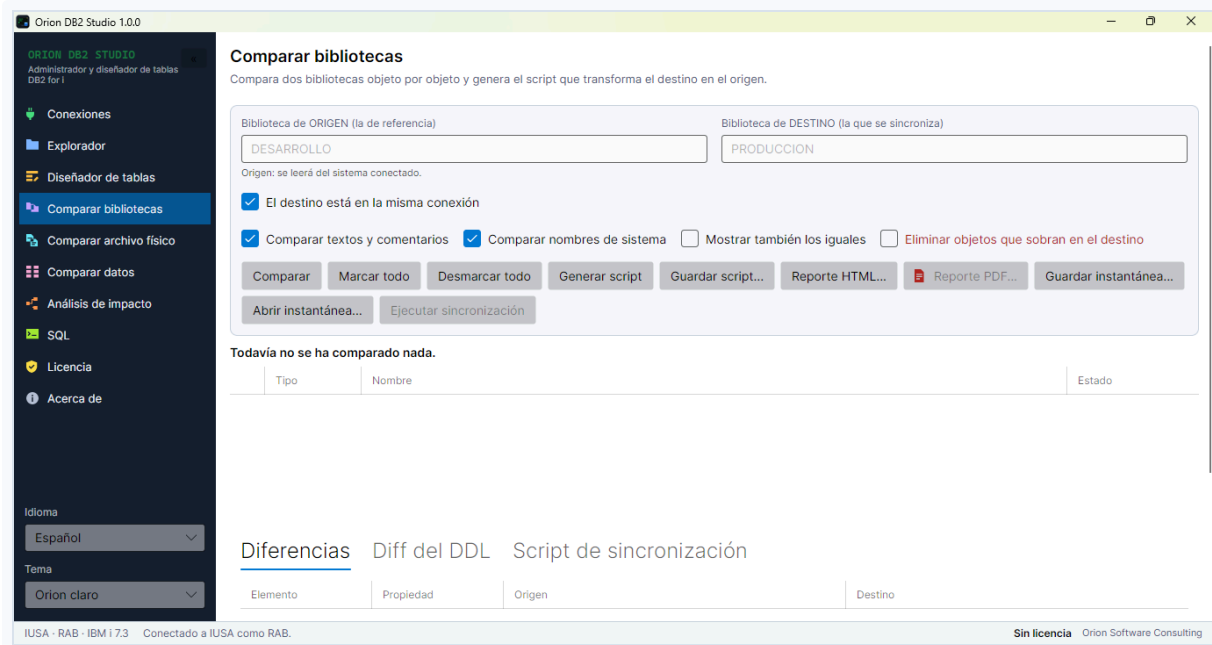
Modificar una tabla no es gratis en DB2 for i. Un `ALTER TABLE` puede obligar al sistema a reconstruir el archivo, y eso con una tabla grande o con lógicos encima lleva tiempo. La aplicación marca cada sentencia con su riesgo y, cuando el cambio altera el formato de registro, le ofrece revisar el **análisis de impacto** antes de firmar.

DB2 for i no sabe renombrar una columna. Cuando usted cambia el nombre de una, el diseñador genera `ADD COLUMN`, `UPDATE` y `DROP COLUMN`, lo marca como pérdida de datos y le advierte de que esos tres pasos no son atómicos si la tabla no está registrada por diario.

7. Comparar bibliotecas

Compara dos bibliotecas objeto por objeto y genera el script que convierte el destino en el origen. Las dos pueden estar en sistemas distintos.

CAPTURA 13



La pantalla de comparación: origen, destino, opciones y la rejilla de resultados.

Qué se compara

Tablas y sus columnas —tipo, longitud, nulos, valor por omisión, CCSID y textos—, restricciones, índices, vistas, rutinas, triggers y secuencias. **No** se comparan los permisos **GRANT**.

La comparación se hace sobre el **modelo** leído de los catálogos, no sobre el texto del DDL: dos objetos iguales escritos de forma distinta no aparecen como diferentes.

Opciones

El destino está en la misma conexión	Desmárquelo para comparar contra otro sistema, y elija su conexión
Comparar textos y comentarios	Incluye <code>LABEL ON</code> y <code>COMMENT ON</code>
Comparar nombres de sistema	Tiene en cuenta el nombre corto del objeto
Mostrar también los iguales	Para ver la biblioteca entera, no solo lo que cambia
Eliminar objetos que sobran en el destino	Incluye los <code>DROP</code> en el script. Va en rojo por lo que significa

El resultado

Cada fila lleva su estado —Solo en origen, Solo en destino, Diferente o Igual— y una casilla. El script se genera solo con lo que marque.

Dos pestañas explican cada diferencia: **Diferencias** la desglosa propiedad por propiedad, y **Diff del DDL** enseña los dos DDL lado a lado, línea por línea.

El script de sincronización

Generar script arma los `CREATE`, `ALTER` y `DROP` en orden de dependencias: primero lo que otros objetos necesitan. Puede guardarlo con **Guardar script...** o aplicarlo con **Ejecutar sincronización**, que pasa antes por la pantalla de confirmación.

Instantáneas

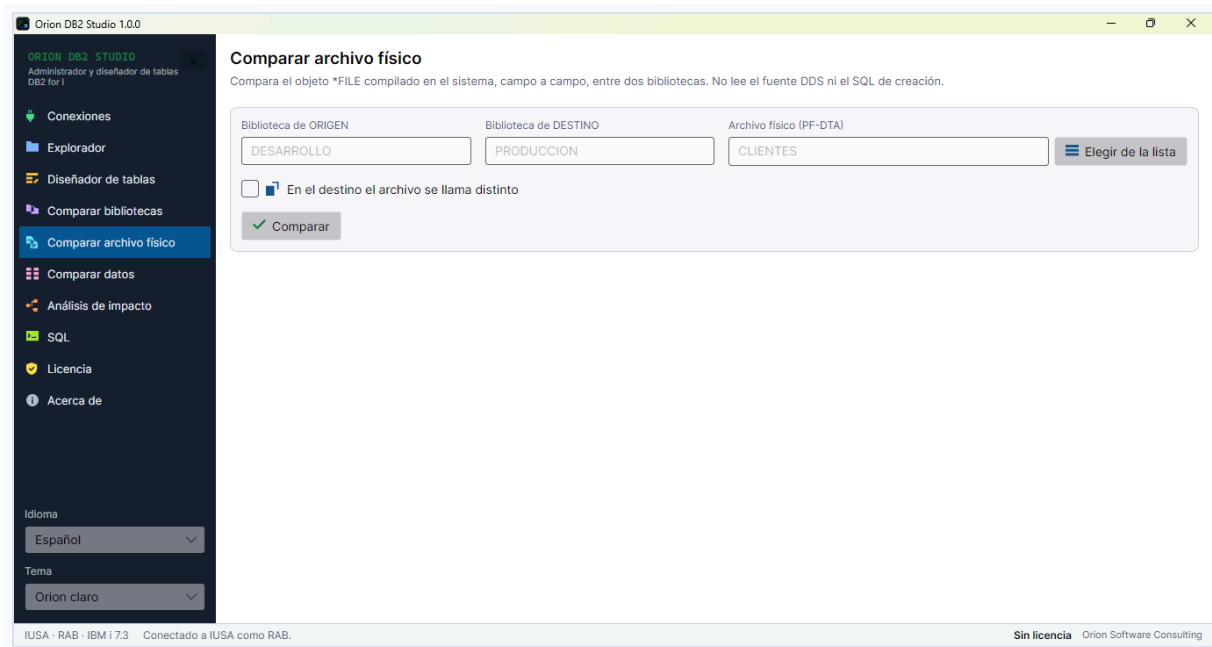
Guardar instantánea... escribe el modelo completo de una biblioteca en un archivo JSON. Sirve para comparar el mismo sistema contra sí mismo dentro de unos meses, o para comparar sin tener delante el sistema de origen: **Abrir instantánea...** la carga como origen.

Los reportes de la comparación salen en **HTML** —autocontenido, para enviar por correo— y en **PDF**.

8. Comparar un archivo físico

Compara dos archivos físicos **campo por campo**, leyendo el objeto compilado y no el fuente DDS ni el SQL con que se creó. Es la pantalla a la que acudir cuando un programa se cae con **CPF4131** y hay que averiguar en qué se diferencian el archivo de desarrollo y el de producción.

CAPTURA 14



Comparación de archivo físico: el archivo, las dos bibliotecas y el resultado por campos, llaves y atributos.

Todo lo que hace contra el IBM i es de **solo lectura**.

Qué hay que indicar

1. La **biblioteca de origen** y la **de destino**.
2. El **archivo**. Con **Elegir de la lista** sale la lista de archivos físicos de datos de la biblioteca de origen.
3. Si en el destino se llama distinto, marque la casilla y escriba el otro nombre.
4. Pulse **Comparar**.

Qué contesta

Lo primero, el **identificador de nivel de formato** de los dos lados: la huella que el sistema calcula de la estructura del archivo. Si las dos huellas son iguales, los programas compilados contra uno abren el otro sin level check.

Después, las diferencias clasificadas en **críticas**, **advertencias** e **informativas**, repartidas en tres pestañas:

Campos	Ordinal y definición en los dos lados, qué cambia y con qué gravedad. Los campos se emparejan por nombre, no por posición
Llaves	La vía de acceso de cada lado y en qué se diferencian
Atributos del archivo	Lo que define el objeto, más allá de sus campos

El resultado se baja en **PDF** con **Descargar reporte PDF**.

El identificador de nivel cubre **el formato de registro**. La llave y los atributos del archivo pueden diferir con la misma huella, y por eso se comparan aparte.

9. Comparar datos

Compara los registros del mismo archivo entre dos bibliotecas. La comparación se ejecuta **dentro del IBM i**: a su equipo solo llegan los conteos y las páginas de resultados que mire, no los datos enteros.

CAPTURA 15

Orion DB2 Studio 1.0.0

ORION DB2 STUDIO
Administrador y diseñador de tablas DB2 for i

Conexiones

Explorador

Diseñador de tablas

Comparar bibliotecas

Comparar archivo físico

Comparar datos

Análisis de impacto

SQL

Licencia

Acerca de

Idioma
Español

Tema
Orion claro

Comparar datos de archivo físico

Compara los REGISTROS del mismo archivo entre dos bibliotecas. Toda la comparación se hace en el IBM i: a esta máquina solo llegan conteos y páginas de resultados.

QUÉ SE COMPARA

Biblioteca de ORIGEN

DESARROLLO

Biblioteca de DESTINO

PRODUCCION

Archivo físico (PF-DTA)

FACTDET

Elegir de la lista

☐ En el destino se llama distinto

Miembro *FIRST

REGISTROS EN EL ORIGEN

0

REGISTROS EN EL DESTINO

0

Los totales se leen al elegir el archivo, antes de comparar nada.

CÓMO SE EMPAREJA UN REGISTRO CON EL OTRO

☒ Usar la llave única del archivo
la llave única del archivo

☐ Elegir los campos que identifican el registro
Obligatorio si el archivo no tiene llave o la tiene pero no es única.

☐ Comparar por RRN (número relativo de registro)
Empareja el registro 1 con el registro 1, el 2 con el 2...

CAMPOS — marque la llave y lo que no se debe comparar

Excluir suele hacer falta con fechas y horas de auditoría y con el usuario de modificación: cambian siempre y esconden las diferencias que importan.

Campo	Tipo	Llave	Excluir
-------	------	-------	---------

IUSA - RAB - IBM i 7.3 Conectado a IUSA como RAB.

Sin licencia Orion Software Consulting

Los parámetros de la comparación de datos: archivo, miembro, cómo emparejar los registros y qué campos excluir.

Cómo se emparejan los registros

Hay tres formas, y elegir bien es la mitad del trabajo:

Por la llave única del archivo	Lo normal cuando el archivo tiene llave única. Se marca sola
Por los campos que usted elija	Obligatorio si el archivo no tiene llave, o la tiene y no es única
Por RRN	Empareja el registro 1 con el 1, el 2 con el 2. Útil cuando no hay ninguna llave posible

Cuidado con el RRN. El número relativo de registro es la posición física dentro del archivo: un solo **RGZPFM** en uno de los dos lados lo cambia todo y la comparación deja de significar nada. La aplicación lo advierte en pantalla al elegirlo.

Qué excluir

En la rejilla de campos puede marcar cuáles entran en la llave y cuáles **se excluyen** de la comparación. Conviene excluir las fechas y horas de auditoría y el usuario de modificación: son distintos en los dos sistemas casi siempre, y si no se excluyen **todos** los registros salen como diferentes.

Opciones que afectan al tiempo

Filtro WHERE	Se aplica igual a los dos lados, y se valida contra el servidor antes de empezar
Ignorar espacios finales en campos CHAR	En un archivo físico los campos son de longitud fija: el relleno es del formato, no del dato
Comparación rápida por hash	Compara un hash por registro y baja al detalle solo donde difiera. Se desactiva sola si el sistema no tiene <code>HASH_SHA256</code>
Registros por tramo	Cuántos registros se traen de una vez
Máximo de diferencias a detallar en el PDF	El tope del detalle del reporte

Mientras va, y el resultado

La pantalla de progreso enseña la etapa, el tramo, el tiempo transcurrido y las etapas ya terminadas con lo que costó cada una. **Cancelar** corta la consulta en el servidor y borra lo que se hubiera creado en `QTEMP`.

Al terminar hay seis contadores —registros en origen y en destino, iguales, diferentes, solo en origen y solo en destino—, **que son siempre totales** aunque el detalle esté topado, y una lista de en qué campos están las diferencias, con cuántos registros afecta cada uno.

El detalle se exporta a **PDF** y, sin el tope del PDF, a **CSV**.

Insertar los registros que faltan

Insertar en el destino los registros que faltan copia al destino lo que solo está en el origen. Enseña el SQL antes de ejecutar nada y pide confirmación escrita cuando el destino no está registrado por diario, porque entonces no habrá forma de deshacerlo.

Solo inserta. No actualiza los registros que difieren ni borra los que sobran, y no funciona en modo RRN. Es deliberado: una sincronización completa de datos entre dos entornos es una decisión de negocio, no un botón.

10. Análisis de impacto

Contesta a la pregunta que en IBM i nadie responde de memoria: **si cambio esta tabla, qué deja de funcionar mañana**. El sistema no avisa. Un programa compilado guarda la huella del formato de registro y, al abrir el archivo, o coincide o se cae con **CPF4131**.

CAPTURA 16



La pantalla de análisis antes de empezar: la tabla, las bibliotecas donde buscar y los cambios que llegan del diseñador.

Cómo se lanza

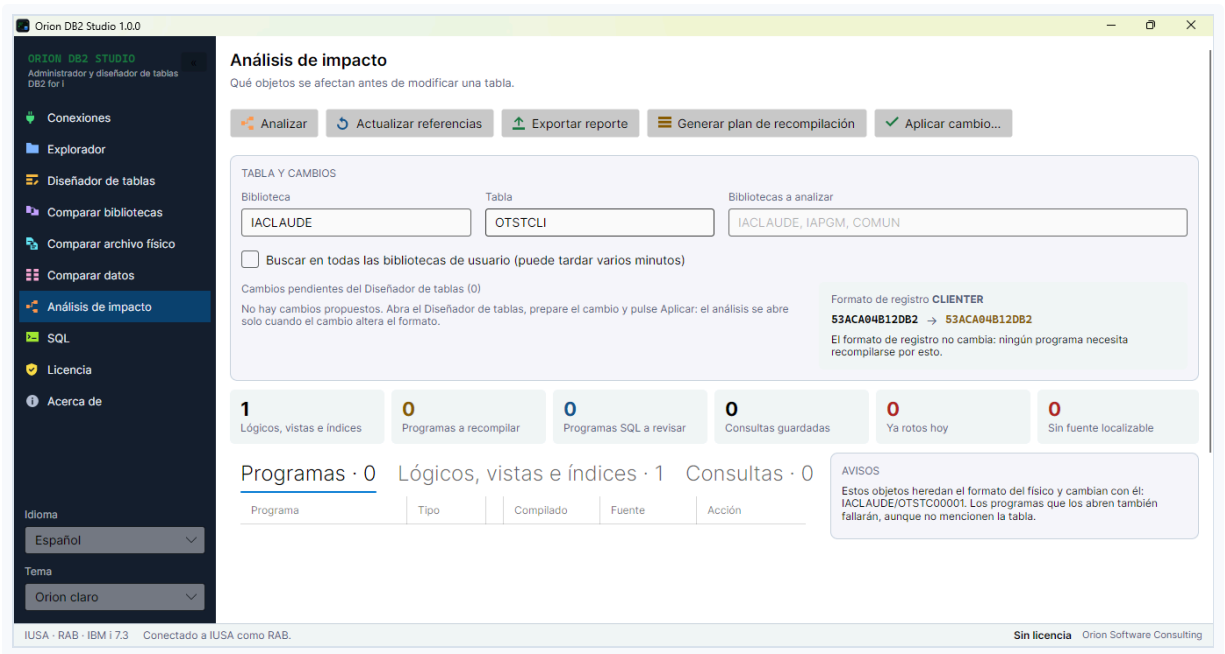
1. Indique la **biblioteca** y la **tabla**.
2. En **Bibliotecas a analizar**, las bibliotecas donde buscar programas, separadas por comas. O marque **Buscar en todas las bibliotecas de usuario**.
3. Pulse **Analizar**.

Mientras trabaja, un panel arriba enseña el spinner, la etapa, el contador de bibliotecas, la barra de avance, el tiempo transcurrido y el botón de **Cancelar**.

Todas las bibliotecas de usuario puede tardar varios minutos. La aplicación ejecuta un **DSPPGMREF** por biblioteca, y en un sistema con cientos de bibliotecas eso es mucho trabajo. Las referencias se guardan en una caché local, así que el segundo análisis del mismo día es mucho más rápido; **Actualizar referencias** la ignora y vuelve a leerlo todo.

Qué contesta

CAPTURA 17



El resultado: la huella actual y la nueva, los seis contadores y el detalle por pestañas.

Arriba, la huella del formato: la actual y la que quedaría tras el cambio. Si son iguales, ningún programa necesita recompilarse por esto.

Debajo, seis contadores y su detalle:

Lógicos, vistas e índices	Lo que cuelga de la tabla
Programas a recompilar	Los que se caerán con level check
Programas SQL a revisar	Los que usan SQL: no fallan por la huella, pero pueden referirse a una columna que cambió
Consultas guardadas	Las *QRYDFN de las bibliotecas analizadas
Ya rotos hoy	Programas cuya huella compilada ya no coincide con el archivo, antes de que usted toque nada
Sin fuente localizable	Los que no se pueden arreglar recompilando

Cómo sabe la huella nueva

No la calcula: la mide. El algoritmo del identificador de nivel de formato no es público, y adivinarlo sería peor que no intentarlo: una huella inventada que saliera igual a la actual diría «no hay que recompilar nada» justo cuando sí hay que hacerlo.

Lo que hace es **simular**: copia la estructura del archivo —sin datos— a una biblioteca temporal, le aplica el cambio y lee la huella del resultado con `DSPFD`. La biblioteca temporal se borra siempre, también si el análisis falla o si usted cancela.

Si el cambio es **destructivo**, el propio IBM i lo cancela desde una conexión cliente —manda un mensaje de consulta que nadie puede contestar— y la huella sale marcada como **estimada**. La aplicación lo dice en pantalla en vez de dar por buena una medición que no pudo hacer.

«Hereda el formato» tampoco se adivina

Un lógico que comparte el formato del físico se rompe igual que él, aunque ningún programa lo mencione. Para saber cuáles comparten formato, la aplicación **compara las huellas**: misma huella, mismo formato. No se deduce del DDS ni del `ADD ALL COLUMNS`.

El plan de recompilación

Generar plan de recompilación convierte el análisis en una lista de pasos en orden: respaldo, lógico sustituto si hace falta, el cambio de la tabla y las recompilaciones —primero los módulos, después los programas, después los programas de servicio—. El script sale en SQL y CL, editable antes de ejecutarlo, y se exporta a HTML.

La alternativa del lógico sustituto

Cuando hay programas sin fuente, recompilar no es una opción. Para esos casos la aplicación propone **renombrar el físico y crear un lógico con el nombre, el formato y la huella originales**: los programas de I/O nativo siguen funcionando sin tocarlos.

El lógico sustituto tiene un precio que conviene saber: los programas que **escriben** a través de él dejan las columnas nuevas con su valor por omisión, porque no las conocen.

El reporte del análisis se exporta a **HTML** y a **Excel**.

11. Análisis de aplicación

El análisis de impacto contesta por una tabla. Este módulo contesta por la **aplicación entera**: qué hay en la biblioteca, quién usa qué y cómo documentarlo. Entró en la versión 1.1.0 y vive en una sola entrada de la barra lateral, con siete pantallas dentro.

Todo sale de una instantánea

Ninguna de estas pantallas habla con el IBM i mientras se usa. Primero se **analiza** la biblioteca —una vez— y eso deja una instantánea fechada en una base local de su equipo; después se consulta tantas veces como haga falta, **sin conexión y sin cargar el sistema**. Cada pantalla dice siempre de qué fecha son los datos que enseña.

La recogida es de **solo lectura**: consultas a los catálogos y mandatos `DSPxxx` que escriben a un archivo temporal de `QTEMP`, que muere con el trabajo. Funciona con una conexión marcada como de solo lectura y con la licencia vencida.

Analizar bibliotecas

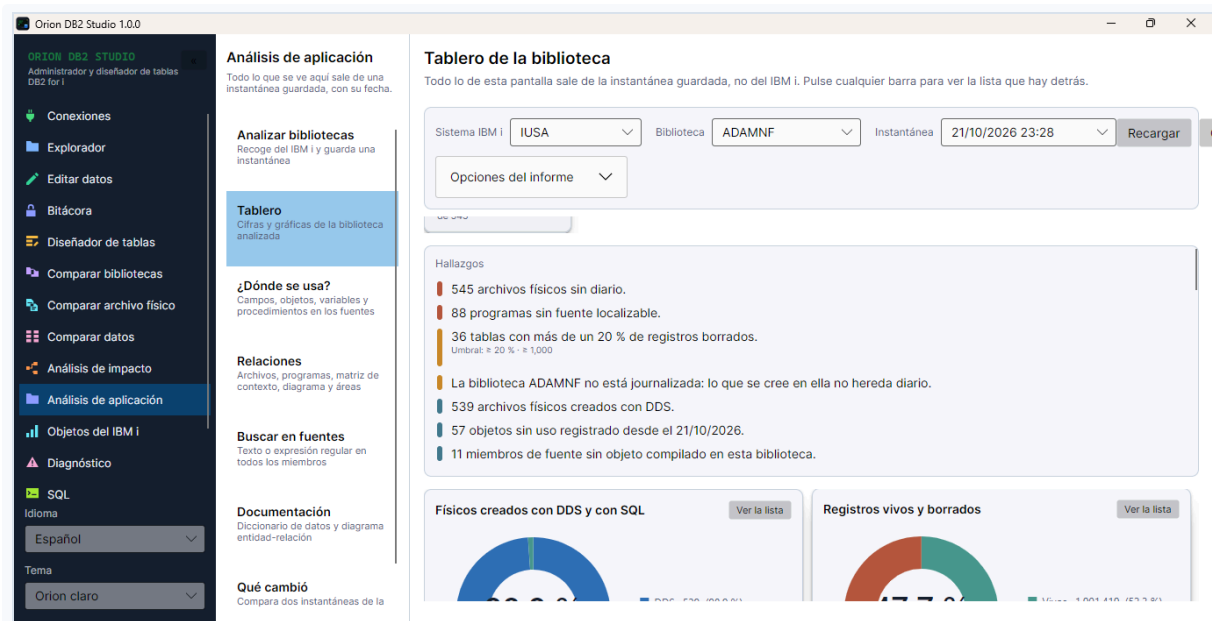
Se elige la biblioteca y qué partes se recogen. Lo que más tarda con diferencia son los **fuentes**: medido contra un IBM i 7.3, leer y analizar un miembro cuesta unos 436 ms, así que una biblioteca con 2.000 miembros son unos quince minutos la primera vez. A partir de la segunda solo se bajan los miembros que cambiaron, y volver a pasar por uno que no cambió cuesta 52 ms.

Inventario	Objetos de la biblioteca, con tamaño, fechas y uso
Archivos	Formatos, campos, llaves, índices, restricciones y disparadores
Miembros	Los de todos los archivos, con sus fechas
Referencias	Qué objeto usa cada programa, del objeto compilado
Relaciones	Dependencias entre archivos, de <code>DSPDBR</code>
Fuentes	El texto de los miembros y su análisis sintáctico

Tablero

Las cifras de la biblioteca en tarjetas, una lista de **hallazgos** y dieciséis gráficas. Los hallazgos son lo que hay que mirar: archivos físicos sin diario, programas sin fuente localizable, tablas con demasiado espacio muerto, objetos sin uso registrado. Cada gráfica se pulsa y abre la lista que hay detrás, que se exporta a CSV.

CAPTURA 18



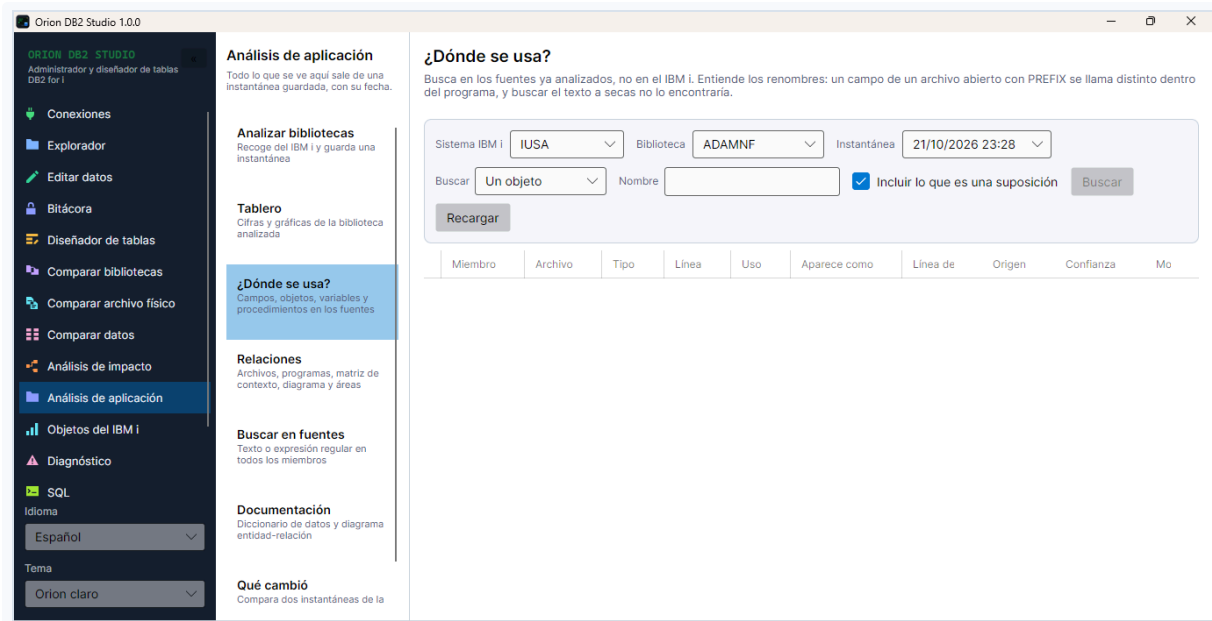
El tablero de una biblioteca: las cifras arriba y, debajo, los hallazgos.

«Generar informe PDF» deja el tablero entero en un documento con portada, resumen ejecutivo, una sección por gráfica con su tabla y sus anexos. Se puede pedir en español o en inglés, con el logotipo de su empresa en la portada.

¿Dónde se usa?

La pregunta que en un IBM i nadie contesta de memoria. Se busca un **campo**, un **objeto**, una **variable** o un **procedimiento**, y la respuesta trae el miembro, la línea, el texto de esa línea y si el nombre se lee o se escribe.

CAPTURA 19



La búsqueda: se elige qué se busca y de qué archivo es el campo.

Lee **RPG IV** —fijo, libre y mezclado—, **RPG III**, **CL** y **COBOL**, con el SQL embebido de todos ellos. Y entiende los renombres, que es lo que lo separa de buscar el texto a secas: un archivo abierto con **PREFIX(CL_)** hace que el campo **CLNOM** se llame **CL_CLNOM** dentro de ese programa, y buscar «CLNOM» no encontraría nada. La pantalla enseña con qué nombres buscó.

Cada resultado lleva su **confianza**. Un **CHAIN** sobre un archivo es un hecho; que un nombre suelto en una expresión sea ese campo y no una variable que se llama igual es una suposición, y sale marcada como tal. La casilla «incluir lo que es una suposición» las deja fuera.

Los miembros que **no se pudieron analizar** salen en su propia lista. «No aparece» y «no se miró» no son lo mismo, y la diferencia la paga quien cambie el campo creyendo que nadie más lo usa.

Relaciones, matriz y diagrama

- **Archivos por programa:** qué abre cada uno, con qué formato y qué le hace. Junta lo que dice el objeto compilado con lo que dice el fuente, que además ve el SQL embebido.
- **Campos por archivo:** atributos, textos, encabezados, llaves y en cuántos programas se usa cada campo.
- **Matriz de contexto:** archivos en las filas, programas en las columnas, exportable a Excel.
- **Diagrama:** jerarquía de archivos, árbol de llamadas o entidad-relación, con zoom y arrastre, y exportación a **SVG** y **PNG**.
- **Áreas de aplicación:** agrupar los objetos por prefijo o por lista para filtrar el tablero y el informe.

Buscar en los fuentes

El **grep** de la biblioteca: texto o expresión regular sobre todos los miembros, con mayúsculas, palabra completa, líneas de contexto y la opción de saltarse los comentarios. Se puede buscar en **todas las bibliotecas analizadas** a la vez y filtrar por archivo fuente. Los resultados se exportan a CSV.

Documentación: el diccionario de datos

El documento que piden una auditoría o un traspaso y que casi nadie tiene escrito: archivos, campos con su tipo y su texto, llaves, índices, restricciones, disparadores y los programas que usan cada archivo, con el **diagrama entidad-relación**.

CAPTURA 20

Las opciones del diccionario: formato, secciones y diagrama.

HTML	Autocontenido, con el diagrama dentro; para la intranet
PDF	Para entregar o archivar
Word	Para editarlo y añadirle lo que la herramienta no sabe
Excel	Una hoja por clase, para filtrar y ordenar

Las relaciones entre tablas que **no** son llaves foráneas declaradas son **deducidas** por coincidencia de nombre y tipo de campo. En una base creada con DDS no hay foráneas, así que casi todas lo serán. El documento las marca en los cuatro formatos —línea punteada en el diagrama, columna «¿es un hecho?» en el Excel— y conviene leerlas con eso presente.

Qué cambió

Compara dos instantáneas de la misma biblioteca y dice qué se creó, qué se borró y qué cambió entre las dos fechas, destacando los cambios de formato, que son los que obligan a recompilar.

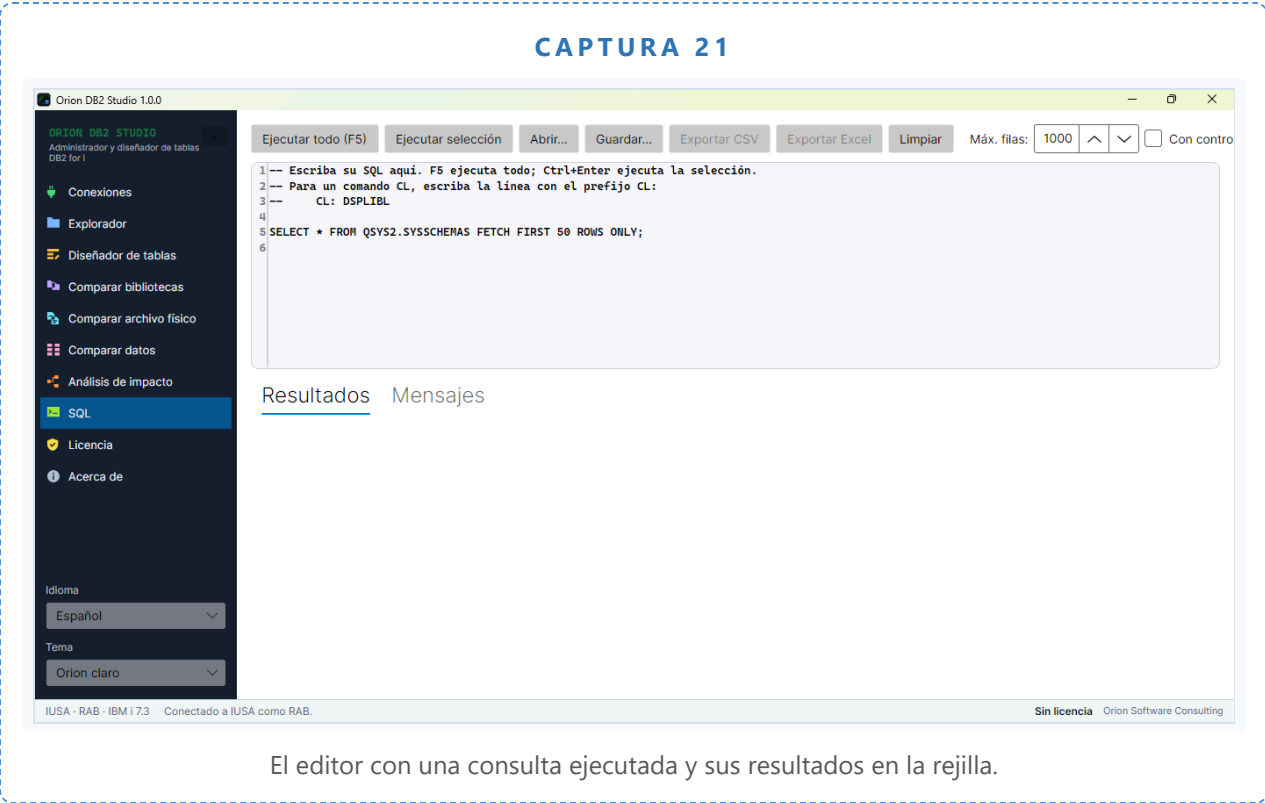
Desde la línea de mandatos

```
orion-db2 analyze    --conn IUSA --libraries MILIB --with-source
orion-db2 dashboard  --system IUSA --library MILIB --pdf analisis.pdf
orion-db2 where-used  --system IUSA --library MILIB --name CLNOM --file CLIENTES
orion-db2 doc         --system IUSA --library MILIB --out diccionario.html
```

Ninguno de los cuatro escribe en el IBM i. `dashboard`, `where-used` y `doc` ni siquiera se conectan si no se les pide `--refresh`: trabajan con la última instantánea guardada, así que se pueden dejar en una tarea nocturna que no depende de que el sistema esté levantado.

12. El editor SQL

Para lo que no cabe en una pantalla: consultas sueltas, sentencias de mantenimiento y mandatos CL, contra la conexión activa.



Ejecutar

Ejecutar todo (F5)	Todo el contenido del editor, separando por ;
Ejecutar selección	Solo lo que esté seleccionado, o todo si no hay selección (Ctrl+Enter)
Máx. filas	Tope de filas que se traen a la rejilla
Con control de compromiso	Envuelve los cambios en una transacción y deshace si algo falla

Mandatos CL

Una línea que empiece por **CL:** se ejecuta como mandato del sistema a través de **QCMDEXC**, con la misma pantalla de confirmación que el resto de los cambios. Por ejemplo:

```
CL: CHGJOB INQMSGRPY(*SYSRPYL)
```

Resultados y mensajes

La pestaña **Resultados** trae la rejilla; **Mensajes** lleva el registro de lo ejecutado, con el tiempo de cada sentencia y las filas afectadas. Lo que haya en la rejilla se exporta a **CSV** y a **Excel**.

Dos límites que conviene conocer. El separador parte las sentencias por `;`, así que **no se pueden crear procedimientos, funciones ni triggers** que lleven dentro un bloque `BEGIN ... END`: para eso, guarde el script y ejecútelo con `RUNSQLSTM`. Y «Máx. filas» es un tope duro, no una paginación: si el resultado trae más, no las verá.

El editor todavía **no autocompleta** nombres de bibliotecas, tablas ni columnas. Resalta la sintaxis y ejecuta; el autocompletado está previsto para una versión próxima.

13. Seguridad: lo que no se hace sin confirmar

La aplicación trabaja contra bases de datos de producción. Lo que sigue no son opciones que se puedan desactivar en preferencias: están en el camino.

La pantalla de confirmación

Antes de cualquier `CREATE`, `ALTER`, `DROP`, `DELETE`, `UPDATE` o mandato CL, la aplicación enseña una pantalla con:

- El **sistema** y el **usuario** con que se va a ejecutar.
- La **biblioteca** afectada.
- **Cuántas sentencias** se van a ejecutar.
- El **riesgo** del script.
- El **script completo**, para leerlo antes de firmarlo.

Aparece en los cinco caminos que escriben: el diseñador, la sincronización de bibliotecas, la de datos, el plan de impacto y el editor SQL.

Conexiones de producción

Una conexión marcada como **PRODUCCIÓN** cambia dos cosas: la aplicación enseña una **franja roja permanente** con el sistema y la biblioteca, y la confirmación exige **escribir el nombre de la biblioteca** para habilitar el botón de ejecutar.

Cambios que pierden datos

Quitar una columna, acortar un campo, bajar los decimales o cambiar a un tipo incompatible se marcan como **pérdida de datos**: salen en rojo, exigen la misma confirmación escrita y el botón cambia de texto a «Ejecutar de todos modos».

Modo solo lectura

Una conexión marcada como **Solo lectura** rechaza toda sentencia que modifique y todo mandato CL **en la capa de datos**, no solo ocultando botones. Es la forma de dar la herramienta a alguien que solo tiene que mirar.

Las contraseñas

Se guardan cifradas con el sistema operativo —DPAPI en Windows, ligado a su cuenta de usuario— y nunca aparecen en texto plano en el archivo de configuración ni en los registros.

Respaldos

Antes de modificar una tabla existente, el diseñador **ofrece guardar el DDL actual** del objeto. El plan de impacto va más lejos: incluye un respaldo de los datos con `CREATE TABLE ... AS` en la biblioteca que usted indique.

Y en la línea de mandatos

El ejecutable `orion-db2` **nunca ejecuta cambios** salvo que se lo pida con `--apply --yes`, y **se niega** a aplicarlos contra una conexión marcada como producción, remitiendo a la aplicación de escritorio.

14. Reportes y línea de mandatos

Reportes

Reporte	Formatos	Desde dónde
Comparación de bibliotecas	HTML y PDF	Comparar bibliotecas
Comparación de archivo físico	PDF	Comparar archivo físico
Comparación de datos	PDF y CSV	Comparar datos
Análisis de impacto	HTML y Excel	Análisis de impacto
Plan de recompilación	HTML	Análisis de impacto
Diccionario de datos	HTML	Solo por línea de mandatos
Resultado de una consulta	CSV y Excel	Editor SQL

Los HTML son **autocontenidos**: un solo archivo que se puede enviar por correo y se abre en cualquier navegador.

El ejecutable orion-db2

Para automatizar desde un script, un job programado o una canalización. Lleva los mismos perfiles de conexión que la aplicación de escritorio.

ddl — el DDL de una biblioteca

```
orion-db2 ddl --conn IUSA --library IACLAUDE --out C:\ddl\
```

Genera un archivo por objeto. Con `--tipos` se limita a ciertos tipos y con `--archivo-unico` sale todo junto.

compare — comparar dos bibliotecas

```
orion-db2 compare --source IUSA:DESARROLLO --target IUSA:PRODUCCION \  
--script sincronizar.sql --report informe.html
```

El origen puede ser una instantánea `.json` en vez de un sistema en vivo. Códigos de salida: **0** si son iguales, **1** si hay diferencias y **2** si hubo error, para encadenarlo en un script.

impact — el análisis de impacto

```
orion-db2 impact --conn IUSA --table IACLAUDE/OTSTCLI \  
                --change "ADD COLUMN CLIEMA CHAR(60)" --report impacto.html
```

Solo lectura: lo único que crea es la biblioteca temporal de la simulación, y la borra siempre. Salidas: **0** sin impacto, **1** hay que recompilar, **2** hay bloqueantes o programas ya rotos, **3** error.

export — datos y diccionario

```
orion-db2 export --conn IUSA --sql "SELECT * FROM IACLAUDE.OTSTCLI" \  
                --out clientes.xlsx
```

Con `--diccionario` en vez de `--sql` saca el diccionario de datos de una biblioteca en HTML.

15. Solución de problemas

No conecta y dice que se agotó el tiempo de espera

El sistema está apagado, no hay ruta hasta él, o un cortafuegos descarta el tráfico sin contestar. Haga ping al sistema y compruebe la VPN.

Dice que el sistema rechazó la conexión

El IBM i contestó, pero no hay nadie escuchando en el puerto. Arranque los host servers con `STRHOSTSVR SERVER(*ALL)` y compruebe con `NETSTAT *CNN` que escuchan los puertos 8470, 8471 y 8476.

Dice que el sistema no tiene TLS configurado

Los host servers están arriba y contestan sin cifrar, pero el puerto seguro está cerrado. Desmarque **Usar TLS**, o pida al administrador que configure SSL en el Digital Certificate Manager y reinicie los servicios.

El perfil de usuario se deshabilitó

El IBM i lo hace tras varios intentos fallidos, según `QMAXSIGN`. Pida `CHGUSRPRF USRPRF(SUPERFIL) STATUS(*ENABLED)`.

Un ALTER no se aplica y el error habla de SQL0952

El sistema envió un mensaje de consulta —**CPA32B2**, «cambiar el archivo puede provocar que se pierdan datos»— y el trabajo del servidor lo contestó con «cancelar», que es su respuesta por omisión. No es un fallo de la aplicación ni un tiempo agotado. Tres salidas:

- Hacer el cambio en dos pasos: crear la tabla nueva, copiar con `INSERT ... SELECT` y renombrar.
- Ejecutar el script desde una sesión 5250 y responder I.
- Pedir al administrador la respuesta automática del sistema (`ADDRPYLE` con `MSGID(CPA32B2) RPY(I)`) y ejecutar antes, en la misma conexión, `CL: CHGJOB INQMSGRPY(*SYSRPYL)`.

Un UPDATE o un DELETE fallan con SQL7008

La tabla no está registrada por diario y la operación lo exige. Hace falta un diario en la biblioteca y después `STRJRNPF`. Las bibliotecas creadas con `CREATE SCHEMA` ya traen el suyo; las creadas con `CRTLIB`, no.

La sesión se cae sola tras un rato sin usarla

Es la VPN o el cortafuegos cerrando conexiones ociosas. La aplicación lo detecta y avisa; vuelva a pulsar **Conectar**.

Los acentos salen mal

Es el CCSID. Déjelo vacío en el perfil para que use el del perfil de usuario del IBM i, o póngale el que corresponda a su sistema —284 para España y buena parte de Latinoamérica—.

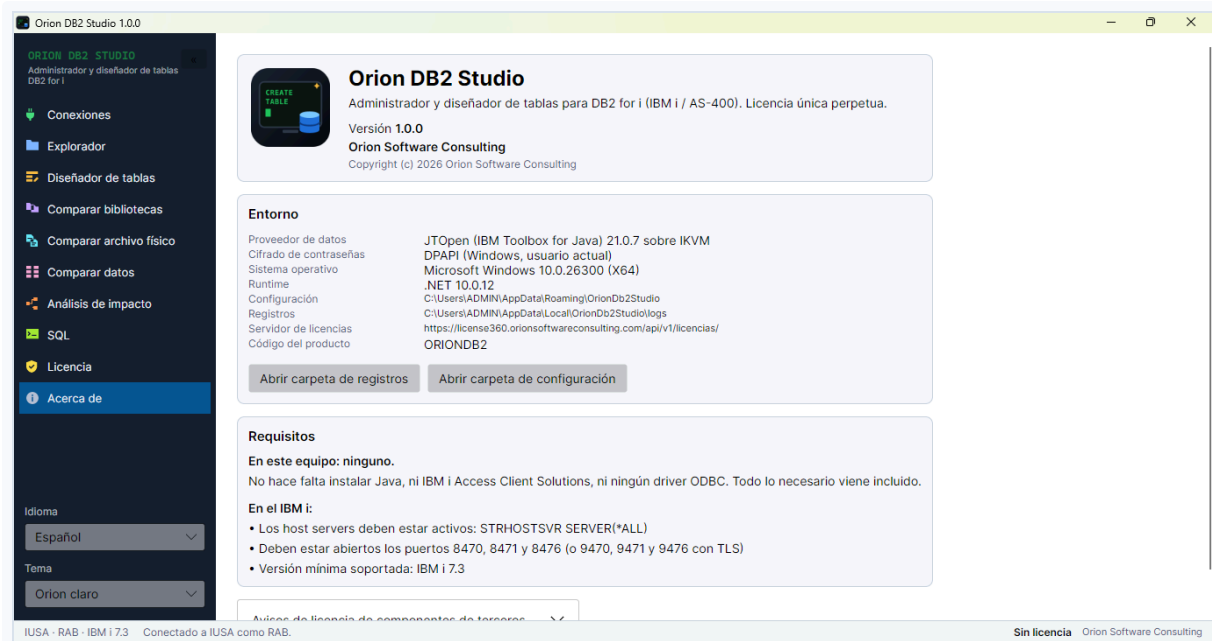
El análisis de impacto no encontró el fuente de un programa

La aplicación busca el miembro fuente en la biblioteca y el archivo que el propio objeto declara. Hoy no busca en el IFS ni compara el contenido del miembro con el objeto compilado: si el fuente se movió, aparecerá como no localizable.

¿Dónde están los registros de la aplicación?

En la carpeta de registros del usuario, que se abre desde **Acerca de**. Guarda catorce días. Si va a reportar un problema, el archivo del día es lo primero que hay que adjuntar.

CAPTURA 22



«Acerca de»: la versión, el entorno de ejecución, los requisitos y los accesos a las carpetas de configuración y de registros.

16. Apéndice: lista de capturas

Las 22 capturas de este manual, en orden. Todas están tomadas contra un IBM i real con la biblioteca de pruebas `IACLAUDE`.

Figura	Qué enseña
CAPTURA 01	La pantalla de Conexiones, que es por donde se empieza. A la izquierda la barra lateral con las nueve secciones; abajo, los selectores de idioma y de tema.
CAPTURA 02	La pantalla de Licencia: el estado actual, los datos del titular, el botón de la prueba gratuita, el cuadro para activar una clave comprada y el identificador de este equipo.
CAPTURA 03	«Probar conexión» contra un IBM i 7.3: el aviso con el resultado y el mismo mensaje en la franja de arriba.
CAPTURA 04	Conectado: el mensaje en la franja superior y, abajo a la izquierda, el sistema, el usuario y la versión de IBM i.
CAPTURA 05	El explorador tras cargar el filtro: las bibliotecas que coinciden, cada una con su texto descriptivo.
CAPTURA 06	El árbol abierto: biblioteca, familias de objetos y, dentro de Tablas, los archivos con su distintivo DDS.
CAPTURA 07	Propiedades de una tabla: lo que hay que mirar antes de tocarla.
CAPTURA 08	Las columnas de un archivo físico creado con DDS, con la llave señalada en la primera fila.
CAPTURA 09	El panel de la derecha, con las pestañas Propiedades, Columnas, DDL y Datos.
CAPTURA 10	Las primeras filas de un archivo, dentro del propio explorador.
CAPTURA 11	El diseñador con las columnas de auditoría recién añadidas: usuario y fecha de creación y de modificación, con sus valores por omisión.
CAPTURA 12	La pestaña de llave primaria: se marcan las columnas que la forman y el orden de cada una.
CAPTURA 13	La pantalla de comparación: origen, destino, opciones y la rejilla de resultados.
CAPTURA 14	Comparación de archivo físico: el archivo, las dos bibliotecas y el resultado por campos, llaves y atributos.
CAPTURA 15	Los parámetros de la comparación de datos: archivo, miembro, cómo emparejar los registros y qué campos excluir.
CAPTURA 16	La pantalla de análisis antes de empezar: la tabla, las bibliotecas donde buscar y los cambios que llegan del diseñador.
CAPTURA 17	El resultado: la huella actual y la nueva, los seis contadores y el detalle por pestañas.

Figura	Qué enseña
CAPTURA 18	El tablero de una biblioteca: las cifras arriba y, debajo, los hallazgos.
CAPTURA 19	La búsqueda: se elige qué se busca y de qué archivo es el campo.
CAPTURA 20	Las opciones del diccionario: formato, secciones y diagrama.
CAPTURA 21	El editor con una consulta ejecutada y sus resultados en la rejilla.
CAPTURA 22	«Acerca de»: la versión, el entorno de ejecución, los requisitos y los accesos a las carpetas de configuración y de registros.

English version

El manual completo en inglés empieza en la página siguiente.



ORION
SOFTWARE
CONSULTING

Orion DB2 Studio

User manual

Version 1.1.0 • .NET 10

Contents

1. What it is and what you need
2. Installation
3. Licence and 30-day trial
4. Connecting to an IBM i
5. The object browser
6. The table designer
7. Comparing libraries
8. Comparing a physical file
9. Comparing data
10. Impact analysis
11. Application analysis
12. The SQL editor
13. Security: nothing happens without confirmation
14. Reports and command line
15. Troubleshooting
16. Appendix: list of screenshots

1. What it is and what you need

Orion DB2 Studio is a table manager and designer for **DB2 for i**, the database built into IBM i systems (AS/400, iSeries, IBM Power). You use it to see what a library holds, to create and change tables without writing the SQL by hand, to compare two libraries, and to find out which programs break before you touch a file.

It connects **straight to the IBM i host servers**. There is no need to install Java, IBM i Access Client Solutions or ODBC drivers: the driver travels inside the program.

About the screenshots. They are the same pictures as the Spanish half of this manual, so the menus and buttons in them read in Spanish. The product itself switches to English from **Preferences**, and this text names the English wording of what you see in them.

Platforms

Desktop	Windows 10 and 11, 64-bit, and Linux, 64-bit
Mobile	Android 7.0 or later, in design mode: no connection to the IBM i
Platform	.NET 10, with the interface in Avalonia
Languages	Spanish and English, both switchable without restarting

What it needs on the system side

An IBM i **7.3 or later** reachable over the network, with the host servers running and a user profile enabled. The application **installs nothing on the IBM i**: it leans on the services that IBM i Access Client Solutions already uses.

Minimum release	IBM i 7.3
Services	Host servers started with <code>STRHOSTSVR SERVER(*ALL)</code>
Ports	8470, 8471 and 8476 in the clear; 9470, 9471 and 9476 with TLS
Authority	The profile needs at least <code>*USE</code> over the libraries and objects it is going to read

The release is detected on connecting, by reading `SYSIBMADM.ENV_SYS_INFO`. With it the application disables whatever that release does not support: the `BOOLEAN` type, for instance, is only offered from 7.5 onwards.

What it does, as a list

- **Browser:** libraries, tables, views, indexes, procedures, functions, triggers and sequences, with their properties, their columns, their DDL and their data.
- **Table designer:** create and change tables with the DDL being written alongside and validation as you type.
- **Reverse engineering:** the real DDL of any object through `QSYS2.GENERATE_SQL`, including files created with DDS.
- **Compare libraries:** development against production, with the DDL diff and the synchronisation script.
- **Compare physical file:** two files field by field, read from the compiled object.
- **Compare data:** the records of two libraries, compared inside the IBM i itself.
- **Impact analysis:** which programs, logical files, views and queries stop working when a table changes.
- **Application analysis:** the library dashboard, «where is it used?» across RPG, CL and COBOL source, the relations, the diagram and the data dictionary.
- **SQL editor:** queries and CL commands, with the results in a grid and export to Excel and CSV.
- **Record editor:** view and edit the data in a file, with concurrency control and an audit log.
- **Reports and command line:** HTML, PDF and Excel, and the `orion-db2` executable for automation.

What arrived in version 1.1.0 —the application analysis and the record editor— is described in chapter 11. The record editor, the audit log, the recovery of deleted records and the diagnostic tools have screens of their own and **no chapter in this manual yet**: they will be documented in the next edition.

2. Installation

Requirements

System	Windows 10 or 11, 64-bit, or Linux, 64-bit
Space	Around 150 MB once installed
Rights	Local administrator: the installer writes into Program Files
Other software	None. No Java, no ACS, no ODBC

The size comes from the driver: the application carries JTOpen —IBM's client for talking to the IBM i— compiled to .NET. That is the price of not asking the customer to install Java.

Download

Go to orionsoftwareconsulting.com/Descargas and download the installer for your system. For Windows there are two formats with the same content:

.MSI	The installer. Double-click, and it lands in the Start menu
Portable	The application on its own in a .zip. No administrator needed: unzip it and run it

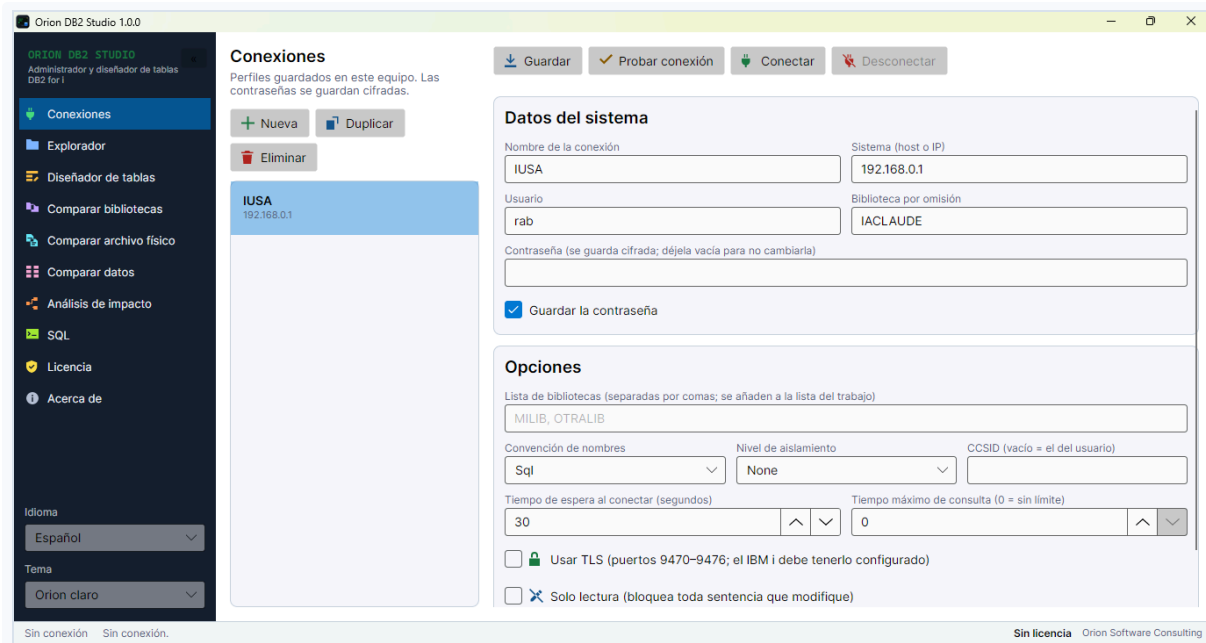
For Linux there are three packages: [.deb](#) for Debian, Ubuntu, Kali and Mint; [.rpm](#) for Fedora, RHEL, Rocky, Alma and openSUSE; and a portable [.tar.gz](#) that needs no root.

Windows will warn about an «unknown publisher» when you open the installer: it is not signed with a certificate yet. That is not a sign that the file has been tampered with. To check, the download page publishes the **SHA-256** of every file; compare it with the one you just downloaded before running it.

First start

The application opens on the **Connections** screen, with none configured, and the status bar warning that there is no licence. Both are sorted out in the next two chapters.

FIGURE 01



The Connections screen, which is where you start. On the left the sidebar with its sections; at the bottom, the language and theme selectors.

Language and theme

At the end of the sidebar there are two lists. **Language** switches the whole interface between Spanish and English without restarting. **Theme** offers five: Orion dark, Orion light, Green screen, Amber and High contrast. All five are measured so that text meets the contrast required by WCAG 2.1 AA, and the one you pick is remembered for next time.

The figures in this manual are taken with the **Orion light** theme, which is the one that prints best. What you see on your screen may use other colours without anything having moved.

3. Licence and 30-day trial

Orion DB2 Studio is sold with a **single perpetual licence**: you pay once and it does not expire. There is one edition, with every feature, and each licence covers **two machines**.

Launch price	USD 22.99
Machines	Up to 2 per licence
Trial	30 days with every feature
When it expires	Read-only mode: browse, view the DDL and generate reports

Where the status shows

In the **Licence** section, and at a glance in the right-hand corner of the status bar, which follows the application through every screen.

FIGURE 02

Orion DB2 Studio 1.0.0

Licencia

Orion DB2 Studio se vende con licencia única perpetua. No hay suscripción.

Sin licencia en este equipo

Este equipo no tiene ninguna licencia activada.

Modo solo lectura: puede explorar objetos, ver su DDL y generar reportes, pero no ejecutar cambios ni comandos CL.

Actualizar estado Revalidar contra el servidor Liberar este equipo

Sus datos

Hacen falta para dar de alta la prueba y para activar una clave comprada. Es lo que queda registrado como titular de la licencia.

Correo

usted@empresa.com

Nombre

Nombre y apellidos

Empresa (opcional)

Probar antes de comprar

La prueba gratuita dura 30 días con todas las funciones. Se pide una sola vez por equipo y hace falta conexión a internet para darla de alta.

Solicitar la prueba gratuita

IUSA - RAB - IBM i 7.3 Conectado a IUSA como RAB. Sin licencia Orion Software Consulting

The Licence screen: current status, the holder's details, the free trial button, the box for activating a purchased key and this machine's identifier.

Asking for the trial

1. Fill in **Email** and **Name**. The email is where the key will arrive if you buy later.

2. Press **Request the free trial**.

3. The application talks to the Orion licence server and leaves the trial activated on this machine.

The trial is granted **once per machine** and **needs an internet connection** at that moment. Afterwards you can work offline: the application revalidates now and then, not on every start.

Activating a purchased key

Type the key into the **Activate** box and press the button. The activation is tied to an identifier for this machine, computed from the operating system itself, which **contains no personal data**. You can see it, and save it to a file, at the bottom of the same screen.

To move the licence to another machine —a new laptop, a disk change— use **Release this machine**: it frees the activation and leaves the slot open for another one.

What happens when the trial ends. The application still opens and nothing is lost: you can browse objects, view their DDL and generate reports. What is blocked is **running changes** and **issuing CL commands**. The status bar says so, and every screen that needs it explains why.

4. Connecting to an IBM i

A **connection** is a profile saved on your machine: the system, the user, the encrypted password and the session options. You can have as many as you administer systems, and the password is never stored in clear text.

Creating the first profile

1. Press **New**.
2. Fill in **Connection name** —the one you will see, for instance «Production»—, **System (host or IP)**, **User** and **Password**.
3. Set the **Default library**, which is the one the screens start from.
4. Press **Save**.

Profile fields

Connection name	What you call it. It never travels to the system
System (host or IP)	The network name of the IBM i, or its IP address
User and password	The IBM i user profile. The password is encrypted by the operating system before being stored
Default library	The one the browser and the SQL editor start from
Save the password	If you clear it, the application asks for it on every connection

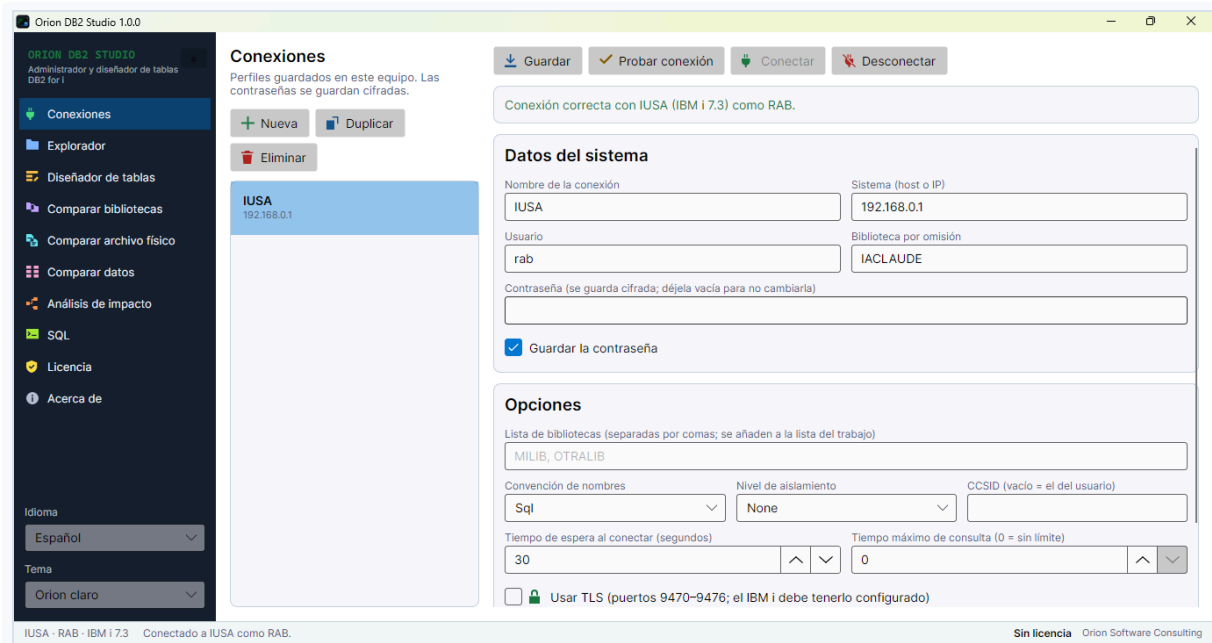
Options

Library list	Added to the job's library list, separated by commas
Naming convention	Sql uses the dot (<code>LIBRARY.TABLE</code>); System uses the slash (<code>LIBRARY/TABLE</code>)
Isolation level	The commitment control for the session
CCSID	Character set. Empty = the one from the user profile
Connection timeout	Seconds before giving up
Query timeout	Seconds per statement. 0 = no limit
Use TLS	Encrypts the connection. The IBM i must have SSL configured
Read only	Blocks every modifying statement and every CL command, in the data layer itself
PRODUCTION	A permanent red band and a typed confirmation before every change

Testing before connecting

Test connection checks that the system can be reached and answers in a dialog with the result. If it works, it says which system it talked to, what IBM i release it runs and which user it signed on as.

FIGURE 03



«Test connection» against an IBM i 7.3: the dialog with the result and the same message in the band above.

If it fails, the message says **why** and **what to do**, instead of a generic error. It tells these cases apart:

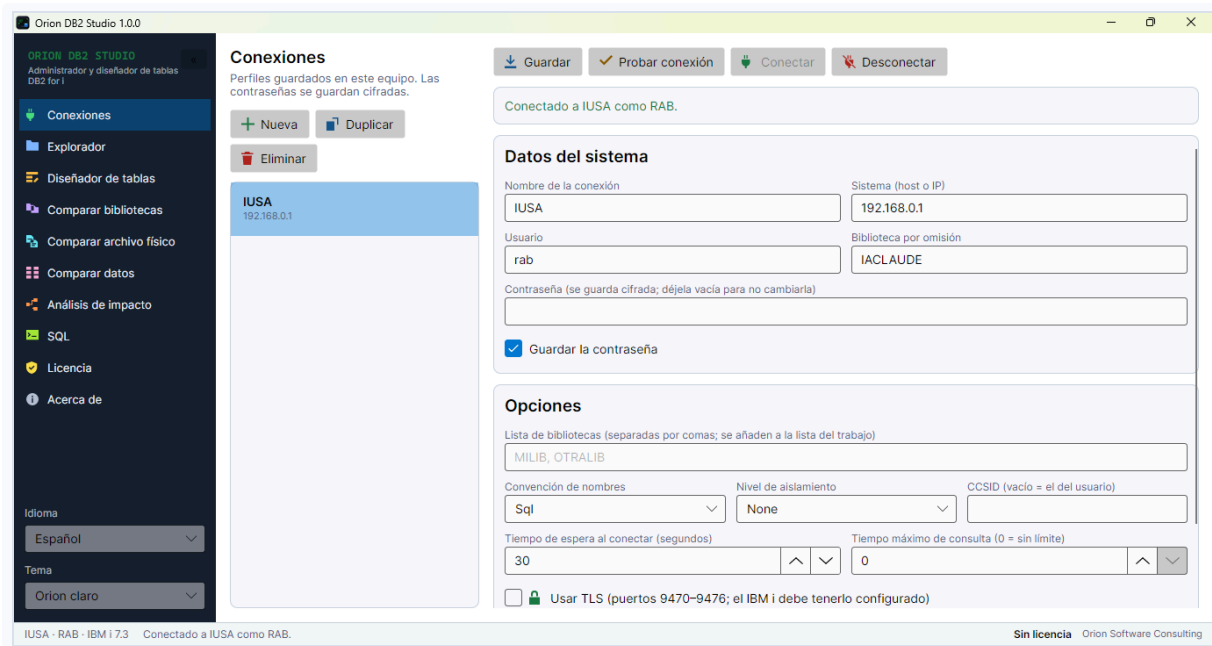
What happened	What it tells you
The name does not resolve	To check the name or use the IP address
The system does not answer	That it may be down, or that a firewall is dropping the traffic
The system refuses the connection	To run <code>STRHOSTSVR SERVER(*ALL)</code> , because the host servers are not up
You asked for TLS and the system has none	To turn TLS off, or to configure SSL on the IBM i. It does not tell you to run STRHOSTSVR, because the service is up
The system only accepts TLS	To tick the TLS box
Wrong user or password	To check it, and it reminds you that the IBM i disables the profile after several attempts
Profile disabled	To ask for <code>CHGUSRPRF STATUS(*ENABLED)</code>
Password expired	To sign on through 5250 or ACS and change it

Before calling the driver, the application **probes the database port**. That is why a system that is down is detected in a few seconds instead of waiting out the whole timeout.

Connecting

Connect opens the session and leaves it active for the rest of the screens. The status bar changes to show the system, the user and the release.

FIGURE 04



Connected: the message in the top band and, bottom left, the system, the user and the IBM i release.

If the connection is marked as **PRODUCTION**, the application asks for confirmation when connecting and then shows a permanent red band at the top, with the system and the library, which cannot be dismissed while that session is open.

If the connection drops on its own

Every twenty seconds the application checks that the session is still alive. If the IBM i, the VPN or the firewall cut it—which is common after a while with no activity—it says so in a dialog, closes the session and makes it plain in the status bar. Press **Connect** again.

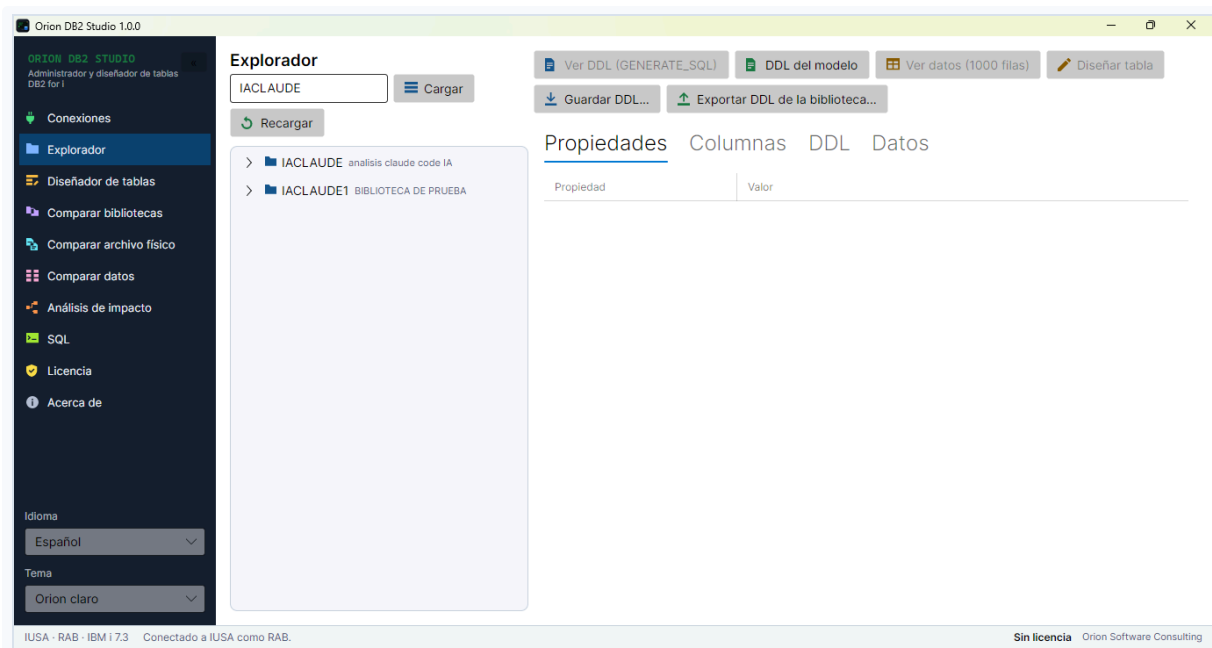
5. The object browser

The **Browser** is the tree of what the system holds: libraries and, inside each one, tables, views, indexes, procedures, functions, triggers and sequences. It loads in pieces, as you open each node, so that a whole library you may not need is not pulled across.

Loading a library

Type the name —or part of it— in the box at the top and press **Load**. A generic name works: **IA** brings back everything starting with IA.

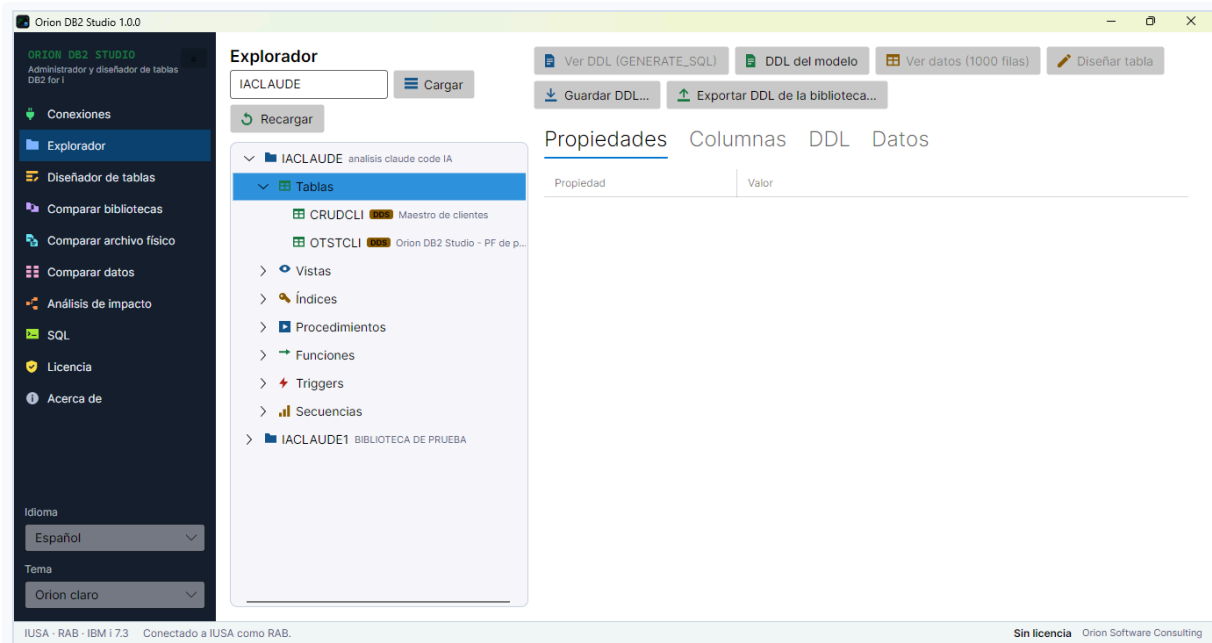
FIGURE 05



The browser after loading the filter: the libraries that match, each with its descriptive text.

Open a library and you see the seven families of objects. Open a family and you see its objects, each with its name, its text and —when it differs from the SQL name— its system name.

FIGURE 06



The tree open: library, object families and, inside Tables, the files with their DDS badge.

The DDS badge marks files created with DDS rather than with SQL. It is neither a fault nor a warning: it is information that changes what can be done with them, and the application treats them exactly like the rest.

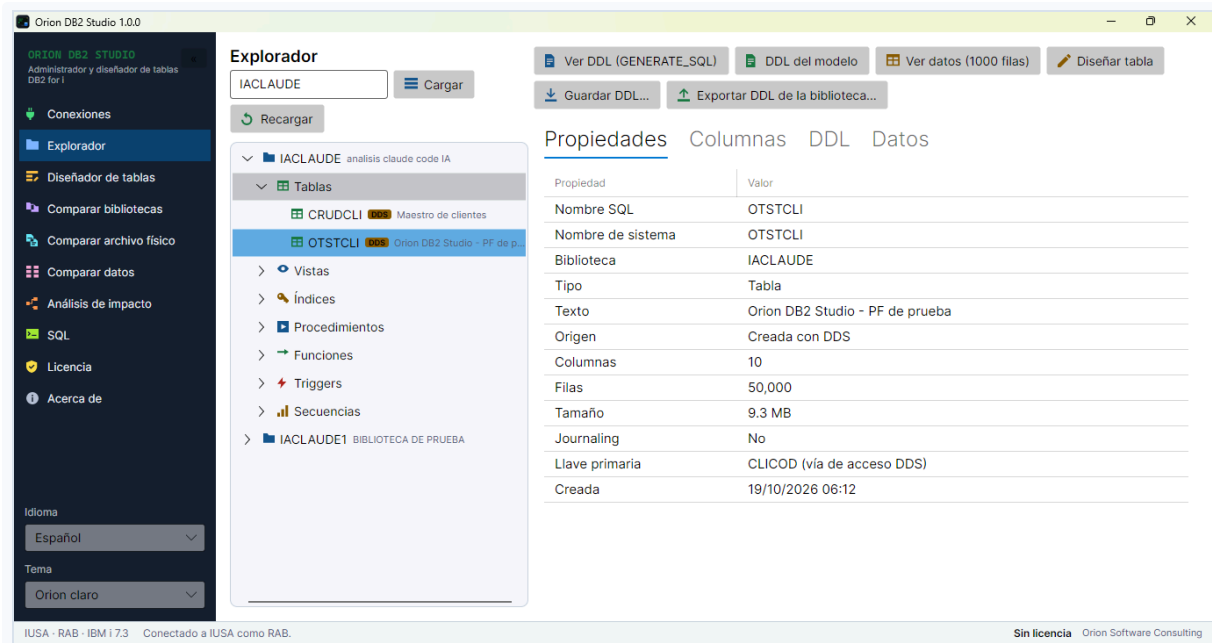
The four tabs

When you select an object, the panel on the right describes it in four tabs.

Properties

SQL name and system name, library, type, text, number of columns and rows, size, whether it is **journalled**, the primary key, the creation date and whether the object was created with SQL or with DDS.

FIGURE 07



Properties of a table: what to look at before touching it.

Columns

One row per column, with the SQL name, the system name, the type, the length, the decimals, the CCSID and whether it allows nulls. The primary key is marked with an icon on the left.

FIGURE 08

Orion DB2 Studio 1.0.0

ORION DB2 STUDIO
Administrador y diseñador de tablas DB2 for i

Conexiones

Explorador

Diseñador de tablas

Comparar bibliotecas

Comparar archivo físico

Comparar datos

Análisis de impacto

SQL

Licencia

Acercar de

Idioma
Español

Tema
Orion claro

Explorador

IACLAUDE

Cargar

Recargar

IACLAUDE analisis claude code IA

Tablas

CRUDCLI Maestro de clientes

OTSTCLI Orion DB2 Studio - PF de p...

Vistas

Índices

Procedimientos

Funciones

Triggers

Secuencias

IACLAUDE1 BIBLIOTECA DE PRUEBA

Ver DDL (GENERATE_SQL)

DDL del modelo

Ver datos (1000 filas)

Diseñar tabla

Guardar DDL...

Exportar DDL de la biblioteca...

Propiedades Columnas DDL Datos

	Columna	Sistema	Tipo	Longit	D	CCS	Nu
1	CLICOD	CLICOD	DECIMAL	6	0		☒
2	CLINOM	CLINOM	CHAR	40		284	☐
3	CLIDIR	CLIDIR	CHAR	60		284	☐
4	CLICIU	CLICIU	CHAR	30		284	☐
5	CLIPAI	CLIPAI	CHAR	3		284	☐
6	CLITEL	CLITEL	CHAR	20		284	☐
7	CLILIM	CLILIM	DECIMAL	11	2		☐
8	CLISLD	CLISLD	DECIMAL	11	2		☐
9	CLIFAL	CLIFAL	DATE				☐
10	CLIACT	CLIACT	CHAR	1		284	☐

IUSA - RAB - IBM i 7.3 Conectado a IUSA como RAB.

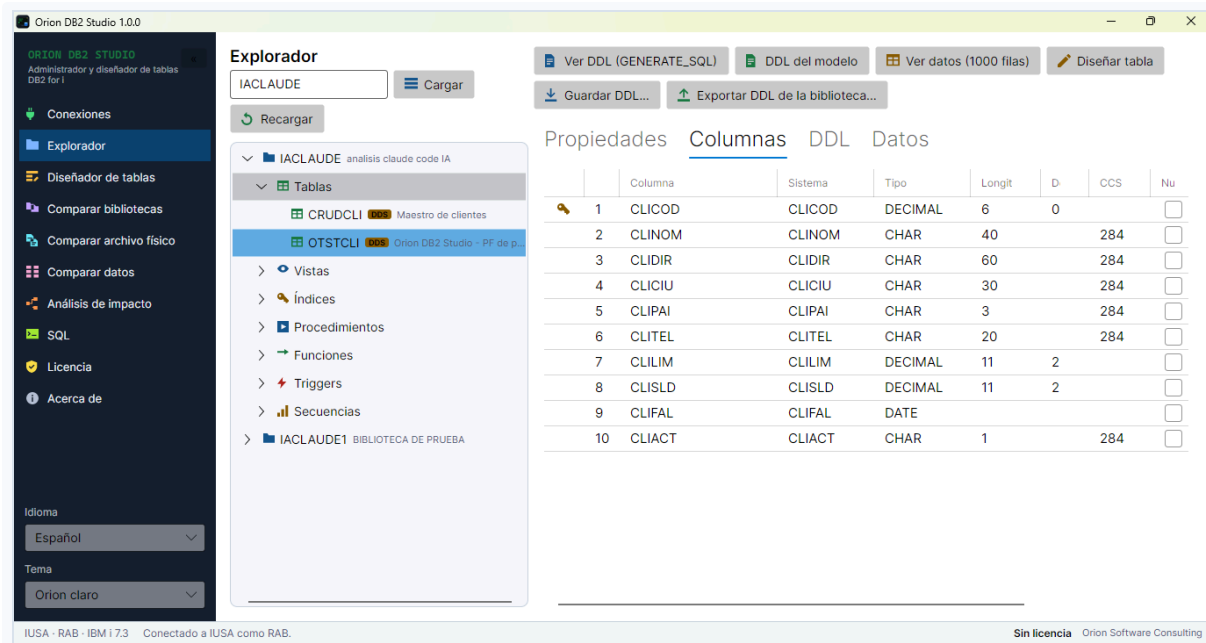
Sin licencia Orion Software Consulting

The columns of a physical file created with DDS, with the key flagged on the first row.

DDL

The **real** DDL of the object, the one the system itself returns through `QSYS2.GENERATE_SQL`. It works for files created with DDS too: it is the way to recover the definition of a file whose source nobody has any more.

FIGURE 09

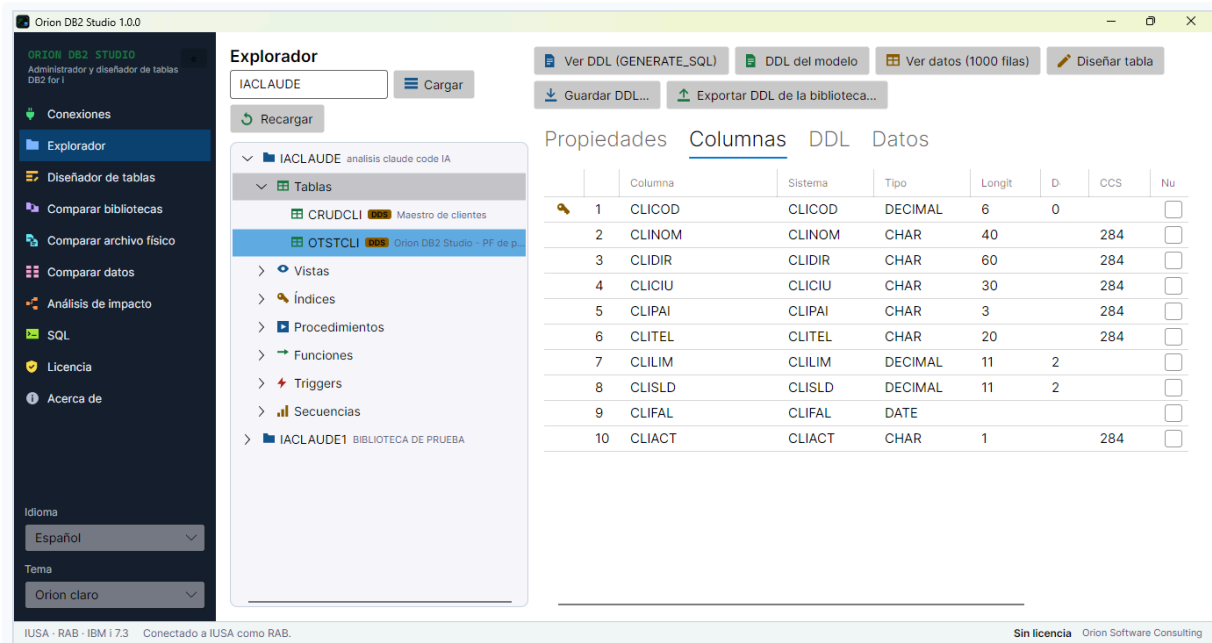


The right-hand panel, with the Properties, Columns, DDL and Data tabs.

Data

The first **1,000** rows of the file, so you can look at the content without leaving for another tool. It is available for tables and views; on an index, the application tells you which the base table is.

FIGURE 10



The first rows of a file, inside the browser itself.

The buttons at the top

View DDL (GENERATE_SQL)	The real DDL of the object, as it stands today
DDL from the model	The DDL Orion would generate from what it read in the catalogue. It is the route for modernising a DDS file to SQL
View data (1000 rows)	The first thousand rows
Design table	Loads the table into the designer
Save DDL...	Saves what the DDL tab shows into a <code>.sql</code> file
Export the library's DDL...	The DDL of every object in the library, one file per object, into the folder you choose

An important difference between the two DDLs. «View DDL» is what the system says. «DDL from the model» is what Orion reconstructs, and today it **does not turn the key** of a DDS file into a `PRIMARY KEY` or a `CREATE INDEX`: check it before using that script to create the object somewhere else.

6. The table designer

The **Designer** builds the DDL for you. It works in two modes: **create** a new table, and **change** one that already exists, where it generates the minimal `ALTER TABLE` by comparing what was there with what you left.

Nothing runs until you confirm it. While you edit, the DDL is written alongside and the validation panel reports the errors.

The header

Library	Where the table will be created
SQL name of the table	The long name, without the 10-character limit
System name (max. 10)	The short object name. Leave it empty and the system makes one up
Table text	<code>LABEL ON TABLE</code> , up to 50 characters
Comment	<code>COMMENT ON TABLE</code> , for the long description

Columns

The grid has one row per column: SQL name, system name, type, length, decimals, CCSID, whether it allows nulls, default value, column heading and text.

The buttons at the top add and remove columns and move them around. Two of them deserve an explanation:

Audit columns	Adds, in one click, the creation and change user and timestamp, with their default values already set
Suggest system names	Proposes the short name of each column by imitating DB2 for i's own algorithm, and avoids duplicates

FIGURE 11

Orion DB2 Studio 1.0.0

ORION DB2 STUDIO
Administrador y diseñador de tablas DB2 for i

Conexiones
Explorador
Diseñador de tablas
Comparar bibliotecas
Comparar archivo físico
Comparar datos
Análisis de impacto
SQL
Licencia
Acerca de

Idioma
Español
Tema
Orion claro

Crear tabla
Diseñe la tabla y vea el DDL generarse en tiempo real. Nada se ejecuta hasta que lo confirme.

Nueva tabla Añadir columna Quitar columna ↑ ↓ Columnas de auditoría Sugerir nombres de sistema

Biblioteca Nombre SQL de la tabla Nombre de sistema (máx. 10)
Texto de la tabla (LABEL ON TABLE, máx. 50) Comentario (COMMENT ON TABLE)

Columnas Llave primaria Únicas Foráneas Checks Índices

	Nombre SQL	Sistema	Tipo	Lo	E	CC:	↑	Por defec	Encabezado (60)	Texto (50)
1	ID		INTEGER				☐			
2	USUARIO_CREACI	USRCRE	VARCHAR	18			☐	USER	Usuario de creaci	
3	FECHA_CREACIO	FECCRE	TIMESTAMP				☐	CURRENT	Fecha de creaci	
4	USUARIO_MODIF	USRMOD	VARCHAR	18			☐	USER	Usuario de modifi	
5	FECHA_MODIF	FECMOD	TIMESTAMP				☐		Fecha de modifi	Última modificación de la

Guardar .sql Ejecutar ☐ 80 col. ☐ DROP antes

IUSA - RAB - IBM i 7.3 Conectado a IUSA como RAB. Sin licencia Orion Software Consulting

The designer with the audit columns just added: creation and change user and timestamp, with their default values.

Keys, constraints and indexes

The tabs under the grid hold the rest of the definition:

Primary key	Tick the columns and the order of each one
Unique	Uniqueness constraints over one or more columns
Foreign keys	With their target library, table and columns, and what to do on delete
Checks	Conditions every row must satisfy
Indexes	Each with its own system name

FIGURE 12

Orion DB2 Studio 1.0.0

ORION DB2 STUDIO
Administrador y diseñador de tablas DB2 for i

Conexiones
Explorador
Diseñador de tablas
Comparar bibliotecas
Comparar archivo físico
Comparar datos
Análisis de impacto
SQL
Licencia
Acerca de

Idioma
Español
Tema
Orion claro

Crear tabla
Diseñe la tabla y vea el DDL generarse en tiempo real. Nada se ejecuta hasta que lo confirme.

Nueva tabla Añadir columna Quitar columna ↑ ↓ Columnas de auditoría Sugerir nombres de sistema

Biblioteca Nombre SQL de la tabla Nombre de sistema (máx. 10)
Texto de la tabla (LABEL ON TABLE, máx. 50) Comentario (COMMENT ON TABLE)

Columnas Llave primaria Únicas Foráneas Checks Índices

Marque las columnas que forman la llave primaria. Todas deben ser NOT NULL.

Llave	Columna	Descendente
☐	ID	☐
☐	USUARIO_CREACION	☐
☐	FECHA_CREACION	☐
☐	USUARIO_MODIF	☐
☐	FECHA_MODIF	☐

Guardar .sql Ejecutar ☐ 80 col. ☐ DROP antes

IUSA - RAB - IBM i 7.3 Conectado a IUSA como RAB. Sin licencia Orion Software Consulting

The primary key tab: you tick the columns that make it up and the order of each one.

Validation and preview

The validation panel does not wait for you to press anything: as you type it reports duplicated or invalid system names, decimals greater than the length, types your IBM i release does not support and tables without a primary key. The summary says how many errors and how many warnings there are.

Save or run

Save .sql	Leaves the script ready for <code>RUNSQLSTM</code>
Run	Applies it against the IBM i, after the confirmation screen
80 col.	Wraps the lines at 80 columns, as <code>RUNSQLSTM</code> wants
DROP first	Adds the preceding <code>DROP TABLE</code> , in create mode only

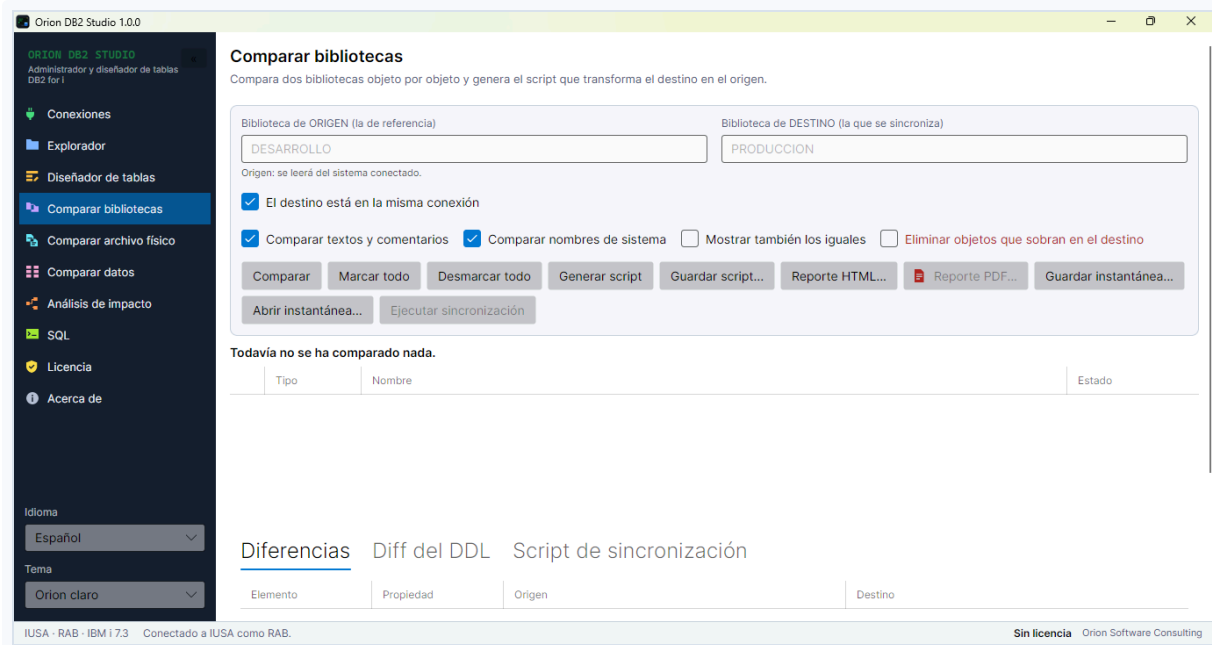
Changing a table is not free in DB2 for i. An `ALTER TABLE` can force the system to rebuild the file, and with a large table or logical files over it that takes time. The application marks every statement with its risk and, when the change alters the record format, offers you the **impact analysis** before you sign.

DB2 for i cannot rename a column. When you change the name of one, the designer generates `ADD COLUMN`, `UPDATE` and `DROP COLUMN`, marks it as data loss and warns you that those three steps are not atomic if the table is not journalled.

7. Comparing libraries

Compares two libraries object by object and generates the script that turns the target into the source. The two can live on different systems.

FIGURE 13



The comparison screen: source, target, options and the result grid.

What gets compared

Tables and their columns —type, length, nulls, default value, CCSID and texts—, constraints, indexes, views, routines, triggers and sequences. **GRANT** privileges are **not** compared.

The comparison runs over the **model** read from the catalogues, not over the text of the DDL: two identical objects written differently do not show up as different.

Options

The target is on the same connection	Clear it to compare against another system, and pick its connection
Compare texts and comments	Includes <code>LABEL ON</code> and <code>COMMENT ON</code>
Compare system names	Takes the short object name into account
Show the identical ones too	To see the whole library, not just what changed
Delete objects that are only in the target	Includes the <code>DROP</code> statements in the script. It is in red for what it means

The result

Every row carries its status —**Only in source**, **Only in target**, **Different** or **Identical**— and a tick box. The script is generated **only from what you tick**.

Two tabs explain each difference: **Differences** breaks it down property by property, and **DDL diff** shows both DDLs side by side, line by line.

The synchronisation script

Generate script lays out the `CREATE`, `ALTER` and `DROP` statements in **dependency order**: first what other objects need. You can keep it with **Save script...** or apply it with **Run synchronisation**, which goes through the confirmation screen first.

Snapshots

Save snapshot... writes the whole model of a library into a JSON file. It is there to compare the same system against itself a few months later, or to compare without the source system in front of you:

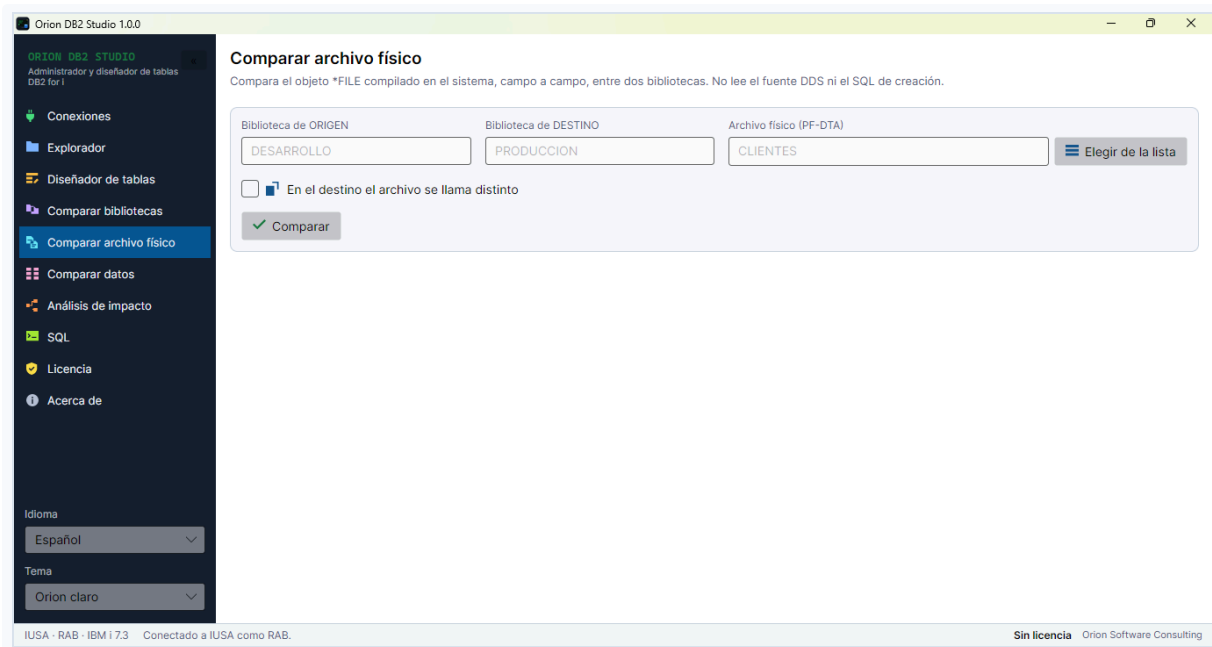
Open snapshot... loads it as the source.

The comparison reports come out in **HTML** —self-contained, to send by email— and in **PDF**.

8. Comparing a physical file

Compares two physical files **field by field**, reading the compiled object and not the DDS source or the SQL it was created with. It is the screen to go to when a program fails with **CPF4131** and you have to work out how the development file and the production one differ.

FIGURE 14



Physical file comparison: the file, the two libraries and the result by fields, keys and attributes.

Everything it does against the IBM i is **read only**.

What you have to give it

1. The **source library** and the **target library**.
2. The **file**. **Pick from the list** brings up the data physical files in the source library.
3. If it is named differently in the target, tick the box and type the other name.
4. Press **Compare**.

What it answers

First, the **record format level identifier** on both sides: the fingerprint the system computes from the structure of the file. If the two match, programs compiled against one open the other with no level check.

Then the differences, graded as **critical**, **warnings** and **informational**, across three tabs:

Fields	Ordinal and definition on both sides, what changes and how serious it is. Fields are matched by name, not by position
Keys	The access path on each side and how they differ
File attributes	What defines the object, beyond its fields

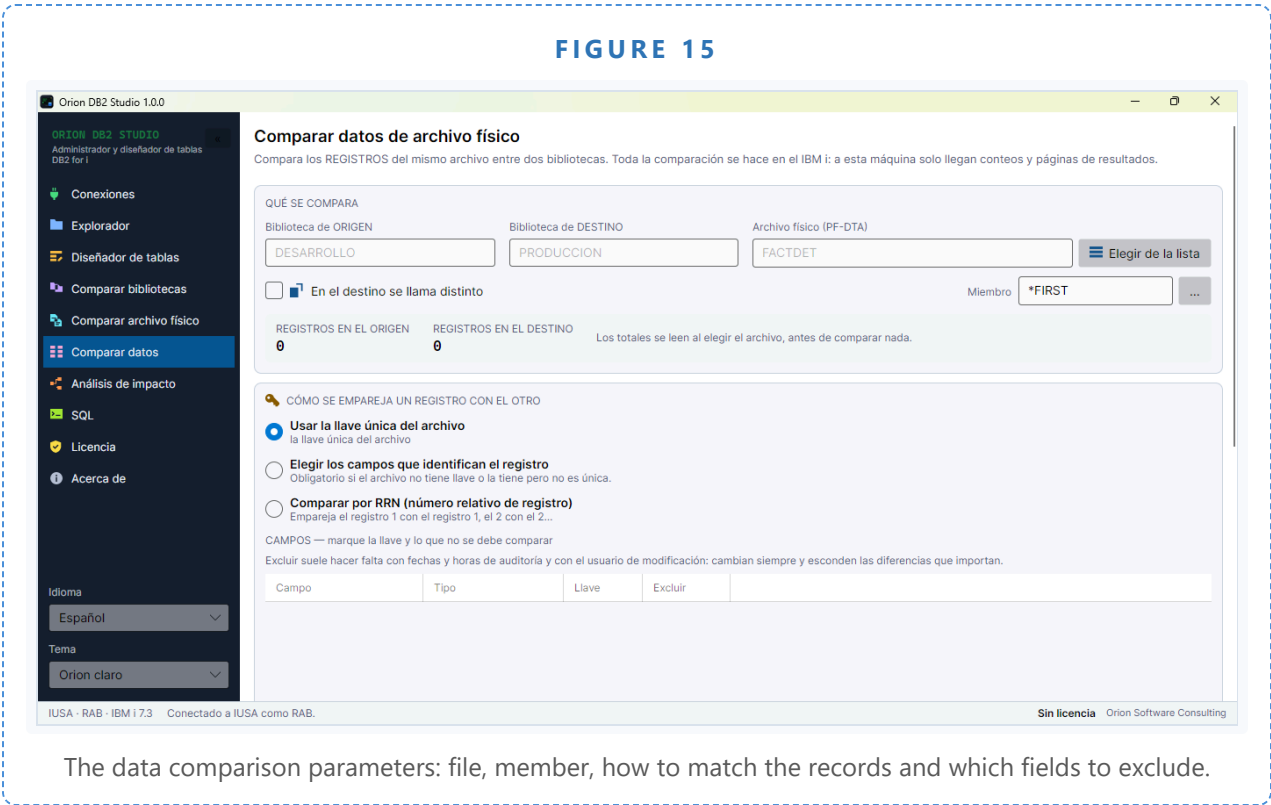
The result downloads as **PDF** through **Download PDF report**.

The level identifier covers **the record format**. The key and the file attributes can differ with the same fingerprint, which is why they are compared separately.

9. Comparing data

Compares the **records** of the same file between two libraries. The comparison runs **inside the IBM i**: only the counts and the pages of results you look at travel to your machine, never the whole data.

FIGURE 15



The data comparison parameters: file, member, how to match the records and which fields to exclude.

How records are matched

There are three ways, and choosing well is half the job:

By the file's unique key	The normal choice when the file has a unique key. It is ticked by itself
By the fields you choose	Required if the file has no key, or has one that is not unique
By RRN	Matches record 1 with 1, 2 with 2. Useful when no key is possible

Careful with RRN. The relative record number is the physical position inside the file: a single **RGZPFM** on either side changes everything and the comparison stops meaning anything. The application warns you on screen when you pick it.

What to exclude

In the field grid you can tick which ones make up the key and which are **excluded** from the comparison. It is worth excluding the audit dates and times and the change user: they differ between the two systems almost always, and if they are not excluded **every** record comes out as different.

Options that affect the time

WHERE filter	Applied identically to both sides, and validated against the server before starting
Ignore trailing spaces in CHAR fields	In a physical file the fields are fixed length: the padding belongs to the format, not to the data
Fast comparison by hash	Compares one hash per record and drills down only where they differ. It turns itself off if the system has no <code>HASH_SHA256</code>
Records per chunk	How many records are fetched at a time
Maximum differences detailed in the PDF	The cap on the report detail

While it runs, and the result

The progress screen shows the stage, the chunk, the elapsed time and the stages already finished with what each one cost. **Cancel** stops the query on the server and clears whatever had been created in `QTEMP`.

At the end there are six counters —records in source and target, identical, different, only in source and only in target— **which are always totals** even if the detail is capped, and a list of which fields the differences are in, with how many records each one affects.

The detail exports to **PDF** and, without the PDF cap, to **CSV**.

Inserting the missing records

Insert into the target the records that are missing copies across what is only in the source. It shows the SQL before running anything and asks for a typed confirmation when the target is not journalled, because then there will be no way back.

It only inserts. It does not update the records that differ nor delete the ones left over, and it does not work in RRN mode. That is deliberate: a full data synchronisation between two environments is a business decision, not a button.

10. Impact analysis

It answers the question nobody on IBM i answers from memory: **if I change this table, what stops working tomorrow**. The system does not warn you. A compiled program stores the record format fingerprint and, when it opens the file, either it matches or the program fails with **CPF4131**.

FIGURE 16



The analysis screen before starting: the table, the libraries to search and the changes coming from the designer.

How it is launched

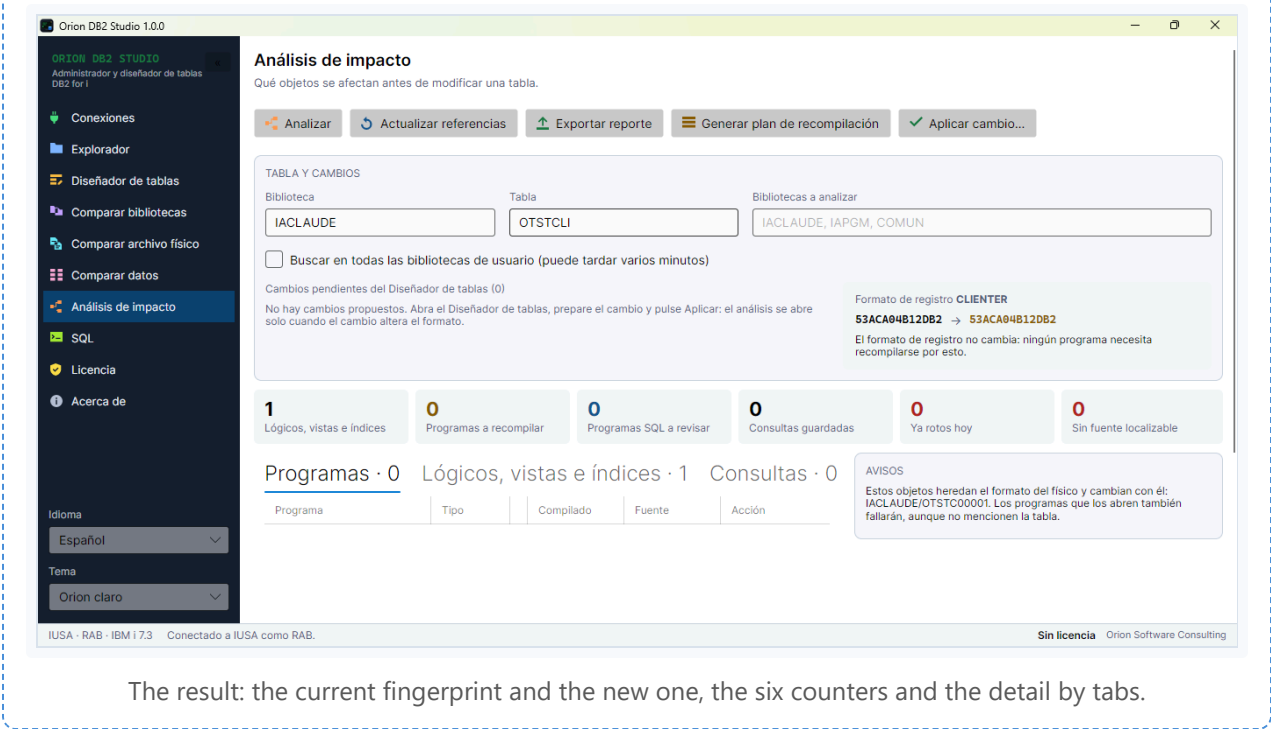
1. Give it the **library** and the **table**.
2. In **Libraries to analyse**, the libraries where programs should be looked for, separated by commas. Or tick **Search every user library**.
3. Press **Analyse**.

While it works, a panel at the top shows the spinner, the stage, the library counter, the progress bar, the elapsed time and the **Cancel** button.

Every user library can take several minutes. The application runs one **DSPPGMREF** per library, and on a system with hundreds of libraries that is a lot of work. The references are kept in a local cache, so the second analysis the same day is much faster; **Refresh references** ignores it and reads everything again.

What it answers

FIGURE 17



At the top, the **record format fingerprint**: the current one and the one that would be left after the change. If they match, no program needs recompiling because of this.

Below, six counters and their detail:

Logical files, views and indexes	What hangs off the table
Programs to recompile	The ones that will fail with a level check
SQL programs to review	The ones using SQL: they do not fail on the fingerprint, but they may refer to a column that changed
Saved queries	The <code>*QRYDFN</code> objects in the analysed libraries
Already broken today	Programs whose compiled fingerprint no longer matches the file, before you touch anything
No source found	The ones that cannot be fixed by recompiling

How it knows the new fingerprint

It does not compute it: it measures it. The algorithm behind the record format level identifier is not public, and guessing it would be worse than not trying: an invented fingerprint that came out equal to the current one would say «nothing to recompile» exactly when there is.

What it does is **simulate**: it copies the structure of the file —with no data— into a temporary library, applies the change and reads the resulting fingerprint with `DSPFD`. The temporary library is always dropped, including when the analysis fails or you cancel.

If the change is **destructive**, the IBM i itself cancels it from a client connection —it sends an inquiry message nobody can answer— and the fingerprint comes back marked as **estimated**. The application says so on screen instead of passing off a measurement it could not take.

«It inherits the format» is not guessed either

A logical file that shares the format of the physical breaks just like it does, even if no program mentions it. To know which ones share a format, the application **compares the fingerprints**: same fingerprint, same format. It is not inferred from the DDS or from `ADD ALL COLUMNS`.

The recompilation plan

Generate recompilation plan turns the analysis into a list of steps in order: backup, substitute logical file if one is needed, the table change and the recompilations —modules first, then programs, then service programs—. The script comes out in SQL and CL, editable before you run it, and exports to HTML.

The substitute logical file

When there are programs without source, recompiling is not an option. For those cases the application proposes **renaming the physical file and creating a logical one with the original name, format and fingerprint**: native I/O programs keep working untouched.

The substitute logical file has a price worth knowing: programs that **write** through it leave the new columns with their default value, because they do not know about them.

The analysis report exports to **HTML** and **Excel**.

11. Application analysis

Impact analysis answers for one table. This module answers for the **whole application**: what the library holds, who uses what and how to document it. It arrived in version **1.1.0** and lives under a single sidebar entry, with seven screens inside.

Everything comes from a snapshot

None of these screens talks to the IBM i while you use it. First you **analyse** the library —once— and that leaves a dated snapshot in a local database on your machine; after that you query it as many times as you need, **offline and without loading the system**. Every screen always says what date the data is from.

The collection is **read only**: catalogue queries and `DSPxxx` commands writing to a temporary file in `QTEMP`, which dies with the job. It works on a connection marked read only and with an expired licence.

Analysing libraries

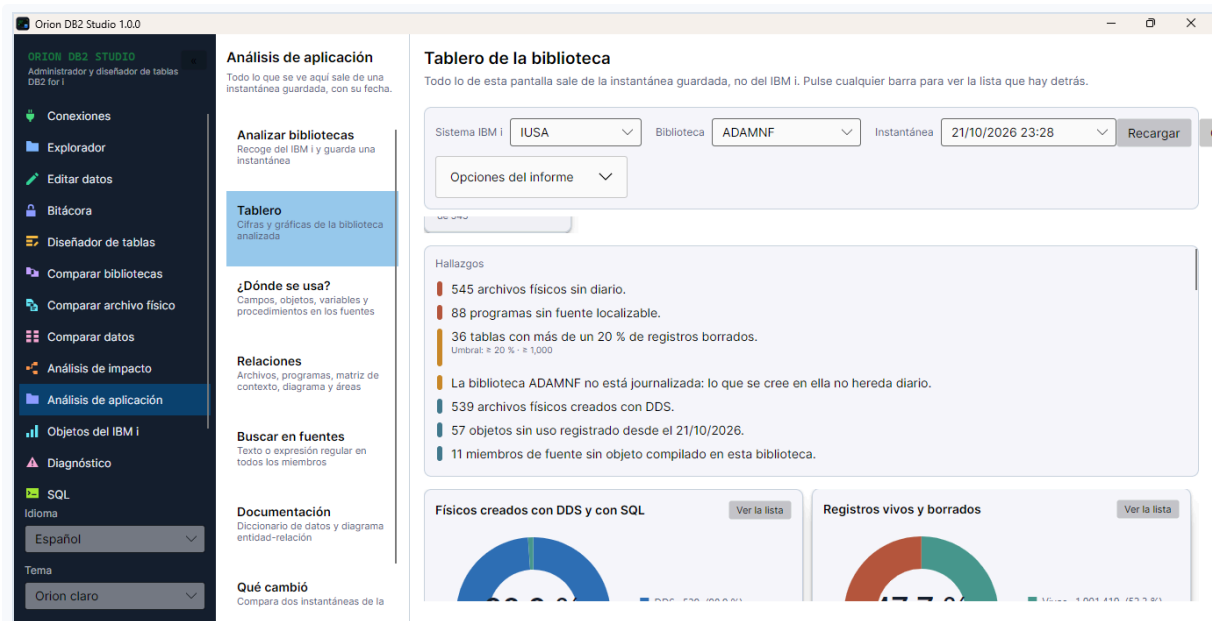
You choose the library and which parts to collect. What takes longest by far is the **source**: measured against a real IBM i 7.3, reading and parsing one member costs around 436 ms, so a library with 2,000 members is about fifteen minutes the first time. From the second run only the members that changed are fetched, and passing over an unchanged one costs 52 ms.

Inventory	Objects in the library, with size, dates and use
Files	Formats, fields, keys, indexes, constraints and triggers
Members	Those of every file, with their dates
References	Which object each program uses, from the compiled object
Relations	Dependencies between files, from <code>DSPDBR</code>
Source	The text of the members and their parsing

Dashboard

The figures of the library in cards, a list of **findings** and sixteen charts. The findings are what you have to look at: physical files without journalling, programs with no source to be found, tables with too much dead space, objects with no recorded use. Every chart can be clicked and opens the list behind it, which exports to CSV.

FIGURE 18



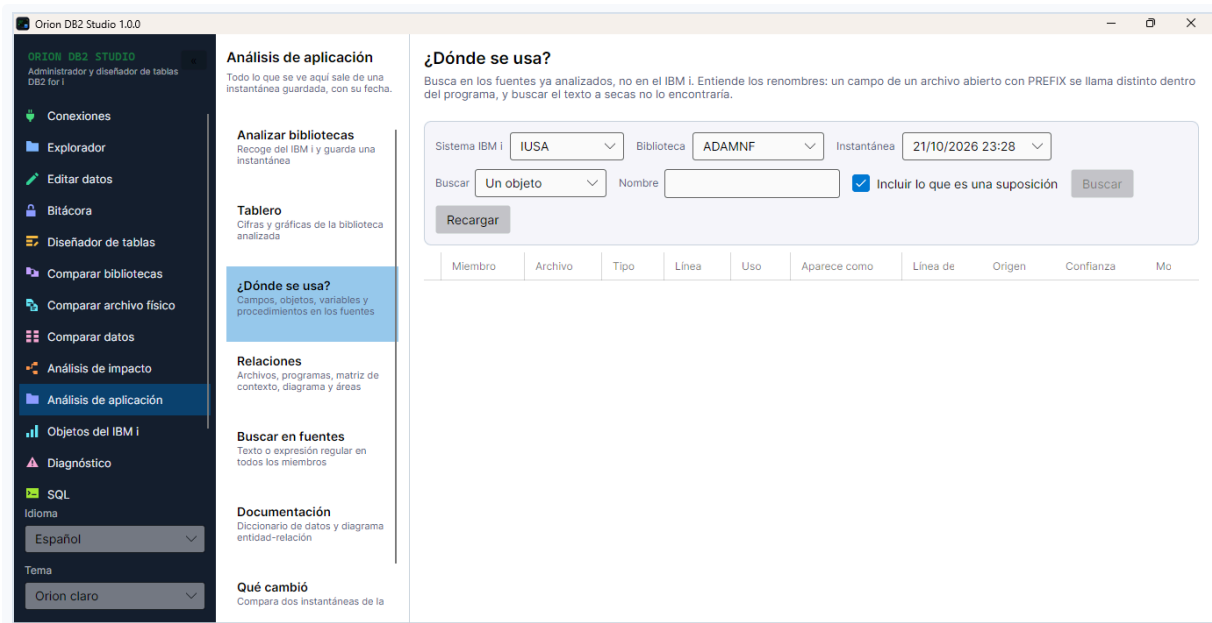
The dashboard of a library: the figures at the top and, below them, the findings.

«Generate PDF report» puts the whole dashboard into a document with a cover, an executive summary, one section per chart with its table and its annexes. You can ask for it in Spanish or English, with your company logo on the cover.

Where is it used?

The question nobody on an IBM i answers from memory. You search for a **field**, an **object**, a **variable** or a **procedure**, and the answer brings the member, the line, the text of that line and whether the name is **read** or **written**.

FIGURE 19



The search: you pick what you are looking for and which file the field belongs to.

It reads **RPG IV** —fixed, free and mixed—, **RPG III**, **CL** and **COBOL**, with the embedded SQL of all of them. And it understands renaming, which is what separates it from a plain text search: a file opened with **PREFIX(CL_)** makes the field **CLNOM** be called **CL_CLNOM** inside that program, and searching for «CLNOM» would find nothing. The screen shows which names it searched with.

Every result carries its **confidence**. A **CHAIN** over a file is a fact; that a loose name in an expression is that field and not a variable with the same name is a guess, and it comes out marked as such. The «include what is a guess» box leaves them out.

Members that **could not be parsed** appear in a list of their own. «It does not appear» and «it was not looked at» are not the same thing, and the difference is paid for by whoever changes the field believing nobody else uses it.

Relations, matrix and diagram

- **Files by program:** what each one opens, with which format and what it does to it. It joins what the compiled object says with what the source says, which also sees the embedded SQL.
- **Fields by file:** attributes, texts, headings, keys and in how many programs each field is used.
- **Context matrix:** files down the rows, programs across the columns, exportable to Excel.
- **Diagram:** file hierarchy, call tree or entity-relationship, with zoom and panning, and export to SVG and PNG.

- **Application areas:** grouping objects by prefix or by list so as to filter the dashboard and the report.

Searching the source

The **grep** of the library: text or a regular expression over every member, with case sensitivity, whole word, lines of context and the option to skip comments. You can search **every analysed library** at once and filter by source file. The results export to CSV.

Documentation: the data dictionary

The document an audit or a handover asks for and almost nobody has written down: files, fields with their type and text, keys, indexes, constraints, triggers and the programs that use each file, with the **entity-relationship diagram**.

FIGURE 20

Orion DB2 Studio 1.0.0

ORION DB2 STUDIO
Administrador y diseñador de tablas DB2 for i

Conexiones

Explorador

Editar datos

Bitácora

Diseñador de tablas

Comparar bibliotecas

Comparar archivo físico

Comparar datos

Análisis de impacto

Análisis de aplicación

Objetos del IBM i

Diagnóstico

SQL

Idioma
Español

Tema
Orion claro

Análisis de aplicación
Todo lo que se ve aquí sale de una instantánea guardada, con su fecha.

Análisis bibliotecas
Recoge del IBM i y guarda una instantánea

Tablero
Cifras y gráficas de la biblioteca analizada

¿Dónde se usa?
Campos, objetos, variables y procedimientos en los fuentes

Relaciones
Archivos, programas, matriz de contexto, diagrama y áreas

Buscar en fuentes
Texto o expresión regular en todos los miembros

Documentación
Diccionario de datos y diagrama entidad-relación

Qué cambió
Compara dos instantáneas de la

Documentación
El diccionario de datos de una biblioteca: archivos, campos, tipos, textos, encabezados, llaves y relaciones, con su diagrama entidad-relación.

Sistema IBM i: IUSA

Biblioteca: ADAMNF

Instantánea: 21/10/2026 23:28

Formato: Página HTML

Secciones

☒ Incluir el diagrama entidad-relación

☒ Contar en cuántos programas se usa cada campo (tarda más)

☒ Abrir al terminar

Nodos máximo: 300

Las relaciones deducidas salen de campos que coinciden en nombre y tipo entre dos tablas. En una base creada con DDS no hay llaves foráneas: la relación existe, pero solo en la cabeza de quien escribió los programas. El documento las marca como deducidas y las dibuja punteadas.

Generar el documento Recargar

The dictionary options: format, sections and diagram.

HTML	Self-contained, with the diagram inside; for the intranet
PDF	To hand over or to file
Word	To edit it and add what the tool cannot know
Excel	One sheet per kind, to filter and sort

Relations between tables that are **not** declared foreign keys are **inferred** from fields matching by name and type. In a database created with DDS there are no foreign keys, so nearly all of them will be. The document marks them in all four formats—a dashed line in the diagram, an «is it a fact?» column in the Excel—and they should be read with that in mind.

What changed

Compares two snapshots of the same library and says what was created, what was deleted and what changed between the two dates, highlighting the format changes, which are the ones that force a recompile.

From the command line

```
orion-db2 analyze    --conn IUSA --libraries MYLIB --with-source
orion-db2 dashboard  --system IUSA --library MYLIB --pdf analysis.pdf
orion-db2 where-used --system IUSA --library MYLIB --name CLNOM --file CLIENTES
orion-db2 doc         --system IUSA --library MYLIB --out dictionary.html
```

None of the four writes to the IBM i. `dashboard`, `where-used` and `doc` do not even connect unless you ask for `--refresh`: they work from the last saved snapshot, so they can be left in a nightly job that does not depend on the system being up.

12. The SQL editor

For what does not fit in a screen: one-off queries, maintenance statements and CL commands, against the active connection.



Running

Run all (F5)	Everything in the editor, split on ;
Run selection	Only what is selected, or everything if nothing is (Ctrl+Enter)
Max. rows	Cap on the rows brought into the grid
With commitment control	Wraps the changes in a transaction and rolls back if something fails

CL commands

A line starting with `CL:` is run as a system command through `QCMDEXC`, with the same confirmation screen as every other change. For example:

```
CL: CHGJOB INQMSGRPY(*SYSRPYL)
```

Results and messages

The **Results** tab holds the grid; **Messages** keeps the log of what was run, with the time of each statement and the rows affected. Whatever is in the grid exports to **CSV** and **Excel**.

Two limits worth knowing. The splitter breaks statements on `;`, so you **cannot create procedures, functions or triggers** that carry a `BEGIN ... END` block inside: for that, save the script and run it with `RUNSQLSTM`. And «Max. rows» is a hard cap, not paging: if the result has more, you will not see them.

The editor does **not autocomplete** library, table or column names yet. It highlights the syntax and runs; autocompletion is planned for a coming version.

13. Security: nothing happens without confirmation

The application works against production databases. What follows are not options you can switch off in preferences: they are in the way.

The confirmation screen

Before any `CREATE`, `ALTER`, `DROP`, `DELETE`, `UPDATE` or CL command, the application shows a screen with:

- The **system** and the **user** it will run as.
- The **library** affected.
- **How many statements** are going to run.
- The **risk** of the script.
- The **full script**, to read before signing it.

It appears on the five paths that write: the designer, library synchronisation, data synchronisation, the impact plan and the SQL editor.

Production connections

A connection marked as **PRODUCTION** changes two things: the application shows a **permanent red band** with the system and the library, and the confirmation requires **typing the library name** to enable the run button.

Changes that lose data

Dropping a column, shortening a field, reducing the decimals or changing to an incompatible type are marked as **data loss**: they come out in red, require the same typed confirmation and the button changes its text to «Run anyway».

Read-only mode

A connection marked as **Read only** rejects every modifying statement and every CL command **in the data layer**, not just by hiding buttons. It is the way to hand the tool to someone who only has to look.

Passwords

They are stored encrypted by the operating system —DPAPI on Windows, tied to your user account— and never appear in clear text in the configuration file or in the logs.

Backups

Before changing an existing table, the designer **offers to save the current DDL** of the object. The impact plan goes further: it includes a backup of the data with `CREATE TABLE ... AS` in the library you name.

And on the command line

The `orion-db2` executable **never runs changes** unless you ask for them with `--apply --yes`, and it **refuses** to apply them against a connection marked as production, pointing you at the desktop application.

14. Reports and command line

Reports

Report	Formats	Where from
Library comparison	HTML and PDF	Compare libraries
Physical file comparison	PDF	Compare physical file
Data comparison	PDF and CSV	Compare data
Impact analysis	HTML and Excel	Impact analysis
Recompilation plan	HTML	Impact analysis
Library dashboard	PDF	Application analysis
Data dictionary	HTML, PDF, Word and Excel	Application analysis
Query result	CSV and Excel	SQL editor

The HTML ones are **self-contained**: a single file that can be emailed and opens in any browser.

The orion-db2 executable

For automation from a script, a scheduled job or a pipeline. It uses the same connection profiles as the desktop application.

ddl — the DDL of a library

```
orion-db2 ddl --conn IUSA --library MYLIB --out C:\ddl\
```

Generates one file per object. `--tipos` limits it to certain types and `--archivo-unico` puts everything together.

compare — compare two libraries

```
orion-db2 compare --source IUSA:DEVELOPMENT --target IUSA:PRODUCTION \  
--script sync.sql --report report.html
```

The source can be a `.json` snapshot instead of a live system. Exit codes: **0** if they are identical, **1** if there are differences and **2** on error, so it can be chained in a script.

impact — the impact analysis

```
orion-db2 impact --conn IUSA --table MYLIB/CUSTOMERS \  
                --change "ADD COLUMN CUEMAIL CHAR(60)" --report impact.html
```

Read only: the only thing it creates is the temporary library for the simulation, and it always drops it.
Exit codes: **0** no impact, **1** recompiling needed, **2** blockers or programs already broken, **3** error.

analyze, dashboard, where-used and doc — the application

```
orion-db2 analyze  --conn IUSA --libraries MYLIB --with-source  
orion-db2 dashboard --system IUSA --library MYLIB --pdf analysis.pdf --logo logo.png  
orion-db2 doc      --system IUSA --library MYLIB --out dictionary.docx
```

The four commands of the analysis module. They work from the saved snapshot and only connect with `--refresh`.

export — data and dictionary

```
orion-db2 export --conn IUSA --sql "SELECT * FROM MYLIB.CUSTOMERS" \  
                --out customers.xlsx
```

With `--diccionario` instead of `--sql` it produces the data dictionary of a library in HTML.

15. Troubleshooting

It does not connect and says the timeout expired

The system is down, there is no route to it, or a firewall is dropping the traffic without answering. Ping the system and check the VPN.

It says the system refused the connection

The IBM i answered, but nobody is listening on the port. Start the host servers with `STRHOSTSVR SERVER(*ALL)` and check with `NETSTAT *CNN` that ports 8470, 8471 and 8476 are listening.

It says the system has no TLS configured

The host servers are up and answer in the clear, but the secure port is closed. Clear **Use TLS**, or ask the administrator to configure SSL in the Digital Certificate Manager and restart the services.

The user profile was disabled

The IBM i does that after several failed attempts, according to `QMAXSIGN`. Ask for `CHGUSRPRF USRPRF(YOURPROFILE) STATUS(*ENABLED)`.

An ALTER does not apply and the error mentions SQL0952

The system sent an inquiry message —CPA32B2, «changing the file may cause data to be lost»— and the server job answered it with «cancel», which is its default reply. It is not a fault of the application nor a timeout. Three ways out:

- Make the change in two steps: create the new table, copy with `INSERT ... SELECT` and rename.
- Run the script from a 5250 session and answer I.
- Ask the administrator for a system reply list entry (`ADDRPYLE` with `MSGID(CPA32B2) RPY(I)`) and run, beforehand and on the same connection, `CL: CHGJOB INQMSGRPY(*SYSRPYL)`.

An UPDATE or a DELETE fails with SQL7008

The table is not journalled and the operation requires it. You need a journal in the library and then `STRJRNPF`. Libraries created with `CREATE SCHEMA` already have one; those created with `CRTLIB` do not.

The session drops after a while unused

That is the VPN or the firewall closing idle connections. The application detects it and warns you; press **Connect** again.

Accented characters come out wrong

That is the CCSID. Leave it empty in the profile so that the one from the IBM i user profile is used, or set the one that matches your system —284 for Spain and much of Latin America—.

The impact analysis did not find a program's source

The application looks for the source member in the library and file the object itself declares. Today it **does not search the IFS** nor compare the content of the member with the compiled object: if the source was moved, it will show as not found.

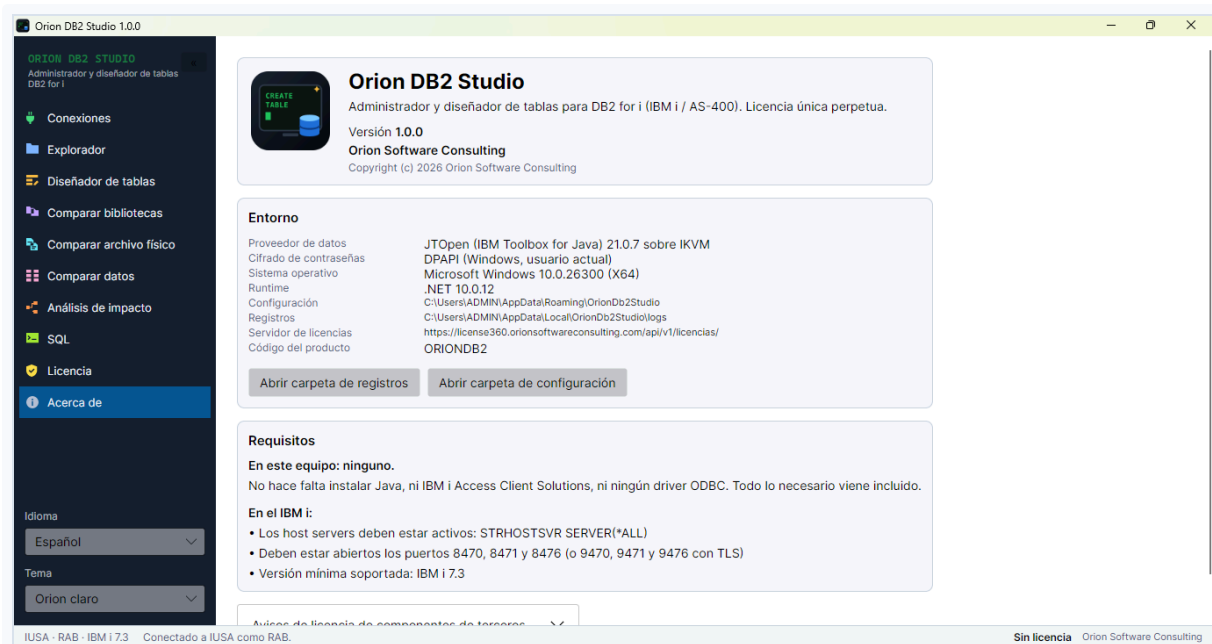
The application analysis left members unparsed

They appear in the problem list with the reason. COBOL, RPG and CL are parsed; anything else —or a member whose text could not be read— is listed instead of being passed over in silence. A library that reports no problems at all and no parsed members is a library analysed without the source option.

Where are the application logs?

In the user's log folder, which opens from **About**. It keeps fourteen days. If you are going to report a problem, the file for the day is the first thing to attach.

FIGURE 22



«About»: the version, the runtime, the requirements and the links to the configuration and log folders.

16. Appendix: list of screenshots

The 22 screenshots in this manual, in order. All of them were taken against a real IBM i using the test library `IACLAUDE`.

Figure	What it shows
FIGURE 01	The Connections screen, which is where you start. On the left the sidebar with its sections; at the bottom, the language and theme selectors.
FIGURE 02	The Licence screen: current status, the holder's details, the free trial button, the box for activating a purchased key and this machine's identifier.
FIGURE 03	«Test connection» against an IBM i 7.3: the dialog with the result and the same message in the band above.
FIGURE 04	Connected: the message in the top band and, bottom left, the system, the user and the IBM i release.
FIGURE 05	The browser after loading the filter: the libraries that match, each with its descriptive text.
FIGURE 06	The tree open: library, object families and, inside Tables, the files with their DDS badge.
FIGURE 07	Properties of a table: what to look at before touching it.
FIGURE 08	The columns of a physical file created with DDS, with the key flagged on the first row.
FIGURE 09	The right-hand panel, with the Properties, Columns, DDL and Data tabs.
FIGURE 10	The first rows of a file, inside the browser itself.
FIGURE 11	The designer with the audit columns just added: creation and change user and timestamp, with their default values.
FIGURE 12	The primary key tab: you tick the columns that make it up and the order of each one.
FIGURE 13	The comparison screen: source, target, options and the result grid.
FIGURE 14	Physical file comparison: the file, the two libraries and the result by fields, keys and attributes.
FIGURE 15	The data comparison parameters: file, member, how to match the records and which fields to exclude.
FIGURE 16	The analysis screen before starting: the table, the libraries to search and the changes coming from the designer.
FIGURE 17	The result: the current fingerprint and the new one, the six counters and the detail by tabs.

Figure	What it shows
FIGURE 18	The dashboard of a library: the figures at the top and, below them, the findings.
FIGURE 19	The search: you pick what you are looking for and which file the field belongs to.
FIGURE 20	The dictionary options: format, sections and diagram.
FIGURE 21	The editor with a query run and its results in the grid.
FIGURE 22	«About»: the version, the runtime, the requirements and the links to the configuration and log folders.