

Manual de 4D Server

-  Introducción
-  4D Server en 10 minutos
-  Uso de 4D Server
-  Ventana de administración de 4D Server
-  Métodos base 4D Server
-  Uso de un 4D Remoto
-  4D Server y el lenguaje 4D
-  Lista alfabética de los comandos

Introducción

 Presentación

 Arquitectura de 4D Server

Presentación

4D Server es el servidor de datos y de aplicaciones multiplataforma de 4D.

Con 4D Server, puede crear y utilizar bases de datos multiusuarios y aplicaciones personalizadas en arquitectura cliente/servidor. La arquitectura cliente/servidor independientemente de la plataforma de 4D Server administra aplicaciones para clientes 4D en Windows y Macintosh. 4D Server incluye herramientas de desarrollo y de seguridad de datos, escalabilidad total y opciones de conectividad a todos los sistemas empresariales.

La arquitectura de 4D Server está totalmente integrada: el cliente y el servidor utilizan una aplicación 4D única y los desarrolladores no tienen que diseñar aplicaciones específicas para el servidor y para los clientes. Además, 4D Server es un servidor "cero administración", es sencillo de instalar, utilizar, administrar y es extremadamente rentable.

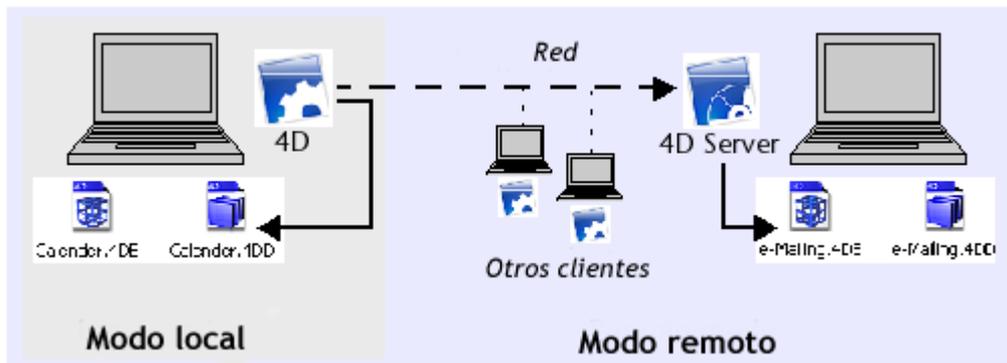
4D Server cierra la brecha entre sistemas basados en compartimiento de archivos de bajo nivel, lentos y poco eficaces y los sistemas más avanzados de RDBMS basados en SQL que a pesar de ser poderosos son complejos de desplegar y de evolucionar en el tiempo. Una aplicación 4D Server puede integrarse fácilmente a los sistemas de información existentes en las empresas (tales como Oracle, Sybase, o todo servidor compatible ODBC). 4D Server direcciona las necesidades de los grupos de trabajo en empresas de todos los tamaños.

Arquitectura cliente/servidor integrada

Un sistema 4D Server funciona con una aplicación única para el servidor y el cliente. El software cliente y la aplicación servidor son dos caras del mismo producto, 4D. La aplicación 4D Server está dividida en dos partes: 4D Server y 4D en modo remoto, que forman la arquitectura cliente/servidor.

La parte 4D Server reside en la máquina servidor y almacena y administra la base de datos en el servidor y permite a los usuarios manipular la base desde sus propias máquinas (los clientes).

La aplicación 4D reside en cada máquina cliente y puede utilizarse en modo local o remoto. En modo local, los usuarios pueden trabajar con una base de datos o una aplicación 4D almacenada localmente en su máquina. En modo remoto, los usuarios acceden a la base en el servidor y realizan operaciones de base de datos tales como añadir registros, generar informes, y modificar el diseño de la base. Todo puede realizarse con 4D en modo local utilizando 4D Server y 4D en modo remoto.



No es necesario ningún middleware ni desarrollos adicionales para operar en entorno cliente/servidor. 4D y 4D Server comparten las mismas herramientas de interfaz, el mismo lenguaje y la el mismo sistema de gestión de la información.

Una aplicación monopuesto local evoluciona fácilmente hacia una configuración cliente/servidor para el grupo de trabajo. Igualmente, una aplicación creada con 4D Server pasa automáticamente a una aplicación monopuesto con 4D en modo local.

Servidor de datos y de aplicaciones "Cero Administración"

4D Server es como 4D, centrado en el usuario. 4D Server es un sistema Plug and Play (PNP) completo.



Ventana de administración de 4D Server

- **Administración centralizada en línea, gráfica:** la ventana de administración de 4D Server muestra automáticamente información importante relacionada con la actividad de la aplicación: memoria total asignada al 4D Server, caché de datos, número y nombre de usuarios conectados, número de procesos y estado de cada uno, monitoreo de backups y peticiones y actividad de los servidores.

- **Auto configurable y escalable:** 4D Server está diseñado para integrar nuevos protocolos, clientes y plug-ins sin que sea necesario reconfigurar o hacer reingeniería.
- **Actualización automática y dinámica de los equipos clientes, control de versiones:** todos los equipos 4D conectados reciben automáticamente y de manera dinámica los nuevos elementos de una aplicación cada vez que la base evoluciona o que un plug-in o componente de plug-in se añade, borra o modifica. Además, si ha creado una aplicación cliente-servidor personalizada, es posible descargar automáticamente las nuevas versiones de las aplicaciones clientes 4D ejecutables cuando la aplicación 4D Server se actualiza.
- **Conexiones automáticas asincrónicas vía el protocolo estándar TCP/IP:** 4D Server y 4D se comunican de manera transparente utilizando el protocolo TCP/IP sin importar la plataforma del cliente y del servidor. Como el protocolo TCP/IP está integrado en todos los sistemas operativos, su uso no requiere ninguna instalación adicional.
- **Gestión simultánea de las sesiones y de los contextos de las conexiones 4D, SQL y HTTP:** 4D Server crea y mantiene automáticamente un entorno de trabajo actual para cada combinación tabla/proceso/usuario. Esta arquitectura basada en las sesiones permite a cada proceso usuario manipular los datos independientemente y simultáneamente. El servidor SQL de 4D Server maneja automáticamente las peticiones SQL internas o externas. El servidor HTTP de 4D Server responde a las peticiones HTTP y eventualmente a las peticiones SOAP.
- **Bloqueo automático de registros:** 4D Server ofrece bloqueo y liberación automática de los registros, evitando problemas comunes asociados a la modificación de los registros "en uso". El bloqueo de registros también elimina problemas asociados con el bloqueo de páginas o archivos que se encuentran en otros sistemas.
- **Sistema de mensajería usuario integrada:** 4D Server ofrece todos los aspectos de interfaz usuario que responden a las necesidades de los entornos de desarrollo integrados modernos. Por ejemplo, 4D Server puede informar a los clientes de las acciones de administración que se estén llevando a cabo, tales como desconexiones y backups.
- **Métodos de inicio y de desconexión automatizados:** 4D Server llama automáticamente varios métodos base que responden a aspectos específicos: **On Server Startup**, **On Server Shutdown**, **On Server Open Connection**, **On Server Close Connection**, y **On Web Connection**. El **Método base On Server Startup** puede inicializar y cargar automáticamente todos los objetos que se utilizarán durante el resto de la sesión.

Un conjunto de funciones inigualables

Además de las funcionalidades de 4D, 4D Server ofrece las siguientes características:

- **Gestión de datos en entorno multiusuarios:** varios usuarios pueden efectuar simultáneamente operaciones de base de datos en la misma tabla o en tablas diferentes tales como añadir, modificar, borrar, buscar, ordenar e

imprimir registros. La integridad de los datos es asegurada por un sistema interno de bloqueo de registros.

- **Desarrollo multiusuarios:** varios usuarios pueden desarrollar y diseñar una base simultáneamente. Por ejemplo, los miembros de un equipo pueden al mismo tiempo modificar los atributos de una tabla, crear y modificar formularios y métodos. La integridad de la estructura está protegida por un sistema interno de bloqueo de objetos.
- **Arquitectura cliente /servidor independiente de la plataforma:** la arquitectura administra el rendimiento de la base de datos de manera idéntica para los clientes Macintosh y Windows. Incluyendo plataformas multidesarrollo plataforma cruzada simultáneo o para las operaciones de entrada de datos y de modificación de datos para los equipos clientes en entornos materiales diferentes.
- **Versión 64 bits:** 4D Server en Windows (desde la versión 12.1) y en Mac OS (desde la versión 15.1) está disponible en versiones 64 bits. La arquitectura 64 bits le permite a sus aplicaciones asignar más memoria RAM.
- **Arquitectura de plug-ins Windows y Mac OS:** la versión Windows y la versión Mac OS de 4D Server le permite instalar a la vez plug-ins Windows y Mac OS en el equipo servidor. Esta arquitectura simplifica la distribución de plug-ins 4D independientemente de la plataforma. Los plug-ins son manejados de forma transparente por 4D y 4D Server, sin importar la plataforma de ejecución del cliente.
- **Servidor HTTP integrado:** tal como 4D en modo local, 4D Server y cada 4D en modo remoto tienen un motor HTTP que permite publicar las bases 4D en la Web. Su base de datos puede ser publicada directamente en la Web. No necesita desarrollar un sistema de base de datos, un sitio web o una interfaz CGI entre ellos. Su base es su sitio web. También puede transformar todo equipo 4D remoto en un servidor web. Para mayor información sobre el motor web integrado de 4D Server y 4D, consulte la sección en el manual Lenguaje de 4D.
- **Seguridad de las conexiones:** puede configurar su 4D Server de manera que las conexiones cliente/servidor se efectúen en modo seguro, por intermedio del protocolo TLS/SSL. Para mayor información, consulte la sección **Encriptar las conexiones cliente-servidor**.
- **Triggers:** un trigger es un método asociado a una tabla. Es una de las propiedades de la tabla. Usted no llama a los triggers; ellos son llamados automáticamente por el motor de la base de datos cada vez que manipula los registros de la tabla (añadir, borrar y modificar). Con 4D Server, los triggers se ejecutan en el equipo servidor. Todo cliente, bien sea 4D o una aplicación conectada vía ODBC, está sujeta a las reglas de la base de datos controladas por los triggers. Para mayor información sobre los triggers 4D, consulte la sección **Triggers** en el manual Lenguaje de 4D.
- **Procedimientos almacenados:** Puede crear métodos 4D que se ejecutarán localmente en su propio proceso en el equipo servidor o en varios equipos clientes. Utilizando la terminología de la industria cliente/servidor estas funciones se llaman "procedimientos almacenados." Sin embargo, 4D Server ofrece una arquitectura que va más allá del concepto estándar de los procedimientos almacenados. Con 4D Server, un procedimiento almacenado es en realidad un proceso servidor (o un proceso cliente, ver a continuación) que

ejecuta el código de manera asincrónica e independientemente de todos los otros procesos ejecutados en los equipos cliente o servidor. En una arquitectura cliente/servidor estándar, un procedimiento almacenado ejecuta y devuelve un resultado (de manera sincrónica o asincrónica). Con 4D Server, puede iniciar un procedimiento almacenado que se ejecuta durante una sesión cliente/servidor y que responde a la demanda, a los mensajes enviados por los clientes. Simultáneamente, puede lanzar otro procedimiento almacenado que no interactúe con los clientes, pero que sincronice los datos con un servidor SQL o con otro 4D Server, utilizando un plug-in de conectividad 4D u ODBC. No hay límite (excepto hardware y memoria) para el número de procedimientos almacenados que puede ejecutar simultáneamente. Un procedimiento almacenado se ejecuta en su propio proceso y por lo tanto, como cualquier proceso usuario, puede mantener su propio contexto de base de datos (selección actual). Adicionalmente, el lenguaje 4D ofrece comandos que permiten a los procesos cliente leer y escribir las variables proceso en los procedimientos almacenados (incluyendo las variables BLOB), permitiendo una comunicación eficaz y flexible entre los clientes y los procedimientos almacenados. Utilizando procedimientos almacenados, puede añadir nuevos servicios personalizados a 4D Server. Para mayor información consulte la sección **Procedimientos almacenados**.

- **Procedimientos almacenados ejecutados en cliente:** 4D Server permite, a partir de un equipo cliente o servidor, ejecutar procedimientos almacenados en uno o varios equipos clientes. Por lo tanto, puede optimizar la repartición de cargas de trabajo entre los equipos y el servidor, o construir aplicaciones utilizando las posibilidades de comunicación entre clientes. Para mayor información, consulte la sección **Procedimientos almacenados**.
- **Ruta de acceso al servidor:** la ruta de acceso a una base de datos servidor puede guardarse con una contraseña de usuario. Esta funcionalidad permite a un usuario conectarse a una base de datos en el servidor haciendo doble clic simplemente en un documento .4DLink. Para mayor información, consulte la sección **Conexión a una base 4D Server**.
- **Registro como servicio:** bajo Windows, 4D Server puede lanzarse como un servicio.
- **Sistema de backup integrado:** bajo Windows, 4D Server incluye un modulo completo de back-up de base de datos y de restauración. Este modulo permite hacer back up de una base en uso, sin tener que salir de la aplicación. Los back-ups pueden lanzarse manual o automáticamente, a intervalos regulares y sin intervención del usuario. En caso de un incidente, la restitución y/o el reinicio de la base pueden igualmente iniciarse automáticamente.
- **Backup por espejo lógico:** en aplicaciones críticas, es posible establecer un sistema de backup por espejo lógico, lo cual permite un reinicio instantáneo si ocurre un incidente en la base operacional.
- **Plug-ins de conectividad:** con los plug-ins de conectividad 4D tales como 4D ODBC Pro, 4D Server y 4D pueden acceder directamente a las bases en mini o grandes sistemas tales como ORACLE o toda fuente de datos ODBC. La información puede compartirse de forma interactiva entre estas bases. Además, 4D ofrece un driver 4D Server ODBC que permite a todos cliente ODBC conectarse y trabajar con 4D Server.

Arquitectura de 4D Server

Con su arquitectura cliente/servidor, 4D Server no sólo guarda y administra la base, también ofrece servicios a los clientes. Estos servicios funcionan a través de una red por intermedio de un sistema de peticiones y de respuestas.

Para buscar un conjunto de registros, por ejemplo, un equipo cliente envía una petición al servidor. Una vez recibida la petición, el servidor ejecuta la búsqueda en local, es decir en el equipo servidor, y cuando termina devuelve el resultado (registros encontrados).

La arquitectura de 4D Server está basada en el modelo cliente/servidor. Por muchos años, la arquitectura cliente/servidor se ha impuesto, convirtiéndose en el modelo más eficiente en bases de datos multiusuarios.

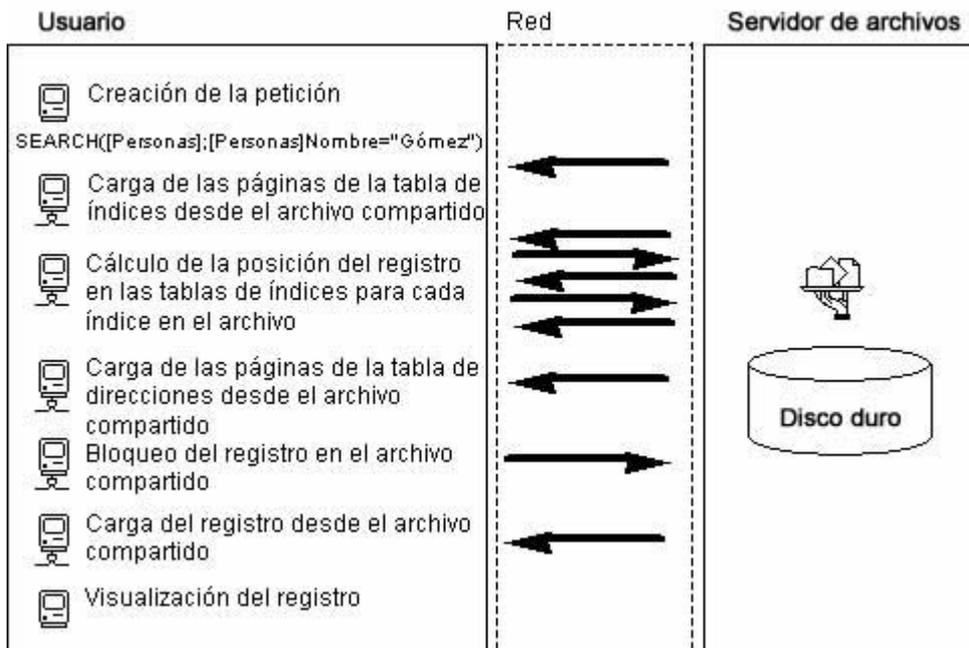
El tipo de arquitectura cliente/servidor de 4D Server es similar al utilizado en el mundo de los minicomputadores. Sin embargo, 4D Server ofrece dos innovaciones importantes:

- Una interfaz intuitiva y gráfica, presente en todos los niveles de la base
- Una arquitectura integrada que ofrece más eficiencia y velocidad

Arquitectura de compartimiento de archivos

Antes de la aparición de la arquitectura cliente/servidor, los sistemas multiusuario utilizaban el compartimiento de archivos como modelo de arquitectura de red. En este modelo, todos los usuarios comparten los mismos datos, pero la gestión de datos no es controlada por un motor de base de datos central. Cada equipo cliente debe guardar una copia de la estructura y del motor de la base, mientras el servidor se encarga de la gestión del software de compartimiento de archivos en la red.

En el modelo de compartimiento de archivos, cada estación de trabajo efectúa en local todas las acciones de modificación en los datos. Esto crea un tráfico de red importante, ya que cada petición consiste en múltiples comunicaciones a través de la red. El esquema siguiente es un ejemplo de tráfico de red generado cuando un usuario busca personas de apellido "Gómez."



Otra desventaja del modelo de compartimiento de archivos es la imposibilidad de utilizar una memoria caché para conservar los registros en memoria. Si los registros se conservan en memoria, puede haber diferentes versiones del mismo registro almacenado en la memoria caché, produciendo inconsistencia en los datos. Por lo tanto, cada vez que un usuario accede a un registro, debe ser descargado del servidor de archivos. Esto produce tráfico de red y aumenta el tiempo necesario para acceder a los datos.

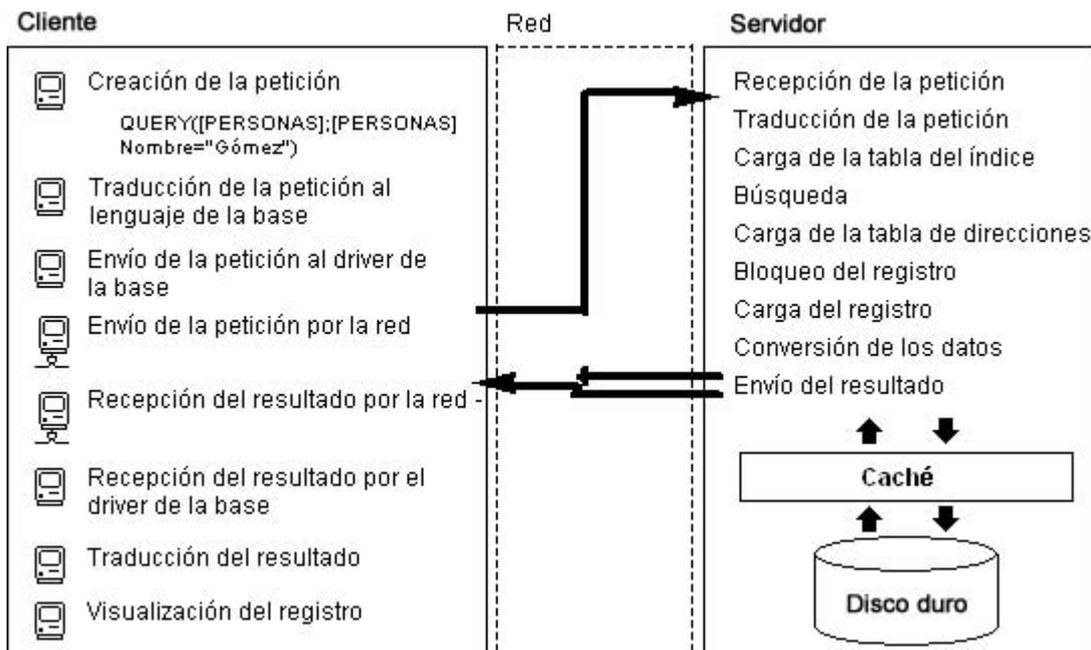
La arquitectura cliente/servidor heterogénea

La arquitectura cliente/servidor es utilizada ampliamente en el mundo de los minicomputadores, para la gestión de bases de datos muy grandes, gracias a su eficiencia y velocidad. En esta arquitectura, el trabajo se divide entre los clientes y el servidor de manera que aumente el rendimiento.

El servidor contiene el motor central de la base, que guarda y administra los datos. El motor de la base es el único software que accede a los datos almacenados en el disco. Cuando un cliente envía una petición al servidor, el servidor envía el resultado. El resultado puede ser de todo tipo desde un simple registro a modificar hasta una lista ordenada de registros.

En general, la mayoría de las arquitecturas cliente/servidor se llaman arquitecturas heterogéneas, porque las aplicaciones frontales ejecutadas en los equipos cliente y el motor de la base de datos ejecutado en el equipo servidor son dos productos diferentes. En esta situación, un driver de base de datos es necesario para servir de traductor entre los clientes y el servidor.

Para buscar un registro, por ejemplo, un cliente envía una petición al servidor. Como la base está almacenada en el servidor, el servidor ejecuta el comando localmente en el equipo servidor y envía el resultado al cliente. La siguiente imagen muestra un ejemplo de tráfico de red generado cuando un usuario busca cada persona de apellido "Smith" y muestra el primer registro encontrado.



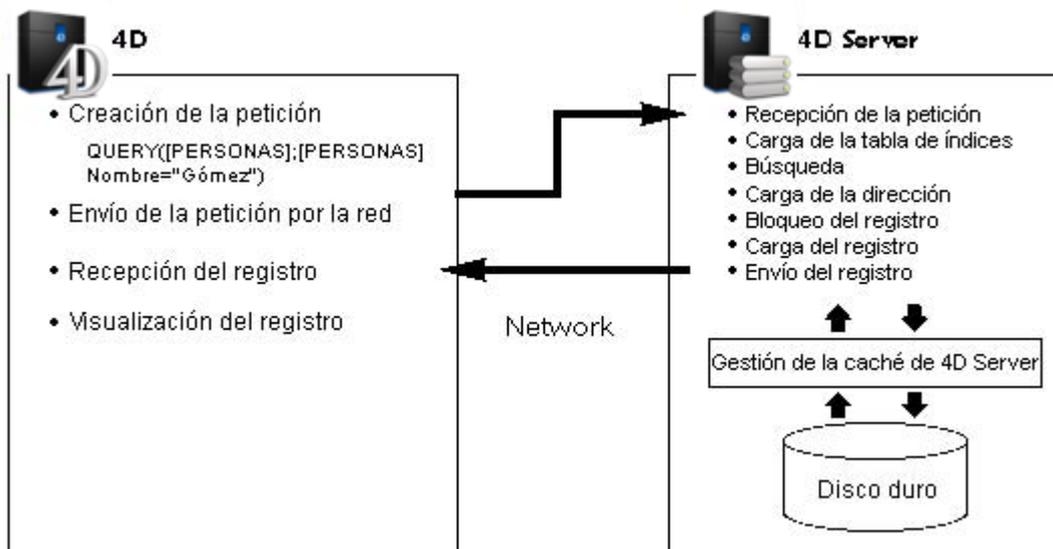
Este ejemplo muestra dos diferencias mayores entre el compartimiento de archivos y la arquitectura cliente/servidor:

- **La arquitectura cliente/servidor autoriza el uso de una memoria caché:** como el motor es el único software que accede físicamente a los datos, el servidor puede utilizar un caché que conserva en memoria los registros modificados hasta que se escriban en el disco. Como los datos se envían desde un sitio central, los equipos clientes se aseguran de recibir siempre la última versión de un registro. Adicional al control de integridad de los datos asegura, el uso de un mecanismo de caché central acelera las operaciones de base de datos reemplazando los accesos al disco por acceso a la memoria. Bajo el modelo de compartimiento de archivos, todos los accesos son de acceso disco.
- **Las operaciones de base de datos de bajo nivel se efectúan en el servidor:** la arquitectura cliente/servidor ofrece un aumento significativo en la velocidad de ejecución, como la navegación de tablas de índice y direcciones, se ejecutan localmente en el servidor, a la velocidad de la máquina. Con el compartimiento de archivos, las mismas operaciones se vuelven lentas por las transferencias de red y las limitaciones del equipo cliente.

Arquitectura cliente/servidor integrada de 4D Server

En la mayoría de las arquitecturas cliente/servidor, la aplicación cliente y la aplicación servidor son dos productos separados, que necesitan una capa de comunicación para poder entenderse entre ellos. Con 4D Server, la arquitectura cliente/servidor es totalmente integrada. 4D Server y 4D son dos aplicaciones que comparten la misma estructura y se comunican directamente.

Como 4D Server y 4D hablan el mismo idioma, no es necesario traducir las peticiones. La división del trabajo entre el cliente y el servidor es transparente y es manejada automáticamente por 4D Server.



La división del trabajo está organizada de tal manera que una petición está asociada a una respuesta. Como lo puede ver en el diagrama anterior, el cliente es responsable de:

- **Peticiones:** el cliente 4D envía las peticiones a 4D Server. Estas peticiones pueden construirse con la ayuda de los editores integrados, tales como el editor de búsquedas y el editor de ordenaciones, utilizando el lenguaje integrado de 4D o vía el SQL. 4D ofrece editores en los cuales los métodos pueden crearse y modificarse. También maneja los elementos de los métodos tales como las variables y arrays.
- **Recepción de respuestas:** el cliente 4D recibe las respuestas de 4D Server y actualiza al usuario por medio de la interfaz de usuario (los registros diferentes se muestran en un formulario, etc.). Por ejemplo, si el cliente busca todos los registros con el apellido "Gómez," 4D recibe los registros de 4D Server y los muestra en un formulario.

El servidor es responsable de las siguientes tareas:

- **Gestión de acceso:** 4D Server utiliza todas las conexiones simultáneas y los procesos creados por los clientes. Esta gestión aprovecha la arquitectura multitareas de 4D Server.
- **Objeto de estructura y de datos:** 4D Server guarda y administra todos los objetos de estructura y de datos, incluyendo campos, registros, formularios, métodos, barras de menús y listas.
- **Caché:** 4D Server mantiene una caché con los registros así como también con los objetos de datos creados por los equipos clientes, tales como selecciones y conjuntos.
- **Operaciones de base de datos de bajo nivel:** 4D Server realiza operaciones de base de datos de bajo nivel, tales como búsquedas y ordenaciones, que implican el uso de tablas de índices y direcciones.

Esta división del trabajo es extremadamente eficaz gracias a la integración de 4D Server y 4D. La integración de la arquitectura de 4D Server está presente en cada nivel:

- **A nivel de la petición:** cuando 4D envía a 4D Server una petición, tal como una búsqueda o una ordenación, 4D envía una descripción de la operación de búsqueda u ordenación utilizando la misma estructura interna que 4D Server.

- **A nivel de la estructura y de los datos:** cuando 4D y 4D Server intercambian un objeto de estructura o de datos, ambas aplicaciones utilizan el mismo formato interno. Por ejemplo, cuando 4D necesita un registro, 4D Server envía directamente los datos en el formato en el que se encuentran en el disco o en la memoria caché. De la misma forma, cuando 4D quiere actualizar un registro y envía los datos a 4D Server, el cual almacena los datos directamente en la caché tal como los recibe.
- **A nivel de la interfaz de usuario:** cuando 4D muestra una lista de registros, el formulario utilizado muestra los registros que juegan un papel en la arquitectura cliente/servidor. Por ejemplo, la siguiente imagen muestra el resultado de una petición en la tabla [Clientes].

Custo_ID	Name	Phone	Fax	City_ID
ALL	Allinone	+358 9 564		HEL
HYP	Hyperbureau	+33 01 456		PAR
MET	Metaltip	+39 02 789		MIL
ELCO	El Computador	+34 91 147		MAD
NIP	Nippon United	+81 3 258		TOK
GOR	Gorky Town	+00011144		MOS
HUI	Hychen Union	+654 3210		LUX
GOU	Gourdin Industries, Inc	+555 111 5		WAS
GLU	Glurp Technologies	+00 44777		MIL
KOA	Koala Enterprises	+6545454		MEL
BRO	Broceliande	+789456		VAR
KLI	Klick	+666 555 4		MIA

Como el tamaño de la ventana sólo permite mostrar doce registros y cinco campos a la vez, 4D Server envía exactamente doce registros. En lugar de enviar la totalidad de los registros, 4D Server envía sólo el número de registros y campos que pueden mostrarse en la ventana. Si el usuario se desplaza por el formulario, 4D Server envía los registros adicionales o campos necesarios. Esta optimización reduce el tráfico de red asegurando que los registros y campos se envíen sólo cuando sea necesario.

4D Server en 10 minutos

-  Verificación de la instalación
-  Crear una base servidor
-  Conexión a la base servidor con un 4D remoto
-  Definir la estructura de la base
-  Procesamiento de datos con 4D Server
-  Crear una barra de menús personalizada
-  Trabajar simultáneamente con varios 4D remotos

Verificación de la instalación

El capítulo de iniciación **4D Server en 10 minutos** permite descubrir rápidamente 4D Server, particularmente le mostrará cómo:

- Crear una base servidor
- Conectar un cliente a la base servidor
- Crear la estructura de la base, incluyendo tablas, campos, formularios, menús y métodos
- Conectar un segundo cliente y trabajar simultáneamente

Para realizar estos ejercicios necesitará de al menos dos computadores:

- un computador con 4D Server, 4D y un navegador web
- otro ordenador en el cual esté instalado 4D

Antes de comenzar a trabajar por primera vez con 4D Server y 4D en modo local, es buena idea verificar su instalación. Para hacerlo, lea esta sección.

Elementos instalados

Este párrafo especifica la ubicación de los elementos instalados en su disco luego de una instalación estándar de 4D + 4D Server.

Windows

Los elementos se instalan en la carpeta **Program Files\4D\4D vXX** y aparece en el menú **Inicio**.

- **4D Server**: esta carpeta contiene la aplicación 4D Server como también los archivos y carpetas asociadas. Para lanzar 4D Server, simplemente haga doble clic en el archivo *4D Server.exe*.
- **4D**: esta carpeta contiene la aplicación 4D así como los archivos y carpetas asociadas. Para lanzar 4D, simplemente haga doble clic en el archivo *4D.exe*.

Mac OS

Los elementos se instalan en la carpeta **Aplicaciones:4D:4DXX** y aparecen en las aplicaciones.

- **4D Server**: software (paquete) de 4D Server. Para lanzar 4D Server, simplemente haga doble clic en este paquete.
- **4D**: 4D software (paquete) de 4D. Para lanzar 4D, haga doble clic en este paquete.

Para este ejercicio en particular, debe instalar 4D en un equipo adicional.

¿A dónde ir ahora?

Note que el protocolo TCP/IP debe estar correctamente configurado para que sus máquinas puedan comunicarse por la red.

Si 4D Server y 4D se instalan correctamente, continúe con la sección **Crear una base servidor**. De lo contrario, si falta alguno de los archivos listados anteriormente, consulte la Guía de instalación y proceda a la instalación de los archivos.

Crear una base servidor

Esta sección describe la creación de una base servidor a la cual puede conectarse por la red utilizando 4D en modo remoto. Antes de trabajar con 4D Server y 4D por primera vez, es buena idea verificar su instalación. Para hacerlo, lea la sección **Verificación de la instalación**

Nota: en este ejemplo, asumimos que usted activó su licencia 4D Server, como se describe en la Guía de instalación. El empleo de 4D en modo remoto no necesita de una licencia en el equipo cliente. Las licencias son generadas en la máquina 4D Server. Para mayor información, consulte la Guía de instalación.

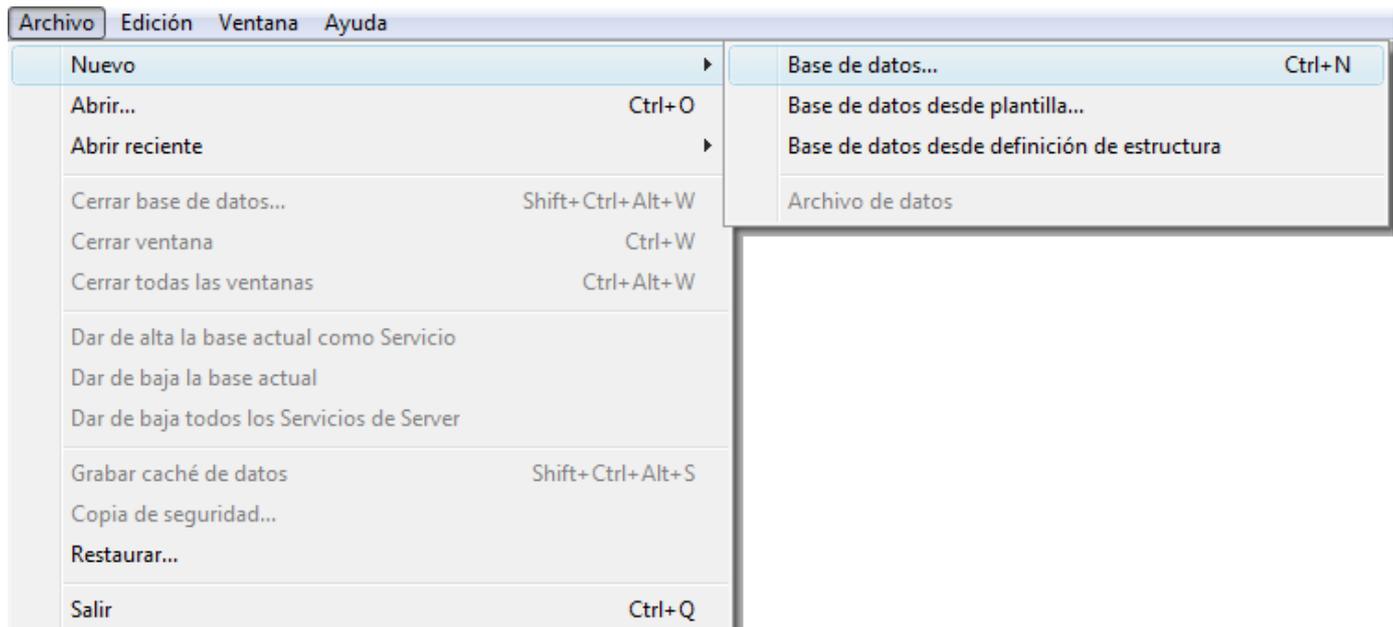
Para crear o abrir una base servidor, lance 4D Server.

1. Lance 4D Server haciendo doble clic en el icono del programa.



Cuando se lanza por primera vez la aplicación, se muestra la caja de diálogo de activación de la aplicación. Posteriormente, se mostrará una ventana vacía al inicio. Este funcionamiento puede configurarse utilizando las Preferencias de la aplicación. En este ejercicio, crearemos una nueva base vacía.

2. Seleccione el comando Nueva base de datos... en el menú Archivo de 4D Server.

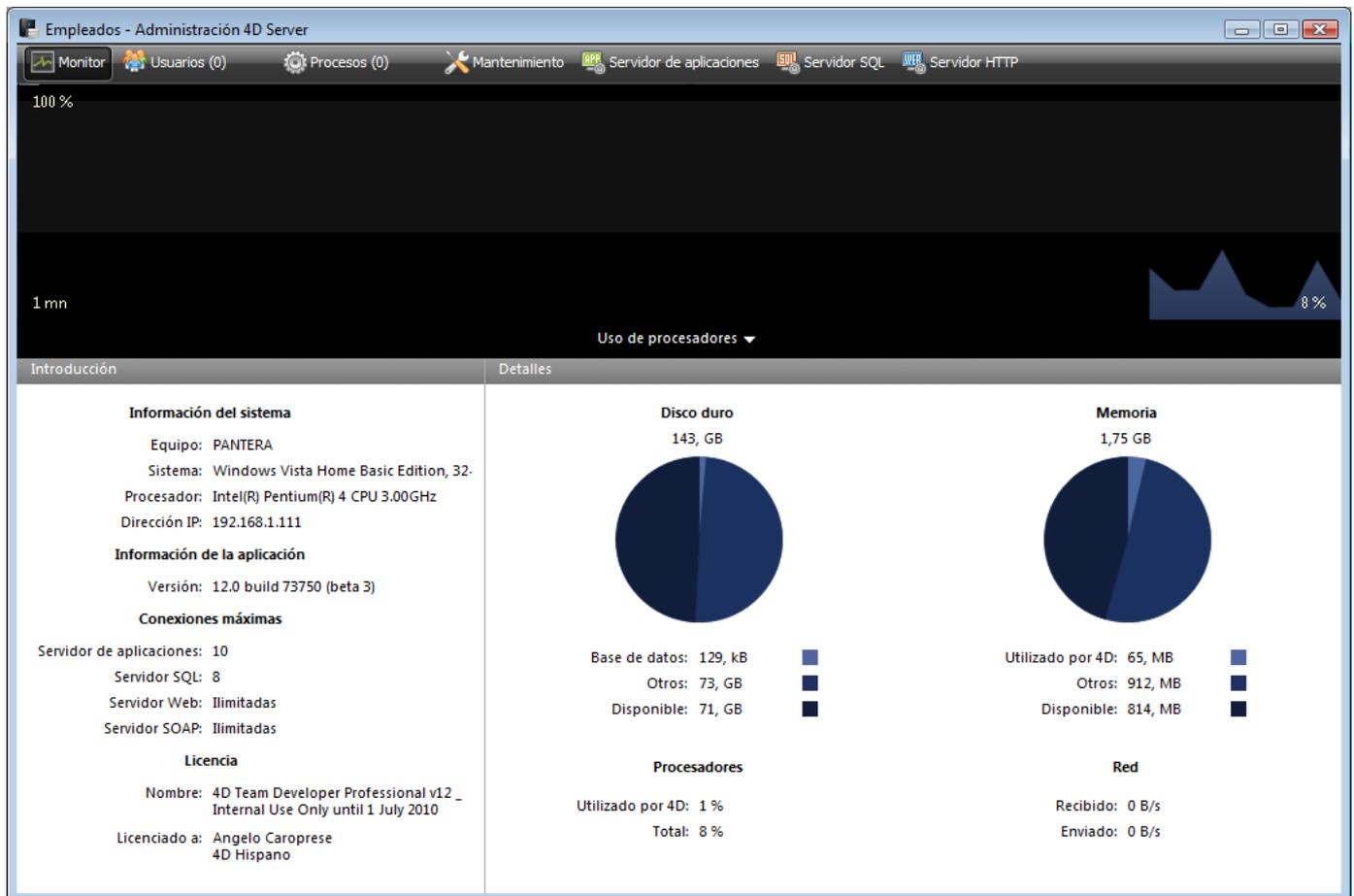


Aparece una caja de diálogo estándar de guardar archivo, la cual le permite definir el nombre y la ubicación de la base a crear.

3. Defina una ubicación, luego ingrese el nombre de su base.

Digite **Empleados**, luego haga clic en **Guardar**.

4D Server crea automáticamente los archivos y carpetas necesarias para la operación de la base, luego aparece la ventana de administración:



La **ventana de administración** de 4D Server se compone de varias páginas, accesibles utilizando pestañas. La página **Monitor** muestra información dinámica relativa a la operación de la base de datos así como también información sobre el sistema y la aplicación 4D Server.

Entre las otras páginas, note que las páginas **Usuarios** y **Procesos** indican respectivamente el número de usuarios conectados a la base y el número de procesos en ejecución actualmente. Note que en este punto, el número de usuarios conectados es cero. Esto significa que actualmente ningún cliente está conectado a la base. El número de procesos en curso de ejecución es doce. Estos doce procesos se crean automáticamente por el motor de la base y los servidores integrados de 4D Server (servidor de aplicación, servidor HTTP, servidor SQL).

¿A dónde ir ahora?

En este punto, la base está disponible para las conexiones distantes 4D Windows y/o Macintosh. Sin embargo, la base no está lista para las conexiones HTTP, porque estas conexiones no están autorizadas por defecto.

Consulte la sección **Conexión a la base servidor con un 4D remoto**. En este tutorial, primero se conectará utilizando un 4D remoto, definirá la estructura de la base y añadirá algunos registros a la base.

Conexión a la base servidor con un 4D remoto

Esta sección trata los siguientes temas:

- Cómo conectar un 4D remoto a la base servidor que usted creó.
- Cómo crear la estructura de la base. Esta sección incluye un tutorial en el cual usted crea tablas y campos en la base, introduce nuevos registros y modifica registros existentes.
- Cómo conectar un segundo usuario.
- Cómo trabajar simultáneamente con los dos clientes remotos.

Conexión a la base

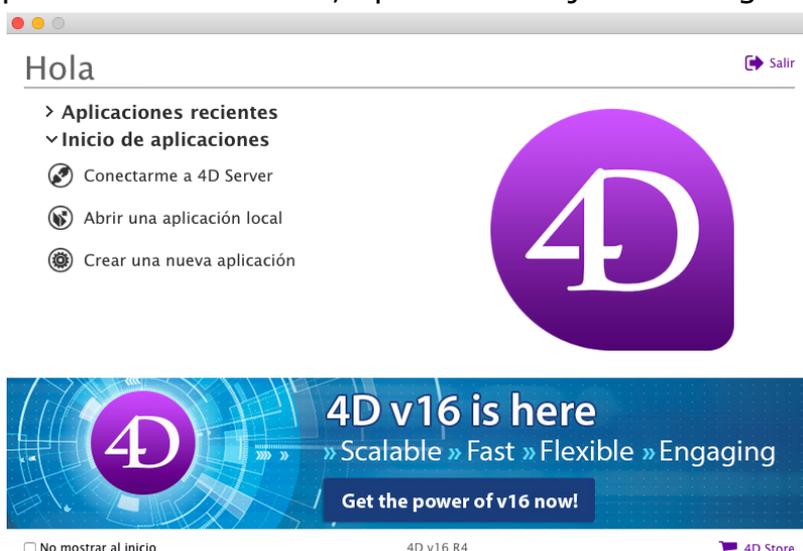
Aunque creó la base con 4D Server (ver la sección **Crear una base servidor**), todas las modificaciones en el diseño y los datos de la base se realizan desde equipos clientes. En esta sección, aprenderá cómo conectarse al servidor y a abrir la base servidor.

1. Doble clic en el icono de la aplicación 4D remota.



Nota: para este tutorial, puede utilizar una aplicación 4D instalada en la misma máquina que 4D Server.

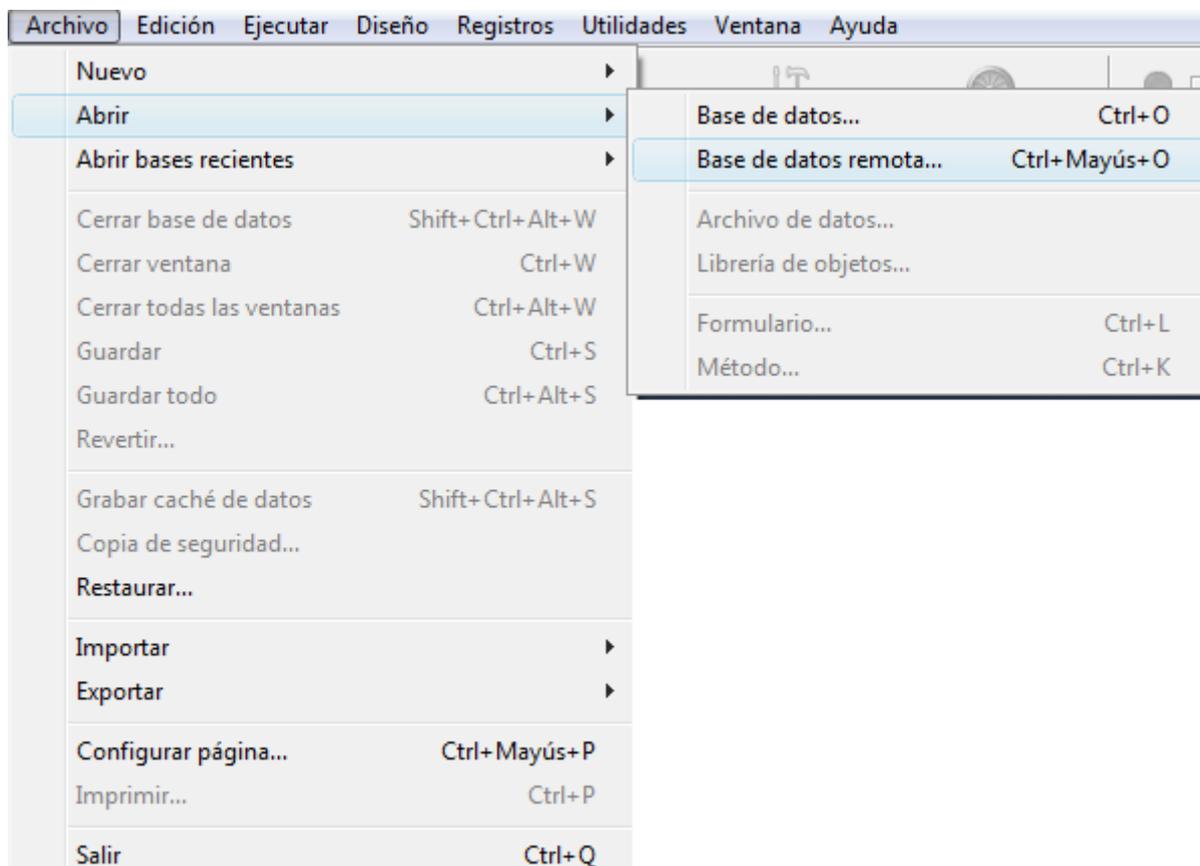
Si este es el primer lanzamiento de la aplicación 4D o si no ha modificado los parámetros de inicio, aparece la caja de diálogo de bienvenida:



2. Haga clic en "Conectarse a 4D Server".

O:

Si no aparece esta caja de diálogo, elija el comando Abrir>Base de datos remota... en el menú Archivo de 4D:



Se muestra la caja de diálogo de conexión, por defecto muestra los servidores utilizados recientemente (la lista está vacía la primera vez que se usa:



3. Haga clic en la pestaña TCP/IP para mostrar la lista de bases 4D publicadas en la red.

La base Empleados debe aparecer en la lista:



4. Seleccione Empleados y haga clic en Aceptar.

La base se abrirá en el equipo remoto, en modo Diseño, listo para crear la estructura de su base.

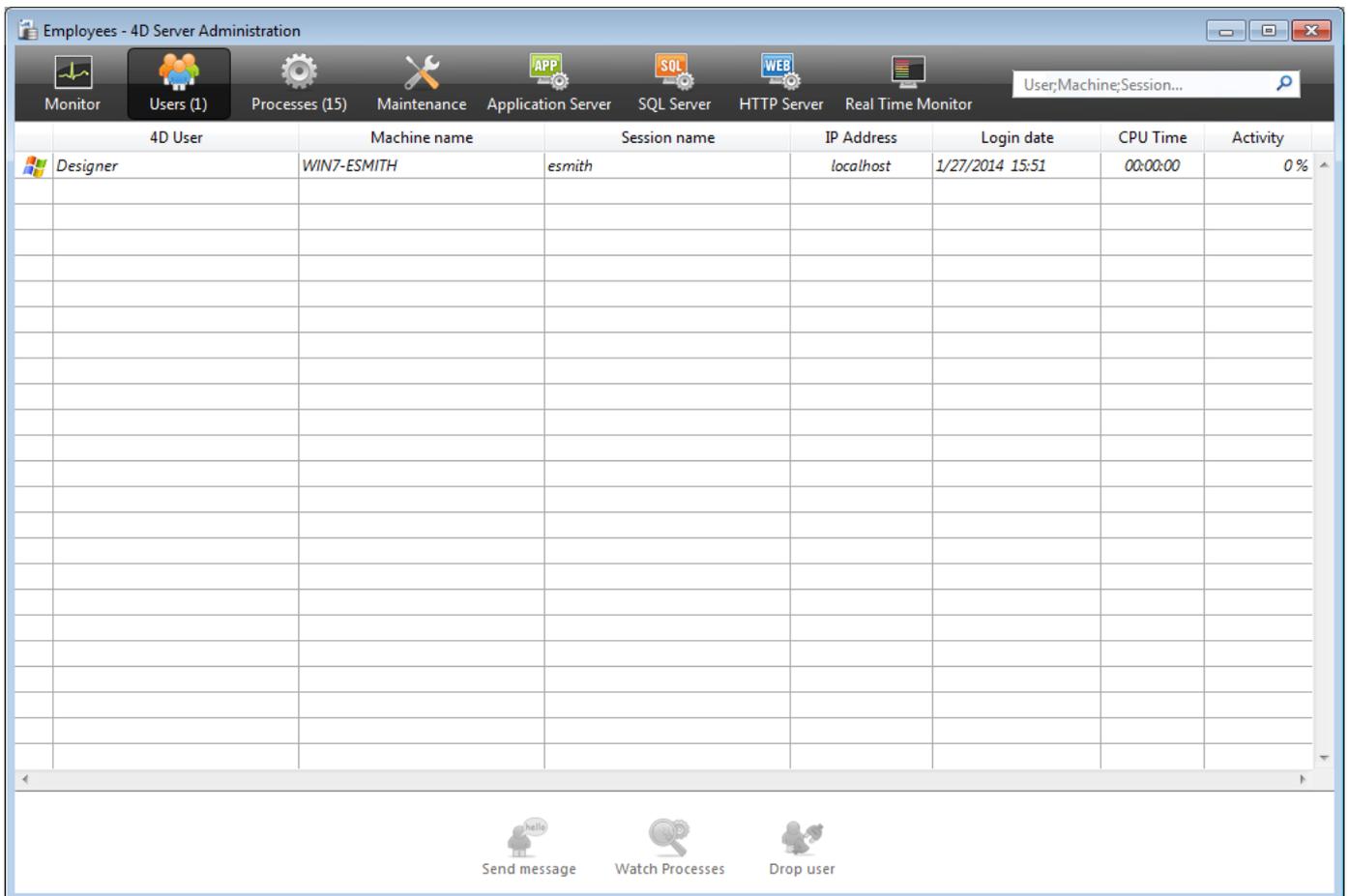
Recomendaciones

Si no ve el nombre de la base que acaba de crear con 4D Server, revise los siguientes puntos:

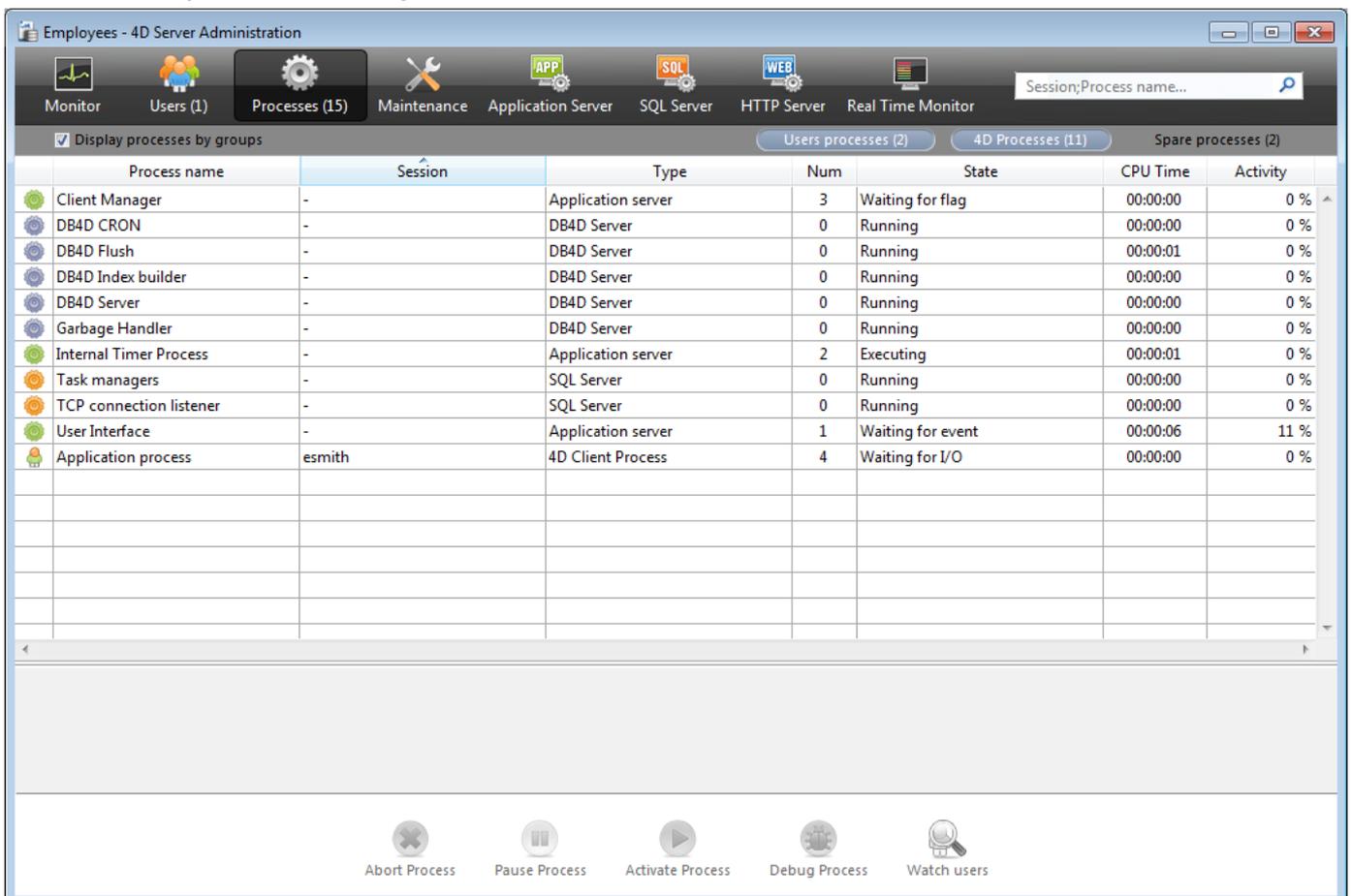
- ¿4D Server se está ejecutando en otro equipo?
- Si utiliza un segundo equipo, están las dos máquinas conectadas a la red?
- ¿Está el protocolo TCP/IP configurado correctamente en ambas máquinas?
- Si no está seguro sobre el uso de la caja de diálogo de conexión, consulte la sección **Conexión a una base 4D Server**.

Actividad del servidor

Cuando mira la ventana de administración de 4D Server, puede notar que su nombre de usuario aparece en la página correspondiente, y que el número de usuarios conectados es uno (1).



Los nuevos procesos en ejecución:



Nota: por defecto, como en la imagen anterior, los procesos en espera están ocultos. Puede hacer clic en el botón **Procesos en espera** para mostrarlos. Los primeros procesos son necesarios para el funcionamiento del servidor y se crearon durante el lanzamiento de 4D Server. Los procesos añadidos resultan de la

conexión del primer usuario al servidor:

- el proceso principal administra la visualización de los registros y el modo Aplicación.
- el proceso diseño administra el modo Diseño.

Cada usuario adicional provocará la creación de varios procesos en la lista.

Puede filtrar la lista de procesos mostrados utilizando los botones **Procesos usuarios**, **Procesos 4D** y el área de filtro ubicada en la parte superior derecha de la ventana de administración.

¿A dónde ir ahora?

Ahora que está conectado, puede trabajar con la base con las mismas funcionalidades que utiliza 4D en modo local. Primero, debe definir la estructura. Consulte la sección **Definir la estructura de la base**.

Definir la estructura de la base

En el equipo 4D remoto, luego de conectarse a la base servidor (ver la sección **Conexión a la base servidor con un 4D remoto**), elija el comando **Database Structure** en el menú **Diseño**.

Aparece la ventana de estructura, vacía por defecto. Vamos a crear una simple tabla.

Crear la tabla [Empleados]

1. Elija Nueva>Tabla en el menú Archivo o en la barra de herramientas de 4D.

O

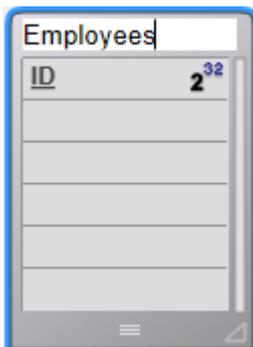
Haga clic derecho en la ventana de estructura y elija Añadir tabla en el menú contextual.

O

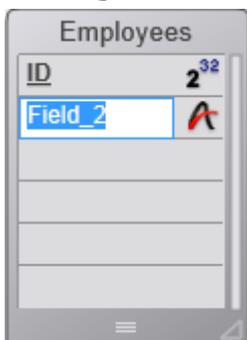
Haga clic en el botón añadir de la ventana de estructura (en forma de +) y elija Tabla.

Se crea la tabla.

2. Haga clic en el área de título y nombre a la tabla Empleados.



3. Haga doble clic en el área de campos para crear un nuevo campo.

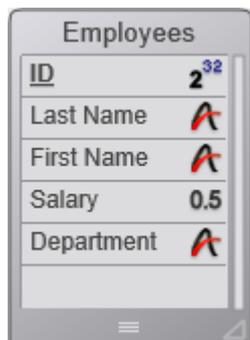


4. Renombre el campo Apellido y conserve el tipo Alfa (255).

Puede hacer doble clic en el campo para mostrar la paleta del Inspector.

5. Añada de la misma forma los campos siguientes en la tabla [Empleados]:

Nombre del campo	Tipo de campo
Nombre	Alfa (255 caracteres)
Salario	Real
Departamento	Alfa (255 caracteres)

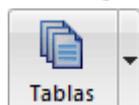


Nota: si otras aplicaciones 4D remotas trabajan simultáneamente con la misma base, los campos que acabó de crear aparecerán en el otro equipo cliente en unos momentos. Los cambios se implementan en el servidor en tiempo real, pero no aparecen en las otras pantallas inmediatamente, para evitar actualizar la pantalla con mucha frecuencia.

Crear formularios para la tabla [Empleados]

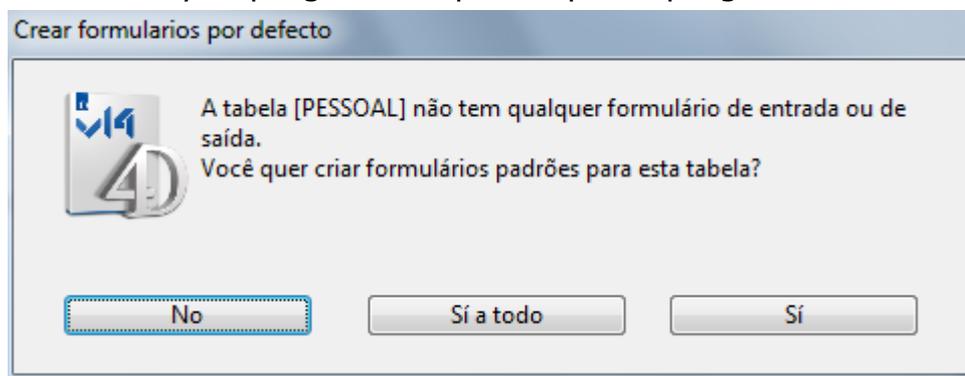
Una vez haya definido el contenido de la tabla [Empleados], debe crear los formularios para poder añadir y trabajar con sus registros. Para hacerlo, puede utilizar el Asistente de creación de formularios y crear los formularios a su conveniencia. Sin embargo, 4D ofrece un recurso práctico para crear rápidamente formularios de entrada y salida por defecto.

1. Haga clic en el botón Tablas de la barra de herramientas de 4D.



O Elija el comando Mostrar la tabla actual en el menú Registros.

Se muestra la ventana de registros. 4D detecta que la tabla no dispone de ningún formulario y le pregunta si quiere que el programa los cree por usted.



2. Haga clic en Sí.

Ahora tiene un formulario de entrada para añadir o mostrar registros uno por uno y un formulario de salida para mostrar o introducir múltiples registros en modo listado.

¿A dónde ir ahora?

Su base servidor está lista para la manipulación de datos. Vaya a la sección **Procesamiento de datos con 4D Server**.

Procesamiento de datos con 4D Server

En la sección **Definir la estructura de la base**, usted creó la tabla *[Empleados]* y le permitió a 4D crear los formularios por defecto para esta tabla. Ahora está listo para introducir registros.

Introducir los registros

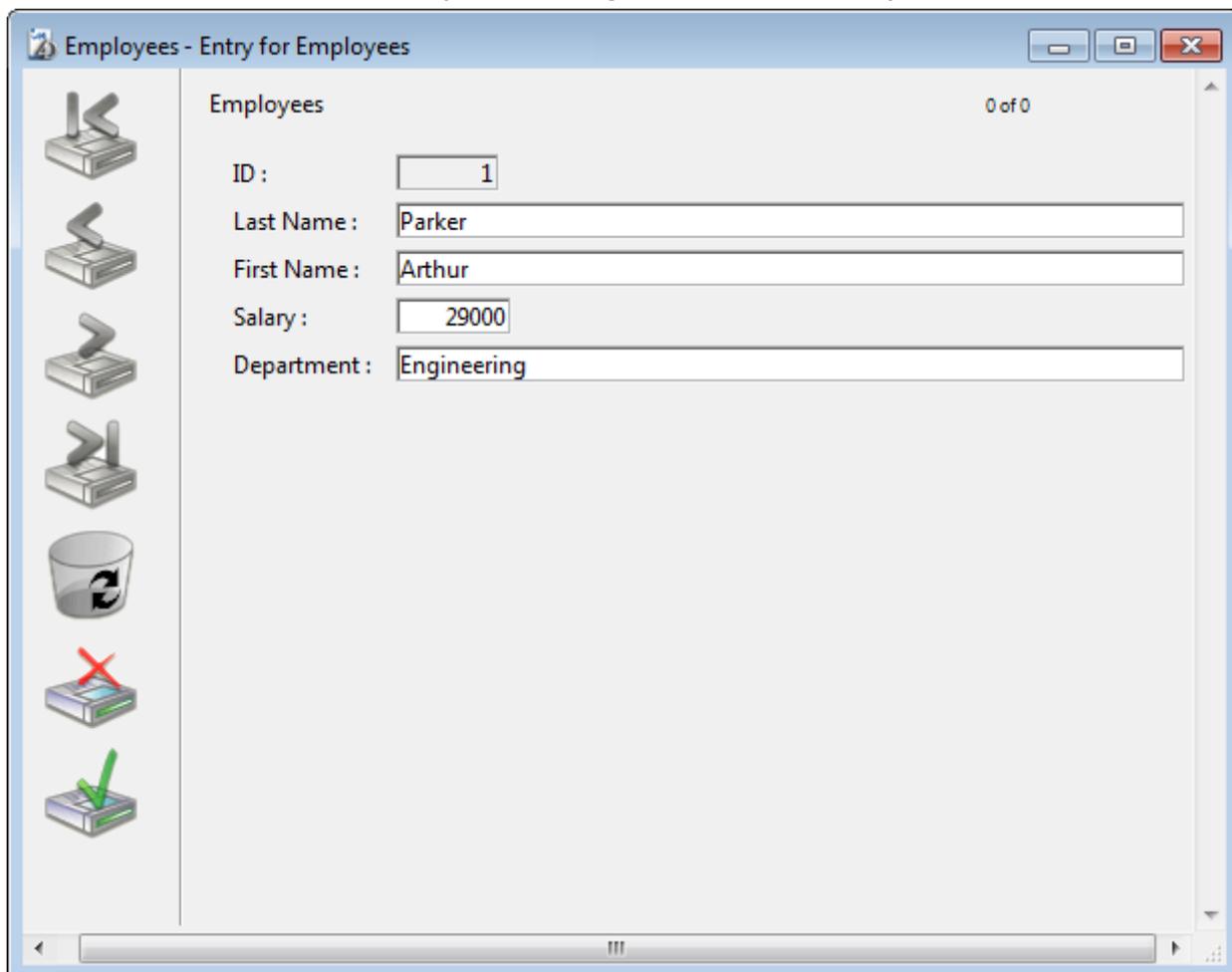
En modo Diseño, 4D le ofrece por defecto las herramientas y editores para la entrada, búsqueda, impresión o modificación de registros. Por lo tanto, puede definir sus propias herramientas para el modo Aplicación.

1. Elija Nuevo registro en el menú Registros.

Aparece el formulario de entrada vacío.

2. Introduzca su primer registro.

Use la tecla **Tab** o el ratón para navegar entre los campos.



The screenshot shows a window titled "Employees - Entry for Employees" with a standard Windows interface (minimize, maximize, close buttons). The window content is a form for entering employee data. On the left side, there is a vertical toolbar with icons for adding, deleting, and validating records. The form fields are as follows:

Field	Value
ID :	1
Last Name :	Parker
First Name :	Arthur
Salary :	29000
Department :	Engineering

3. Haga clic en el botón de validación del formulario para validar su entrada.

Aparece un formulario de entrada vacío, permitiéndole continuar añadiendo nuevos registros.

4. Introduzca cinco registros más con los siguientes valores.

Apellido	Nombre	Salario	Departamento
Hoyos	Miguel	39500	Ingeniería
Suárez	Juan	32500	Producción
Díaz	María	43000	Ingeniería
Gómez	Benjamín	34250	Producción
Rodríguez	Luis	35000	Producción

Una vez haya introducido el último registro, haga clic en el icono de cancelación (que contiene una cruz, sobre el ícono de validación) de manera que cancele el nuevo formulario de entrada vacío. Aparece el formulario de salida.

5. Si no se muestran todos los seis registros, elija Mostrar todos en el menú Registros y redimensione la ventana o las columnas si es necesario.

Debe tener:

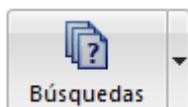
ID :	Last Name :	First Name :	Salary :	Department :
1	Parker	Arthur	29000	Engineering
2	Howard	Mike	39500	Engineering
3	Smith	John	32500	Production
4	Doe	Jane	43000	Engineering
5	Harrison	Bill	34250	Production
6	Jones	Louise	35000	Production

Ahora los registros se guardan en la base en el equipo servidor. Si un segundo 4D remoto está conectado al equipo servidor, podría mostrar los registros que acaba de añadir. Por el contrario, si los otros clientes también estaban introduciendo registros, puede elegir **Mostrar todos** desde el menú **Registros** para mostrar todos los registros, incluyendo los que ha introducido. Los registros almacenados en el servidor son accesibles a todos los usuarios.

Buscar registros

Una vez haya creado los registros en la tabla *[Empleados]*, puede efectuar búsquedas, ordenaciones, impresiones, etc. con los registros. Por ejemplo, busquemos los empleados del departamento de Ingeniería.

1. Haga clic en el botón Buscar en la barra de herramientas.



Aparece el editor de búsquedas:

2. Conserve "[Empleados]Departamento" como criterio de búsqueda y "es" en la lista de comparadores, luego introduzca "Ingeniería":

Query in [Employees]

Create new selection

Find: [Employees]Department is Engineering

Cancel Query

3. Haga clic en el botón Buscar.

La petición se envía al 4D Server, luego 4D Server responde a 4D. El formulario de entrada muestra únicamente los empleados que trabajan en el departamento de Ingeniería.

ID :	Last Name :	First Name :	Salary :	Department :
1	Parker	Arthur	29000	Engineering
2	Howard	Mike	39500	Engineering
4	Doe	Jane	43000	Engineering

Para mostrar todos los registros nuevamente, elija Mostrar todos en el menú Registros.

¿A dónde ir ahora?

En pocos minutos ha creado una base servidor, definido una tabla, añadido registros, luego efectuó una búsqueda utilizando los datos introducidos en la base. Ahora es tiempo de añadir una barra de menús personalizada para su base. Consulte la sección **Crear una barra de menús personalizada**.

Crear una barra de menús personalizada

En esta sección, usted diseñará dos métodos y una barra de menús personalizada. En resumen, va a crear una aplicación 4D.

Crear los dos métodos

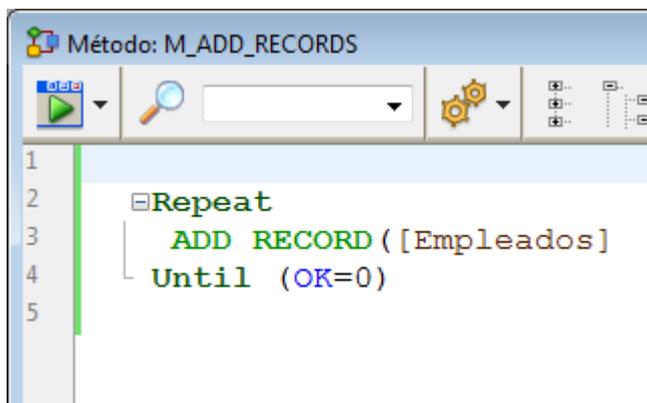
1. Seleccione **Nuevo > Método..** en el menú **Archivo**.

Aparece la caja de diálogo de creación del método.

2. Introduzca **"M_ADD_RECORDS"** en la caja de diálogo **Nuevo método**, luego haga clic en **Aceptar**.

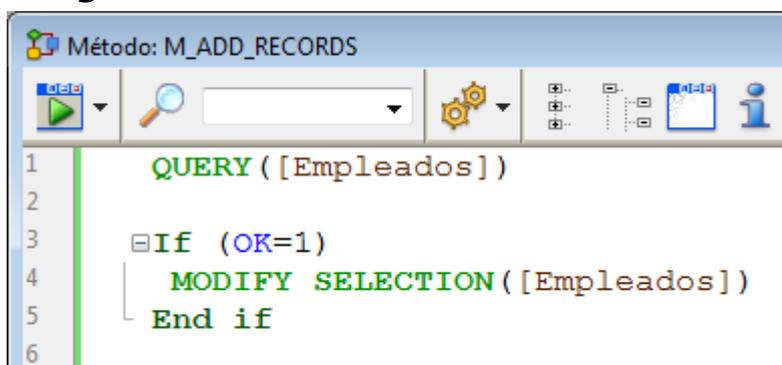
Aparece una ventana del editor de métodos llamada "Método:M_ADD_RECORDS".

3. Introduzca el código del método **M_ADD_RECORDS** como se muestra a continuación:



```
1
2   Repeat
3     ADD RECORD ([Empleados])
4   Until (OK=0)
5
```

4. Cree un segundo método llamado **"M_LIST_RECORDS"** con el siguiente código:



```
1   QUERY ([Empleados])
2
3   If (OK=1)
4     MODIFY SELECTION ([Empleados])
5   End if
6
```

Ahora que los dos métodos de han creado, cree una barra de menús personalizada y asocie los métodos a los comandos de menús.

Crear una barra de menús personalizada

1. Seleccione **Caja de herramientas > Menús** en el menú **Diseño**.

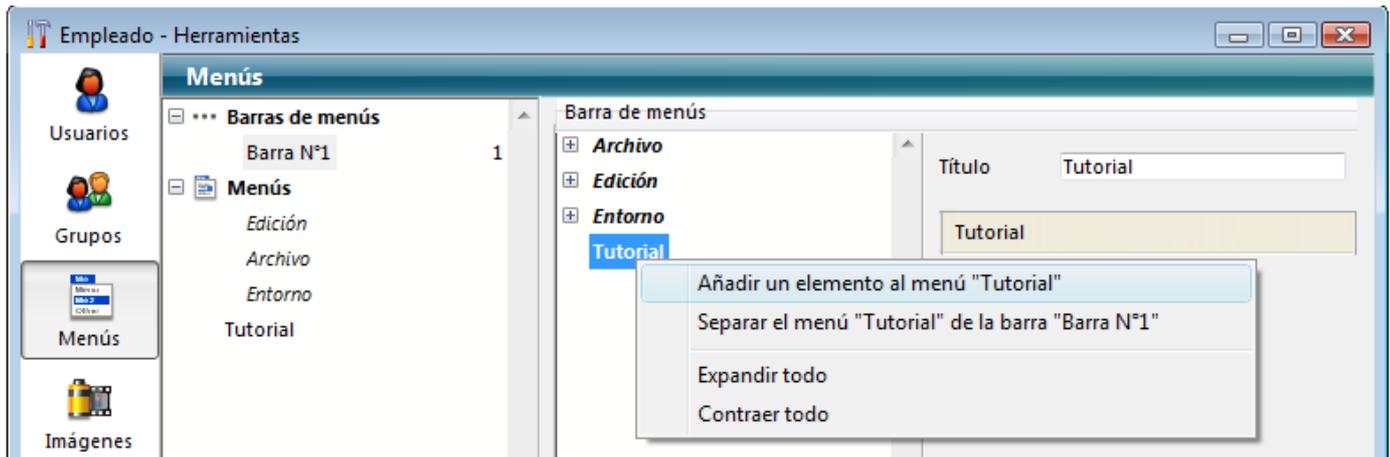
El editor de barras de menús se muestra con una barra de menús por defecto.

2. Seleccione la etiqueta "Barra # 1" haga clic en el botón añadir en la parte central de la ventana.



3. Introduzca "Tutorial" como título del menú y presione Intro.

4. Haga clic derecho en el título "Tutorial" y elija el comando Añadir un elemento al menú "Tutorial":

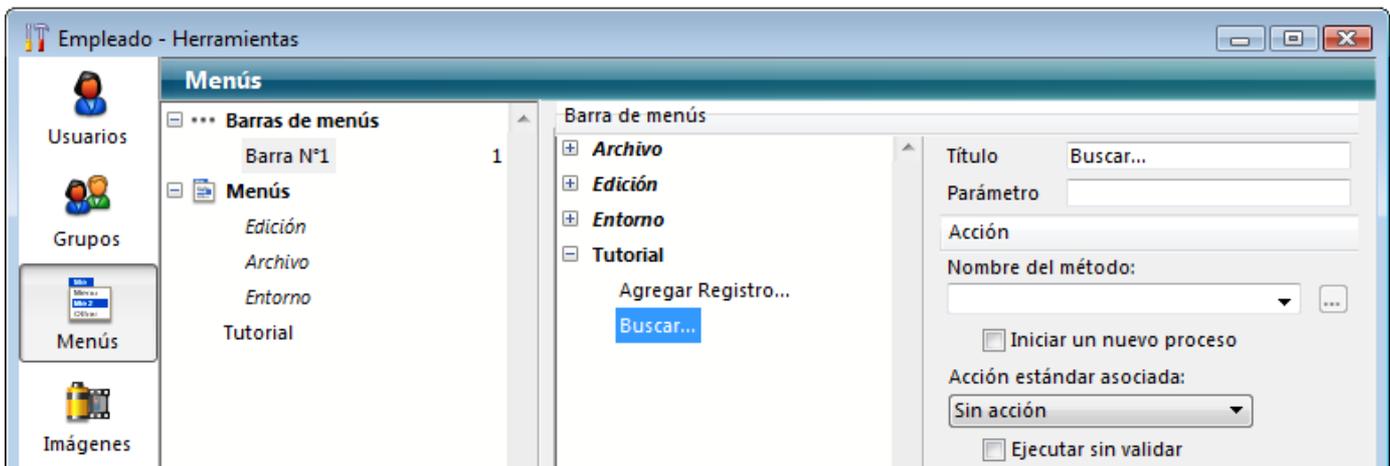


5. Introduzca "Añadir registros..." y presione Intro.

6. Haga clic derecho nuevamente en el título "Tutorial" para añadir un segundo comando de menú al menú "Tutorial".

7. Introduzca "Buscar..." y presione Intro.

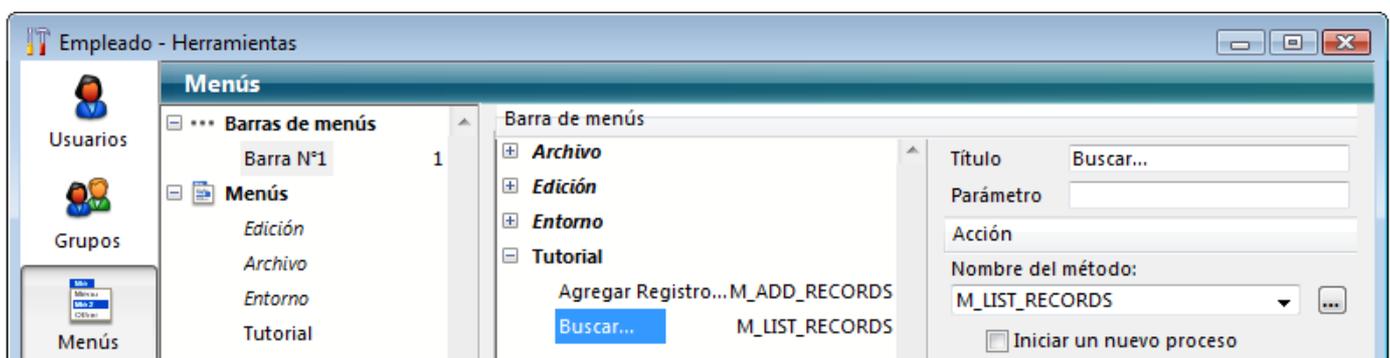
La barra de menús #1 debe verse así:



8. Haga clic en el comando de menú "Añadir registros..." y seleccione "M_ADD_RECORDS" en el combo box Nombre del método.

9. Haga clic en el título "Buscar..." y seleccione "M_LIST_RECORDS" en el combo box Nombre del método.

La barra de menús #1 ahora se ve así:

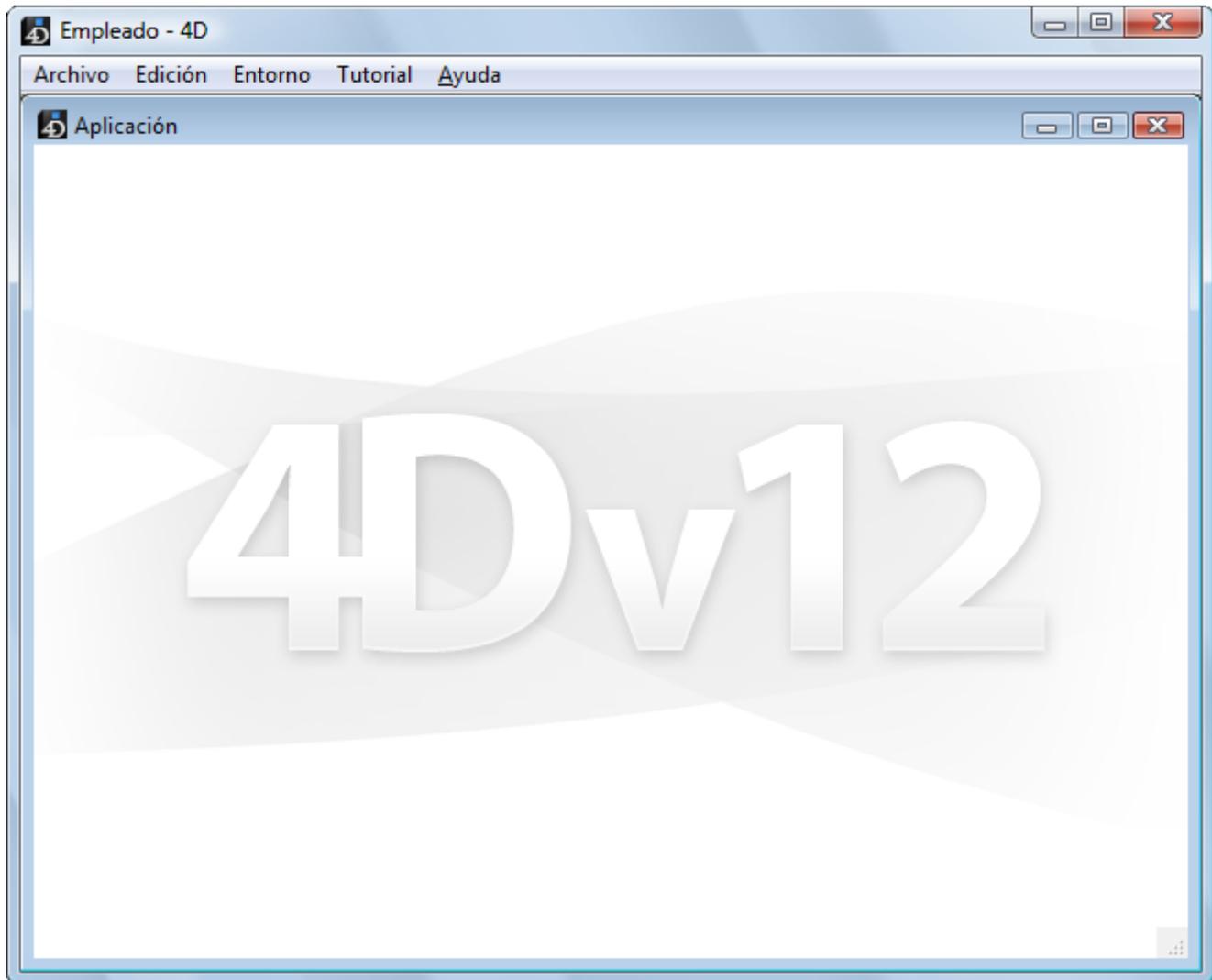


10. Cierre la ventana de la caja de herramientas.

¡Listo!

11. Seleccione probar la aplicación en el menú Ejecución.

Ahora puede utilizar la aplicación con los menús que diseño:



Por ejemplo, si selecciona **Buscar...** en el menú **Tutorial**, aparece el editor de búsquedas estándar de 4D. Puede definir su búsqueda, luego mostrar y modificar los registros encontrados.

El punto interesante, es que sin saberlo, ¡Acaba de desarrollar dos aplicaciones!

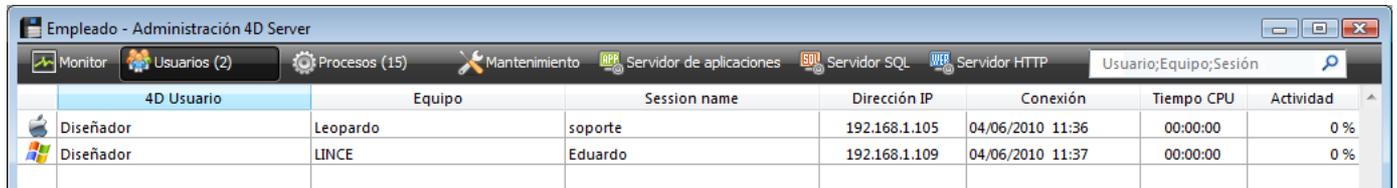
Para saber por qué, consulte la sección **Trabajar simultáneamente con varios 4D remotos**.

Trabajar simultáneamente con varios 4D remotos

Si hace los ejercicios de este tutorial en Windows, puede utilizar la base servidor "como está" en Macintosh. Si hace los ejercicios de este tutorial en Macintosh, puede utilizar esta base servidor "como está" en Windows.

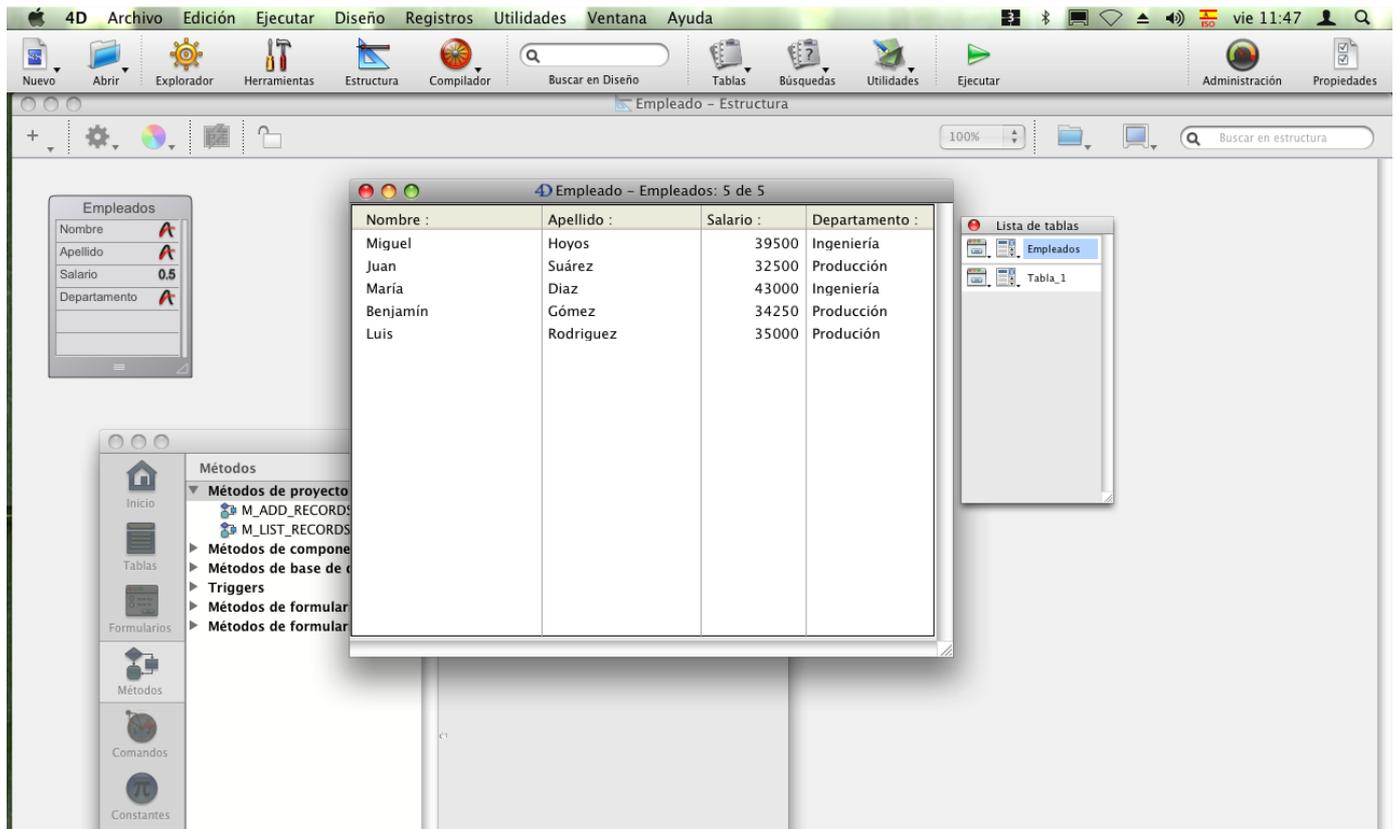
Conexión a la base servidor con un segundo usuario

Para este tutorial, nos conectaremos a la base servidor con un 4D remoto en Windows y un 4D remoto en Mac OS. Tan pronto como se establezca la conexión, podrá ver al segundo usuario en la ventana de administración de 4D Server (la primera columna indica el sistema operativo del equipo remoto):



4D Usuario	Equipo	Session name	Dirección IP	Conexión	Tiempo CPU	Actividad
Diseñador	Leopardo	soporte	192.168.1.105	04/06/2010 11:36	00:00:00	0 %
Diseñador	LINCE	Eduardo	192.168.1.109	04/06/2010 11:37	00:00:00	0 %

En cada equipo cliente, todo lo que se haga en la base es reutilizable instantáneamente y de manera transparente. Este es el entorno Diseño en el equipo cliente remoto en Mac OS:



The screenshot shows the 4D Design environment on a Mac OS client. The main window displays a table with the following data:

Nombre :	Apellido :	Salario :	Departamento :
Miguel	Hoyos	39500	Ingeniería
Juan	Suárez	32500	Producción
María	Díaz	43000	Ingeniería
Benjamín	Gómez	34250	Producción
Luis	Rodríguez	35000	Producción

On the left, a sidebar shows a list of methods under "Métodos de proyecto":

- Métodos de proyecto
 - M_ADD_RECORDS
 - M_LIST_RECORDS
 - Métodos de compone
 - Métodos de base de c
 - Triggers
 - Métodos de formular
 - Métodos de formular

On the right, a "Lista de tablas" (Table List) shows two tables: "Empleados" and "Tabla_1".

¡Sus seis registros y sus dos métodos están aquí!

Trabajar simultáneamente en los registros

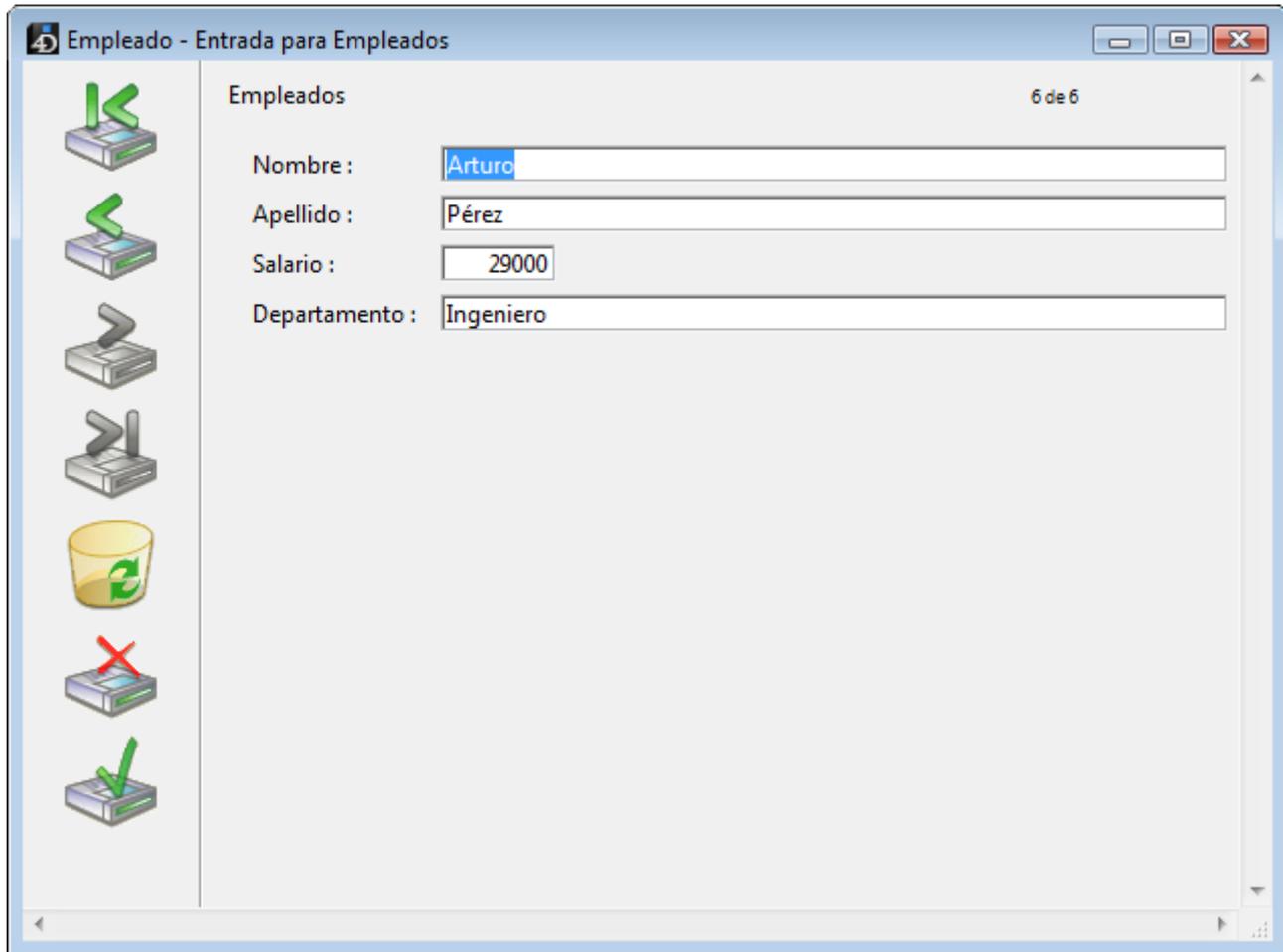
1. En el primer equipo remoto, en modo "Test Application", elija Buscar... en el menú Tutorial y busque los registros donde "Departamento es igual a Ingeniería".

2. Haga lo mismo en el segundo equipo remoto.

En ambos equipos, obtiene una lista compuesta por tres registros.

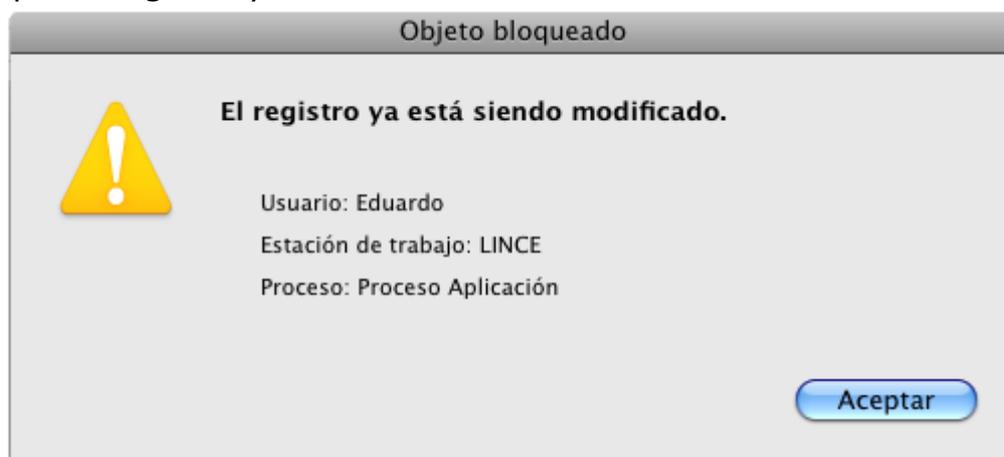
3. En la primera máquina, haga doble clic en el registro "Pérez, Arturo".

Su pantalla debe verse así:



4. Haga lo mismo en el segundo equipo.

4D Server tiene un sistema de gestión integrado de bloqueo de datos que le advierte que el registro ya está en uso:



Sin embargo, usted tiene acceso al registro en modo sólo lectura (puede verlo, pero no modificarlo).

Empleado - Entrada para Empleados 6 de 6

Empleados

Nombre : Arturo

Apellido : Pérez

Salario : 29000

Departamento Ingeniero

5. En el primer equipo, cambie el nombre "Miguel" y valide sus cambios.
 La lista se actualiza con el nuevo nombre.

Empleado - Empleados: 6 de 6

Nombre :	Apellido :	Salario :	Departamento :
Miguel	Hoyos	39500	Ingeniería
Juan	Suárez	32500	Producción
María	Díaz	43000	Ingeniería
Benjamín	Gómez	34250	Producción
Luis	Rodríguez	35000	Producción
Alonzo	Hernandez	29000	Ingeniero

6. En el segundo equipo, cancele la visualización del registro en el formulario de entrada.

¡Se ha actualizado la lista con el nuevo nombre también!



Nombre :	Apellido :	Salario :	Departamento :
Arturo	Pérez	29000	Ingeniero
Juan	Suárez	32500	Producción
María	Díaz	43000	Ingeniería
Benjamín	Gómez	34250	Producción
Luis	Rodriguez	35000	Producción
Alonzo	Hernandez	29000	Ingeniería

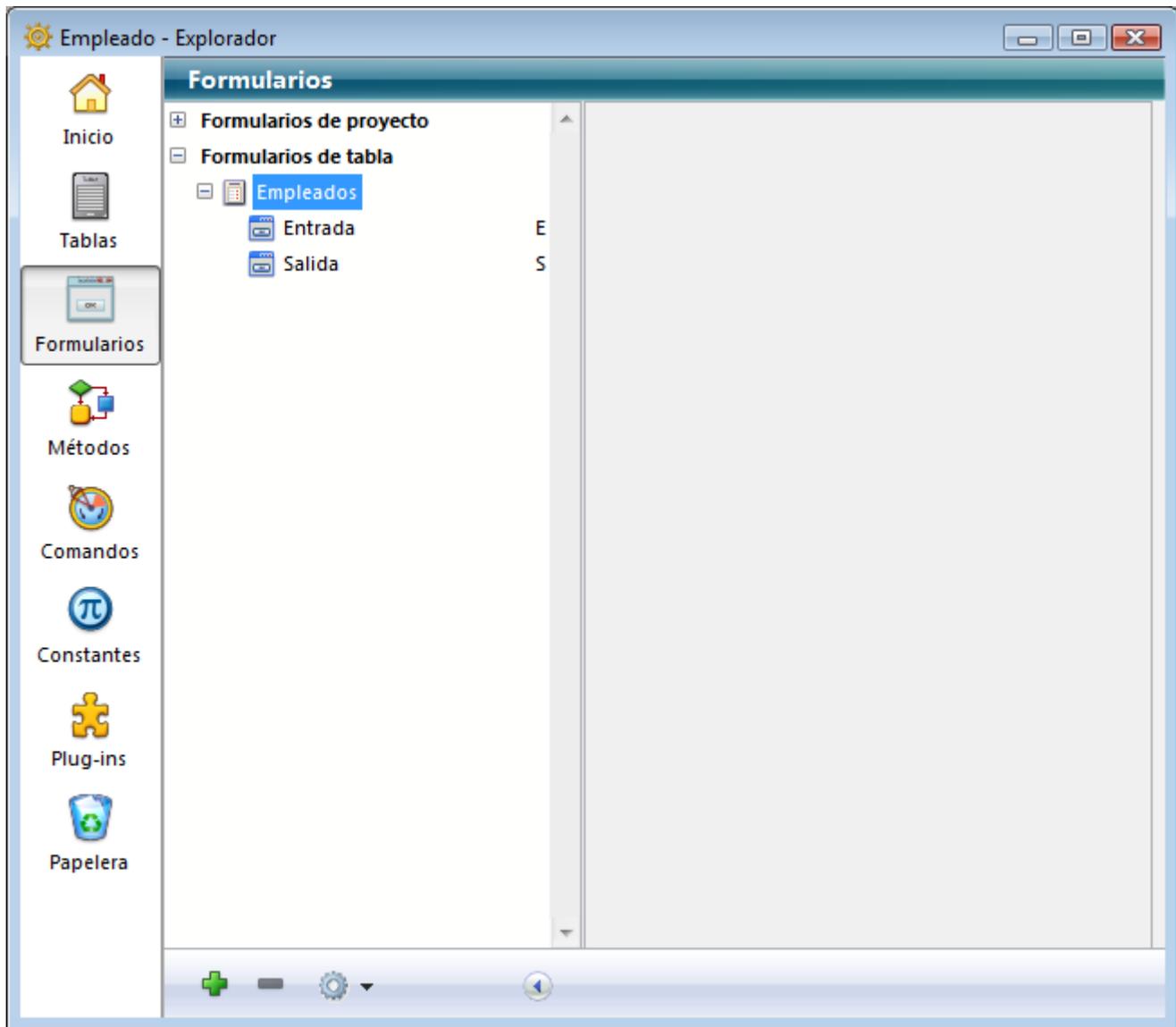
Trabajar simultáneamente con objetos de diseño

4D Server es un servidor de datos y de aplicaciones. Veamos lo que esto significa. En el segundo equipo, presione la tecla **Esc.**, luego elija el comando **Volver al modo Diseño** en el menú **Modo**.

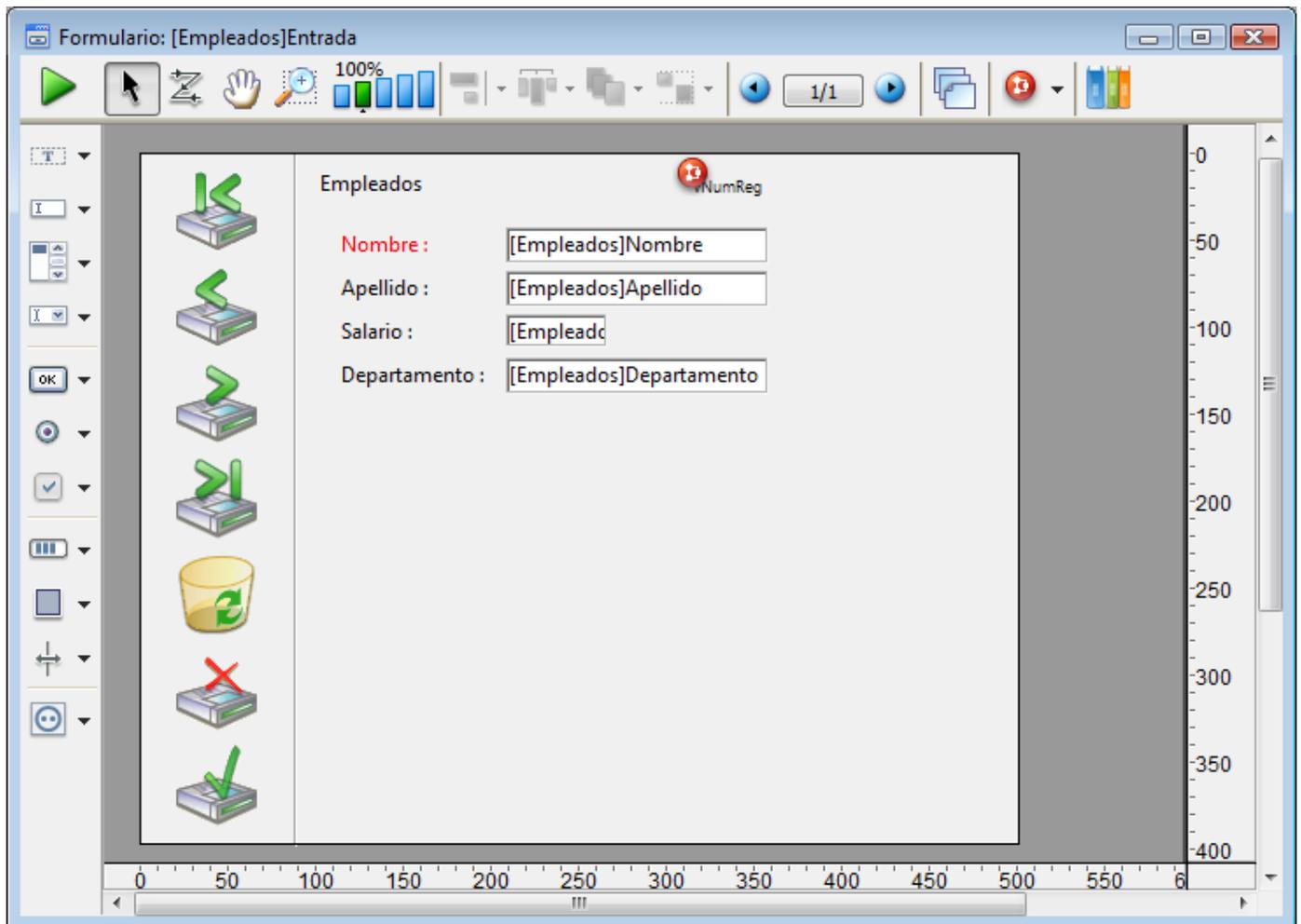
Haga lo mismo en la primera máquina.

En la primera máquina, elija **Explorador> Formularios** en el menú Diseño. Aparece la ventana del Explorador.

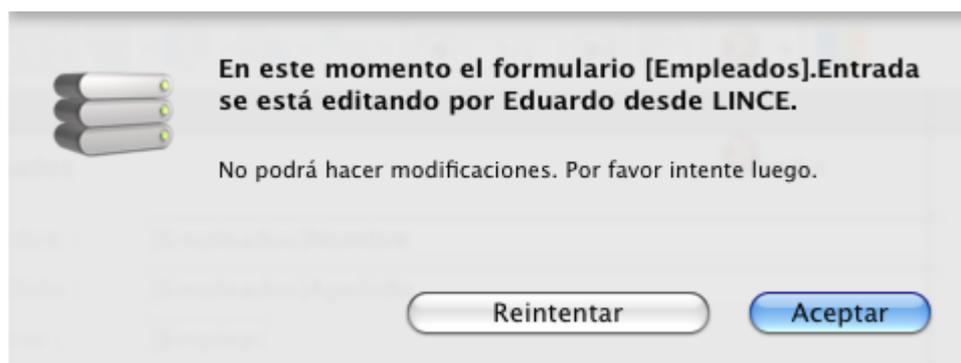
Elija los formularios tabla y despliegue la tabla Empleados:



Haga doble clic en el formulario de entrada. Aparece el editor de formularios, con el formulario de entrada:

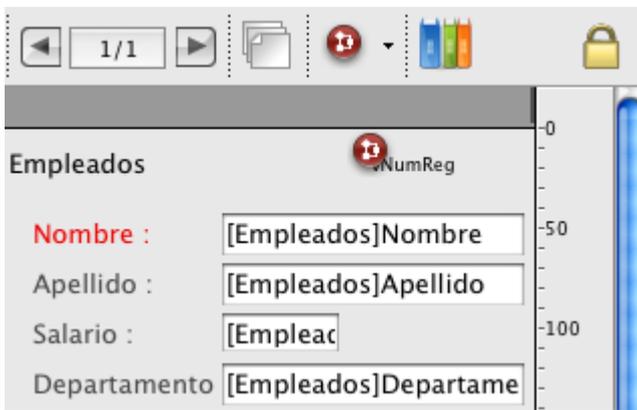


Haga lo mismo en la segunda máquina. Como el formulario está en modo edición en la otra máquina, el mecanismo integrado de bloqueo de objetos 4D Server le informa:

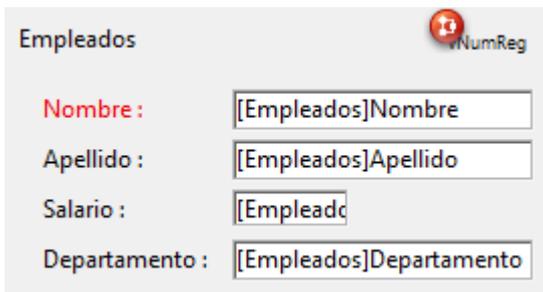


No obstante, puede abrir el formulario en la segunda máquina en el modo de visualización. Puede seleccionar objetos y copiarlos a otros formularios, pero no puede modificar el formulario mismo.

Note el icono de candado en la esquina superior derecha del formulario. Este icono le recuerda que no puede cambiar el formulario.

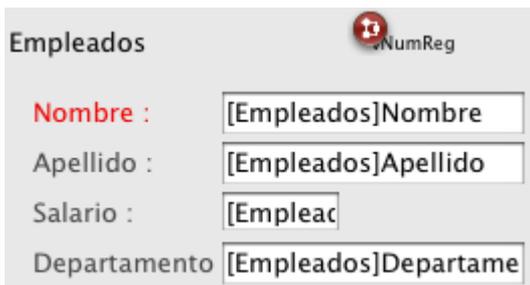


En la primera máquina, seleccione la etiqueta "Apellido" a la izquierda del campo [Empleados]Nombre. Seleccione en el menú jerárquico **Objeto>Color**. Vuelve rojo el color de este objeto.



Seleccione **Guardar el formulario: [Empleados]Entrada** en el menú **Archivo**.

En la segunda máquina, cierre y vuelva a abrir el formulario para recargarlo. El cambio realizado en la otra máquina ahora está disponible en ésta.



4D Server le permite desarrollar al mismo tiempo una base de datos con otros usuarios

Cerrar el servidor

Además de la información de los usuarios 4D remotos durante los accesos simultáneos a los mismos registros u objetos, 4D Server incluye un mensaje incorporado de advertencia de cierre a través de la red.

1. Manteniendo a los dos clientes conectados, en la máquina de servidor, elija **Salir** en el menú **Archivo** (Windows) o el menú de **4D Server** (Mac OS).

Aparece la caja de diálogo Cerrar:

2. Haga clic en **Aceptar**.

Casi instantáneamente, las dos máquinas de cliente son informadas que el servidor se está cerrando. Por ejemplo, si un cliente estaba agregando un registro, el usuario tendría suficiente tiempo para terminar y validar la entrada de datos.

Esta caja de diálogo de advertencia se repite regularmente en cada máquina de cliente.

Nota: alternativamente, puede cerrar el servidor utilizando la opción "Esperar a que todos los usuarios se desconecten" (enviándoles un mensaje que les pide desconectarse tan pronto como sea posible) o forzar la desconexión inmediata de los clientes vía la opción " Desconectar a todos los clientes y salir" .

3. Mientras el servidor cierra, cierre 4D en las dos máquinas remotas.

Conclusión

Con este tutorial, usted ha descubierto lo fácil que es utilizar 4D Server:

- Creó una base desde cero.
- Creó una tabla y le permitió a 4D Server crear los formularios por usted.
- Añadió y manipuló registros.
- Personalizó su aplicación con su propia barra de menús.
- Utilizó la base servidor simultáneamente en Windows y Macintosh.
- Cerró y reinició el servidor.

Para concluir, creó dos aplicaciones personalizadas (Windows y Macintosh) con un solo desarrollo. Además, si necesita utilizar la base de datos en el modo local, puede abrirla directamente con 4D.

Para aprender más sobre 4D Server, lea las secciones de introducción de este manual, así como las otras secciones que describen el funcionamiento de 4D Server.

Para información completa sobre el entorno 4D, consulte:

- El manual de Diseño para aprender acerca de diseñar y utilizar aplicaciones y bases de datos 4D.
- El manual de Lenguaje para aprender acerca de las comandos del lenguaje 4D. Por ejemplo para aprender acerca de la capacidad web de 4D Server, lea la sección **Presentación del servidor web** web en el manual Lenguaje de 4D.

Uso de 4D Server

-  Crear o abrir una base 4D Server
-  Salir de 4D Server
-  Menús de 4D Server
-  Opciones red y cliente-servidor
-  Configuración IP
-  Encriptar las conexiones cliente-servidor
-  Autenticación única (Single Sign On - SSO) en Windows
-  Gestión de la carpeta Resources
-  Registrar una base como servicio
-  Configurar un espejo lógico
-  Utilizar Servicio Volume Shadow Copy en Windows

Crear o abrir una base 4D Server

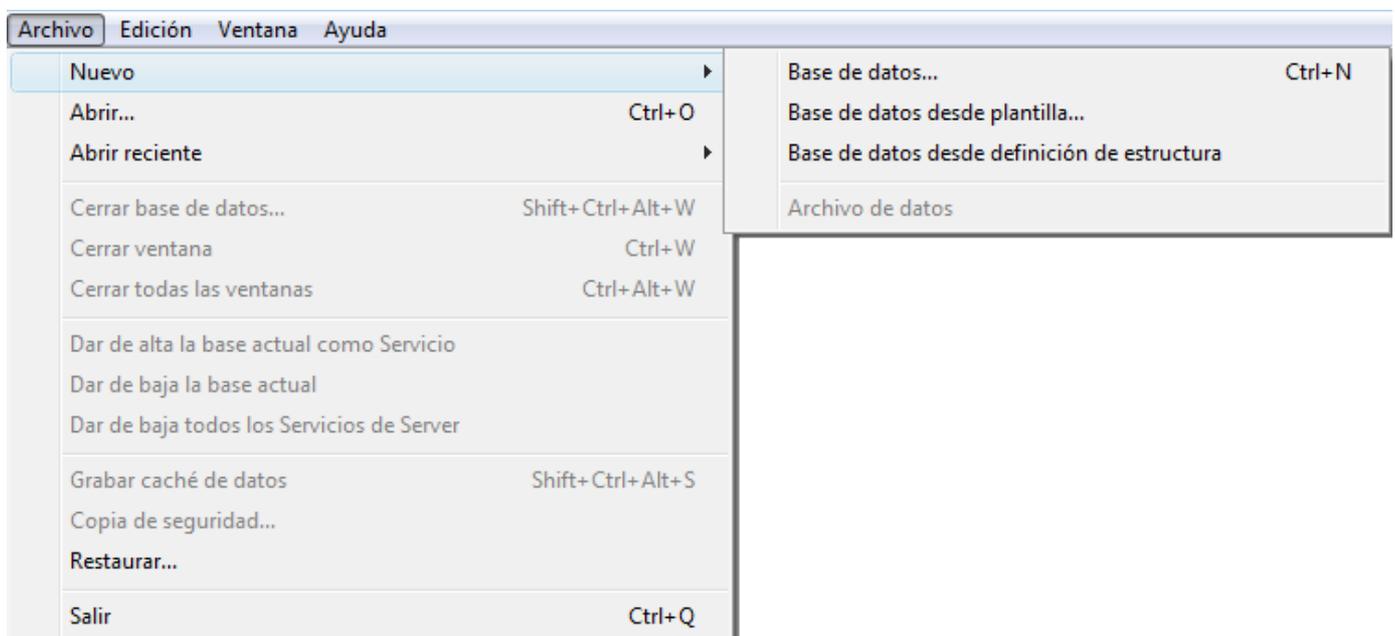
Para crear una nueva base o abrir una base existente, lance 4D Server haciendo doble clic en el icono de la aplicación.



Puede entonces crear una nueva base o abrir una base existente en el menú **Archivo** de 4D Server.

Crear una base

Para crear una nueva base, elija uno de los comandos del submenú **Nuevo>**:



- **Base de datos:** se puede utilizar para crear una base de datos vacía, en otras palabras, sin tablas, formularios o interfaz predefinidos. Cuando selecciona este comando, aparece una caja de diálogo estándar de guardar archivos para que pueda definir el nombre y la ubicación de la base.
- **Base de datos desde plantilla:** crea una base de datos utilizando una plantilla "lista para usar" que puede personalizar posteriormente. Para poder utilizar esta función, las bases plantillas deben ubicarse en una carpeta "4D Templates" o "4D Modèles" al mismo nivel que el archivo 4D Server.exe (Windows) o el paquete 4D Server (Mac OS). Cuando se selecciona este comando, aparece la caja de diálogo de selección de plantillas de bases de datos.
- **Base de datos desde definición de estructura:** permite crear una base de datos basada en una descripción de estructura en formato XML. Esta definición puede venir de una estructura exportada desde 4D o toda aplicación de diseño.

Cuando se selecciona este comando, aparece una caja de diálogo estándar de apertura de archivos, permitiéndole designar el archivo XML a utilizar.

Para obtener más información sobre estas opciones, consulte el Manual de *Diseño*.

Abrir una base

Para abrir una base de datos existente, puede usar una caja de diálogo estándar de apertura de documentos (comando **Archivo>Abrir...**) o seleccionar directamente una base abierta previamente (comando **Archivo>Abrir reciente**).

Si una base de datos ya estaba abierta cuando selecciona un comando Abrir, primero se cerrará. Si algún equipo cliente está conectado, se desconectan con el modo "Espere a que todos los usuarios se desconecten" (ver la sección **Salir de 4D Server**).

Nota: también puede abrir una base de datos existente directamente arrastrando y soltando un archivo de estructura interpretado o compilado (.4db o .4dc) o de un archivo de acceso directo (.4 dlink) en el icono de la aplicación 4D Server.

Salir de 4D Server

Para cerrar 4D Server:

1. Seleccione el comando Salir en el menú Archivo de 4D Server (Windows) o en el menú 4D Server (OS X).

La siguiente ventana se muestra en el equipo servidor:



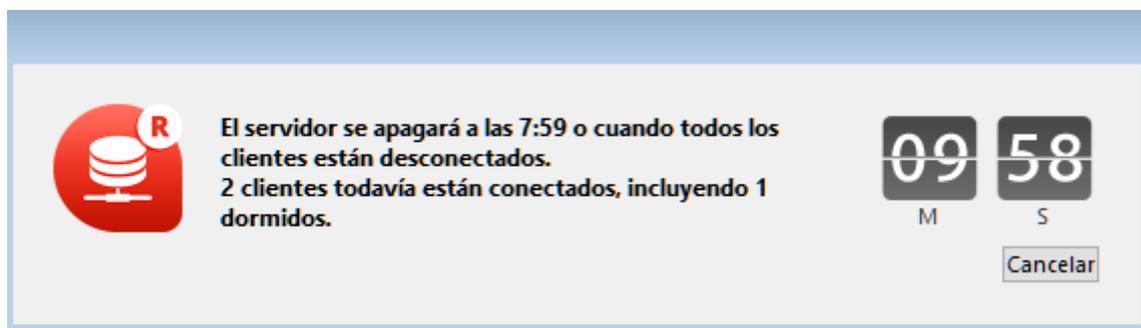
2. Introduzca el número de minutos en los que desea que el servidor se cierre o seleccione la opción "Espere a que todos los usuarios se desconecten".

Tan pronto como haga esto, ningún cliente nuevo se puede conectar al servidor.

Las siguientes opciones están disponibles:

- **Desconectar del servidor en XX minutos.**

Después del período de tiempo especificado, el servidor se cierra y todos los usuarios se desconectan, incluyendo todo cliente en modo dormir. La siguiente ventana aparece en el servidor:



Una ventana idéntica aparece en cada equipo cliente 4D. Esta ventana se repite o actualiza en cada máquina cliente cada 20 segundos o menos, con el fin de obligarlos a salir. Cuando se cumple el tiempo límite, el servidor se cierra, aunque haya máquinas cliente aún conectadas.

- **Esperar a que todos los clientes se desconecten**

El servidor se cerrará sólo después de que salgan todos los clientes, incluyendo los que están en modo dormir. Esta opción podría resultar inadecuada para las operaciones de mantenimiento que se ejecutan durante la hora del almuerzo, por ejemplo, ya que no es probable que haya clientes en modo dormir.

- **Esperar a que los clientes activos se desconecten (Ignorar clientes dormidos)**

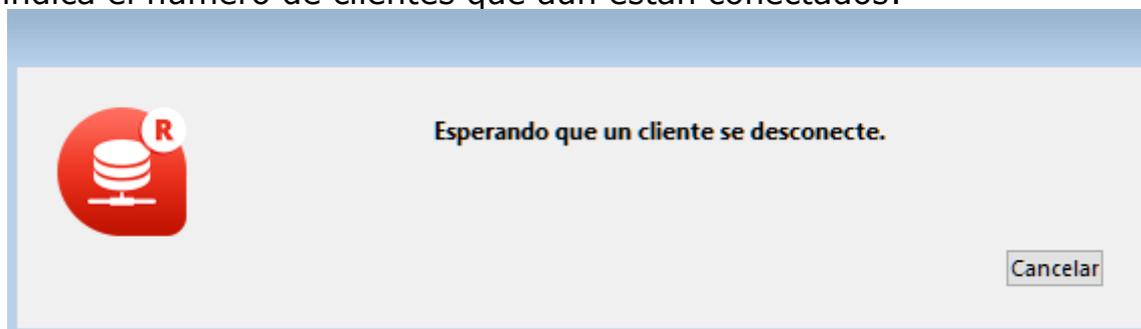
El servidor sólo se cerrará después de que todos los clientes activos se hayan desconectado (en otras palabras, todos los equipos cliente que no están en modo dormir). Con esta opción, los clientes en modo dormir no se consideran como conectado. Utilice esta opción si desea realizar operaciones de mantenimiento durante la hora del almuerzo, por ejemplo.

Cuando se utiliza esta opción, los clientes en modo de suspensión tendrán un error de conexión cuando se despierten.

Nota: esta opción requiere el uso de la capa de red *ServerNet*. Para mayor información, consulte **Nueva capa de red ServerNet (compatibilidad)**.

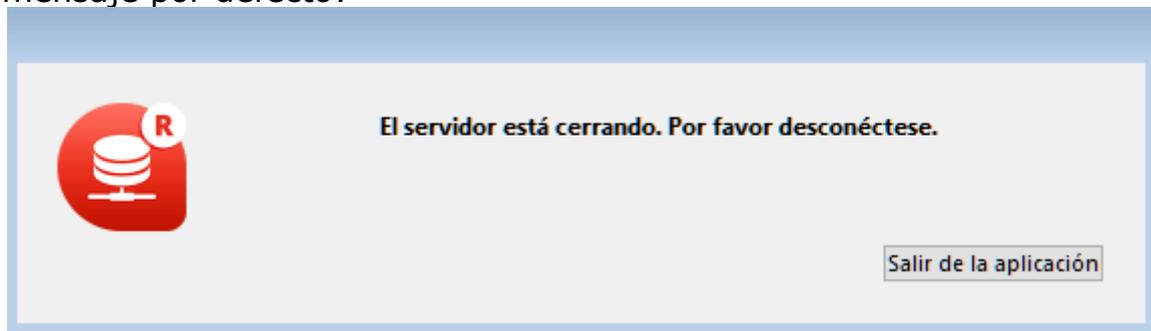
Nota: un *cliente dormido* se refiere a una aplicación 4D remota en una máquina que ha cambiado a modo dormir mientras que la conexión a la máquina servidor sigue activa. Para más información sobre este punto, consulte **Gestión de usuarios dormidos**.

Cuando se elige una de estas opciones, se abrirá la siguiente ventana, que indica el número de clientes que aún están conectados:

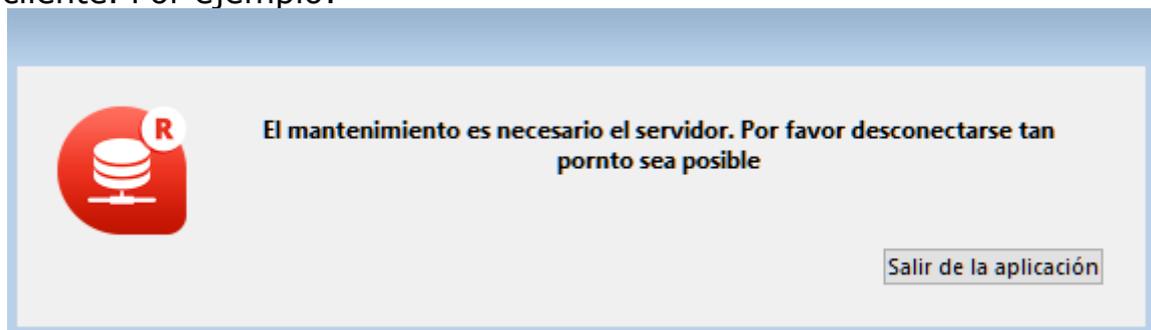


En cada equipo cliente 4D, aparece la siguiente ventana mostrando un

mensaje por defecto:



Si escribe un mensaje personalizado en la caja de diálogo de cierre de 4D Server, se muestra en lugar del mensaje por defecto en cada máquina cliente. Por ejemplo:



- **Desconectar los clientes y salir**

El servidor termina todos los procesos y todas las conexiones y se cierra después de unos segundos.

Notas:

- En todos los casos, si ningún cliente está conectado al servidor cuando la ventana de cierre se valida, 4D Server se cierra inmediatamente.
- Si hace clic en **Cancelar** en la ventana de apagado de 4D Server, el proceso de apagado del servidor se cancela.
- Puede cerrar la base (y desconectar los clientes) sin cerrar la aplicación 4D Server utilizando el comando **Cerrar la base...** Para más información, consulte la sección **Menús de 4D Server**.

Menús de 4D Server

La interfaz de la aplicación 4D Server se compone de los siguientes menús: **Archivo, Edición, Ventana, Ayuda**. Bajo Mac OS, algunos comandos se encuentran en el menú **4D Server** (menú aplicación).

Archivo



Nuevo

Este comando jerárquico tiene submenús que permiten crear una base de datos o un nuevo archivo de datos en el equipo servidor.

Los comandos de creación de bases se detallan en la sección **Crear o abrir una base 4D Server**.

Abrir.../Abrir reciente

Estos comandos permiten abrir una base con 4D Server. El comando **Abrir reciente** muestra un submenú que lista las bases abiertas recientemente por 4D Server. Para reiniciar este menú, seleccione el comando **Borrar lista de recientes**. Los comandos de apertura de bases se detallan en la sección **Crear o abrir una base 4D Server**.

Cerrar la base...

Este comando cierra la base actual sin cerrar la aplicación 4D Server. Cuando usted elije este comando, aparece la caja de diálogo de cierre del servidor, permitiéndole definir el modo de desconexión de los posibles clientes conectados (ver la sección [Salir de 4D Server](#)).

Cerrar la ventana

Este comando cierra la ventana de la aplicación 4D Server ubicada en el primer plano.

Cerrar todas las ventanas

Este comando cierra todas las ventanas de la aplicación 4D Server. Note que en este caso sólo la activación del comando **Cerrar la base...** en el menú **Archivo** le permite saber si la base aún está publicada.

Dar de alta la base actual como servicio/Dar de baja la base actual/Dar de baja todos los servicios de Server

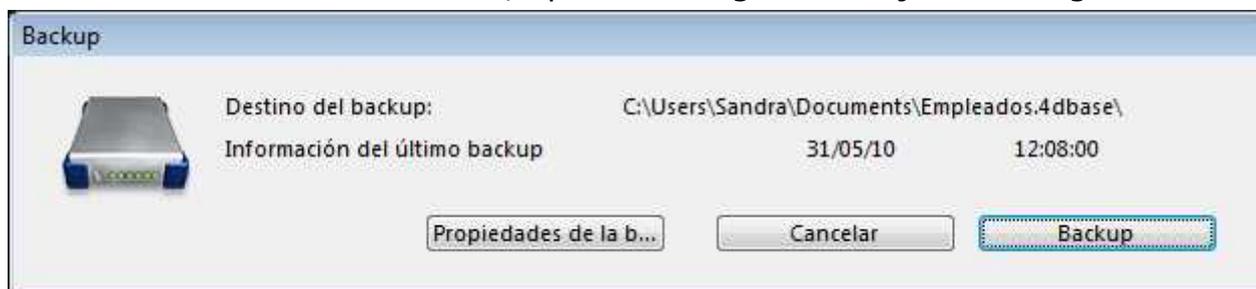
(Comandos disponibles en Windows) Estos comandos permiten administrar el registro de la base como Servicio. Esta función se detalla en la sección [Registrar una base como servicio](#).

Grabar caché de datos

Este comando permite "forzar" el registro en el disco de los datos ubicados en la caché. Por defecto, 4D Server escribe automáticamente la caché en el disco después de un tiempo límite definido en las preferencias de la base (página Base de datos/Gestión de datos).

Copia de seguridad

Este comando permite lanzar un backup de la base en cualquier momento. Cuando usted selecciona este comando, aparece la siguiente caja de diálogo:



- El botón **Backup** lanza inmediatamente un backup que tiene en cuenta los parámetros definidos en la caja de diálogo de Preferencias de la aplicación (archivos a los cuales realizar backp, ubicación de los archivos, número de conjuntos, etc.).
- El botón **Propiedades de la base** abre el tema "Backup" de las Propiedades de la base, permitiéndole visualizar, y si es necesario, modificar los parámetros del backup actual.

- El botón **Cancelar** interrumpe el proceso de backup.
Para mayor información sobre la configuración de backups, consulte el manual de Diseño.

Restaurar...

Este comando muestra una caja de diálogo de apertura de archivos que le permite seleccionar un archivo a restituir.

Salir

Este comando permite cerrar la aplicación 4D Server. Para mayor información, consulte la sección **Salir de 4D Server**.

Nota: bajo Mac OS X, el comando **Salir** está en el menú **4D Server** (menú aplicación).



Edición



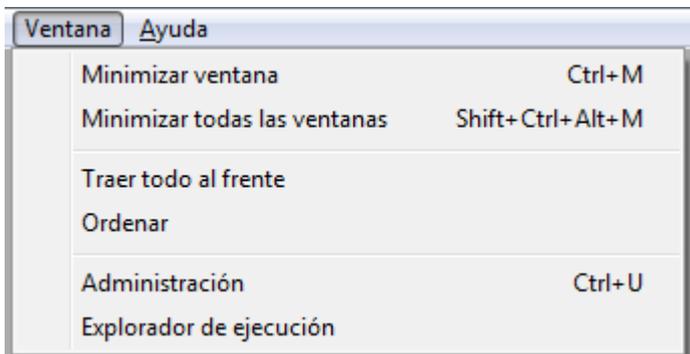
El menú **Edición** de 4D Server incluye los comandos estándar copiar/pegar, el comando **Mostrar el portapapeles**, etc.

Este menú también incluye (bajo Windows) el comando **Preferencias...**, que muestra la caja de diálogo de preferencias de la aplicación, permitiendo definir numerosos funcionamientos de la base. Para mayor información sobre esta caja de diálogo, consulte el manual de Diseño de la documentación de 4D. Las preferencias específicas de 4D Server se describen en las secciones **Opciones red y cliente-servidor** y **Configuración IP**.

Nota: bajo Mac OS, el comando **Preferencias...** se encuentra en el menú **4D Server** (menú aplicación).



Ventana



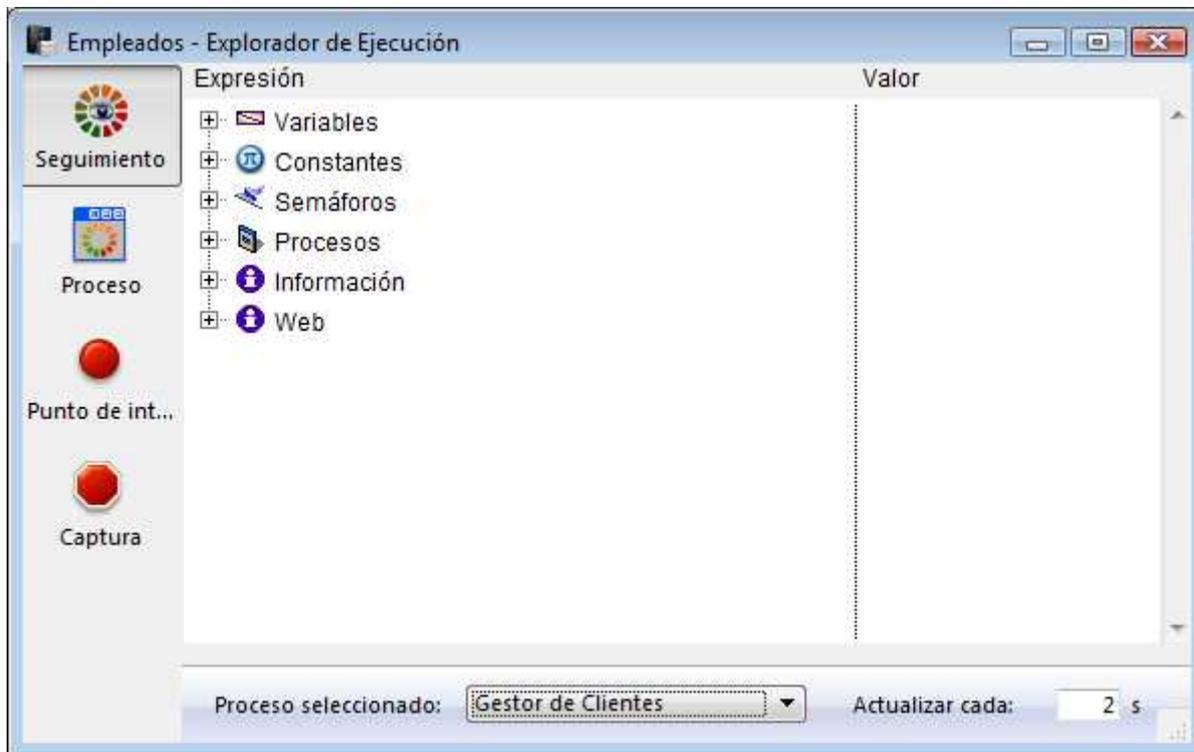
El menú **Ventana** incluye los comandos estándar para organizar las ventanas en el espacio de trabajo (estos comandos difieren dependiendo de la plataforma). También contiene los comandos de visualización de ventanas específicas de 4D Server:

Administración

Este comando muestra la ventana de administración de 4D Server, si se ha cerrado o minimizado. Esta ventana se detalla en el capítulo Ventana de administración de 4D Server (ver la sección [Página Monitor](#)).

Explorador de ejecución

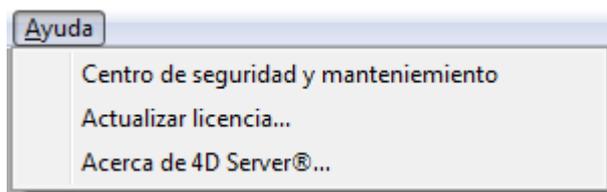
Este comando muestra la ventana del Explorador de ejecución de 4D Server.



El Explorador de ejecución permite visualizar el comportamiento de los diferentes elementos estructurales de la base y verificar que los recursos disponibles se utilicen correctamente. El Explorador de ejecución es particularmente útil en fase de desarrollo y análisis de una base de datos.

La ventana contiene cuatro páginas, accesibles vía los botones correspondientes: Seguimiento, Proceso, Punto de interrupción y Captura. El funcionamiento del Explorador de ejecución en 4D Server es idéntico al de 4D. Para mayor información, consulte el manual de Diseño.

Ayuda



Centro de seguridad y mantenimiento (CSM)

Este comando muestra el Centro de seguridad y mantenimiento (CSM), que agrupa todas las herramientas necesarias para el control, análisis, mantenimiento, backup y compactación de archivos de datos y de estructura.

Este comando está disponible incluso cuando ninguna base ha sido abierta por 4D Server: en este caso, puede utilizarse para abrir una base de datos en "modo de mantenimiento" (que muestra una caja de diálogo estándar de apertura de archivos para que pueda designar la base de datos a abrir). El modo de mantenimiento se aplica más concretamente para operaciones tales como la compactación o la apertura de bases dañadas.

Para obtener más información sobre el CSM, consulte el Manual de Diseño.

Actualizar licencia...

Este comando de menú muestra la ventana utilizada para activar licencias adicionales en su entorno 4D.
Para mayor información sobre esta caja de diálogo, consulte la Guía de instalación.

Acerca de 4D Server...

Este comando muestra la ventana Acerca de 4D Server..., la cual ofrece la siguiente información:

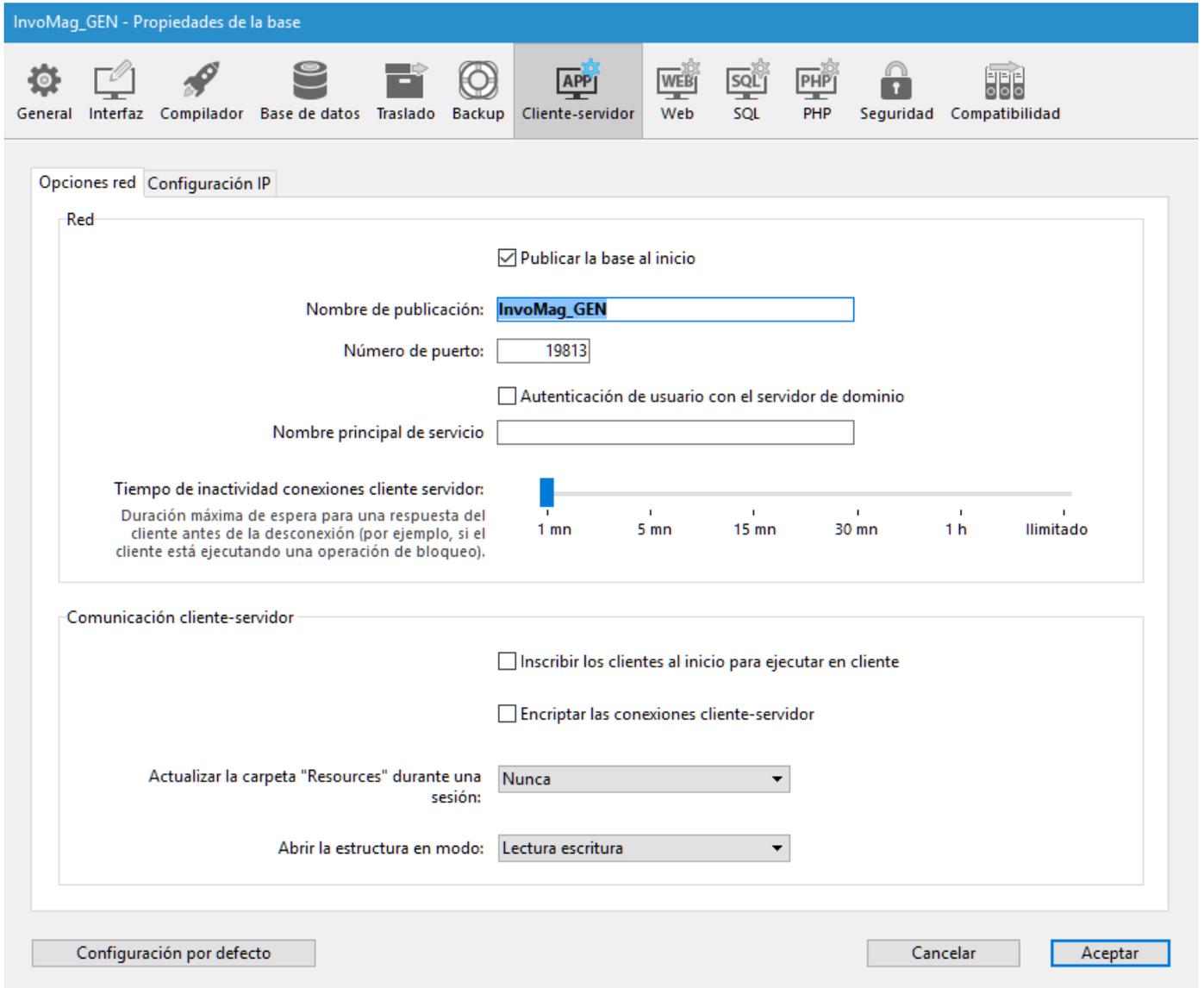
- Versión y licencia de 4D Server
- Derechos de autor e información legal
- 4D en el mundo



Nota: bajo Mac OS, el comando **Acerca de 4D Server** se encuentra en el menú **4D Server** (menú aplicación).

Opciones red y cliente-servidor

Puede definir diferentes parámetros relativos a la red y a la comunicación cliente-servidor en diferentes pestañas de la página **Cliente-Servidor** de las Propiedades de la base (accesibles desde 4D en modo remoto y 4D Server):



The screenshot shows the 'InvoMag_GEN - Propiedades de la base' dialog box with the 'Cliente-servidor' tab selected. The 'Opciones red' section is active, showing the following settings:

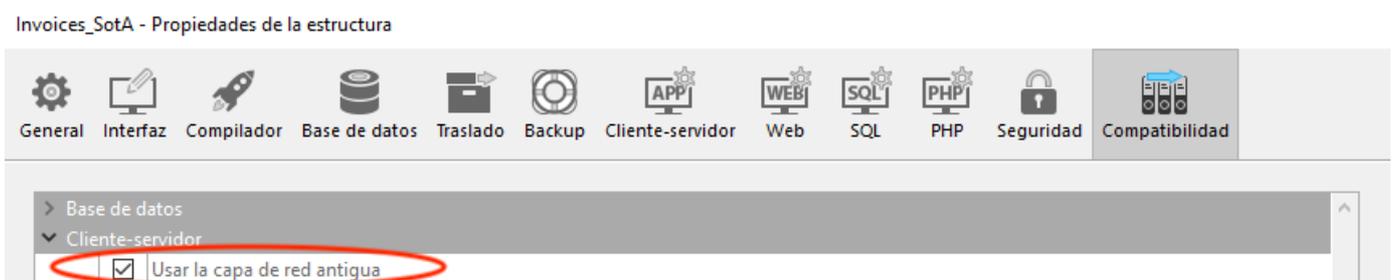
- Publicar la base al inicio
- Nombre de publicación:
- Número de puerto:
- Autenticación de usuario con el servidor de dominio
- Nombre principal de servicio:
- Tiempo de inactividad conexiones cliente servidor: (1 mn, 5 mn, 15 mn, 30 mn, 1 h, ilimitado)

The 'Comunicación cliente-servidor' section shows:

- Inscribir los clientes al inicio para ejecutar en cliente
- Encriptar las conexiones cliente-servidor
- Actualizar la carpeta "Resources" durante una sesión:
- Abrir la estructura en modo:

Buttons at the bottom: Configuración por defecto, Cancelar, and Aceptar.

Además, a partir de 4D Server v14 R5, una opción de compatibilidad le permite activar o desactivar en cualquier momento la capa de red antigua:



The screenshot shows the 'Invoices_SotA - Propiedades de la estructura' dialog box with the 'Compatibilidad' tab selected. The 'Usar la capa de red antigua' checkbox is checked and circled in red.

- Usar la capa de red antigua

Estos parámetros se detallan en esta sección.

Publicar la base al inicio

Esta opción permite indicar si la base 4D Server debe aparecer o no en la lista de bases publicadas en la caja de diálogo de conexión.

- Cuando la opción está seleccionada (opción por defecto), la base se vuelve pública y aparece en la lista de bases publicadas (página **Disponible**).
- Cuando la opción no está seleccionada, la base no se vuelve pública y no aparece en la lista de bases disponibles. Para conectarse, los usuarios deben introducir manualmente la dirección de la base en la página **Personalizado** de la caja de diálogo de conexión.

Nota: si modifica este parámetro, debe reiniciar la base servidor para que sea tenida en cuenta.

Nombre de publicación

Esta opción le permite cambiar el nombre de publicación de una base publicada por 4D Server, es decir, el nombre que se muestra en la página de publicación dinámica Disponible de la caja de diálogo de conexión (ver la sección **Conexión a una base 4D Server**). Por defecto, 4D Server utiliza el nombre del archivo de estructura de la base. Puede escribir cualquier nombre personalizado que desee.

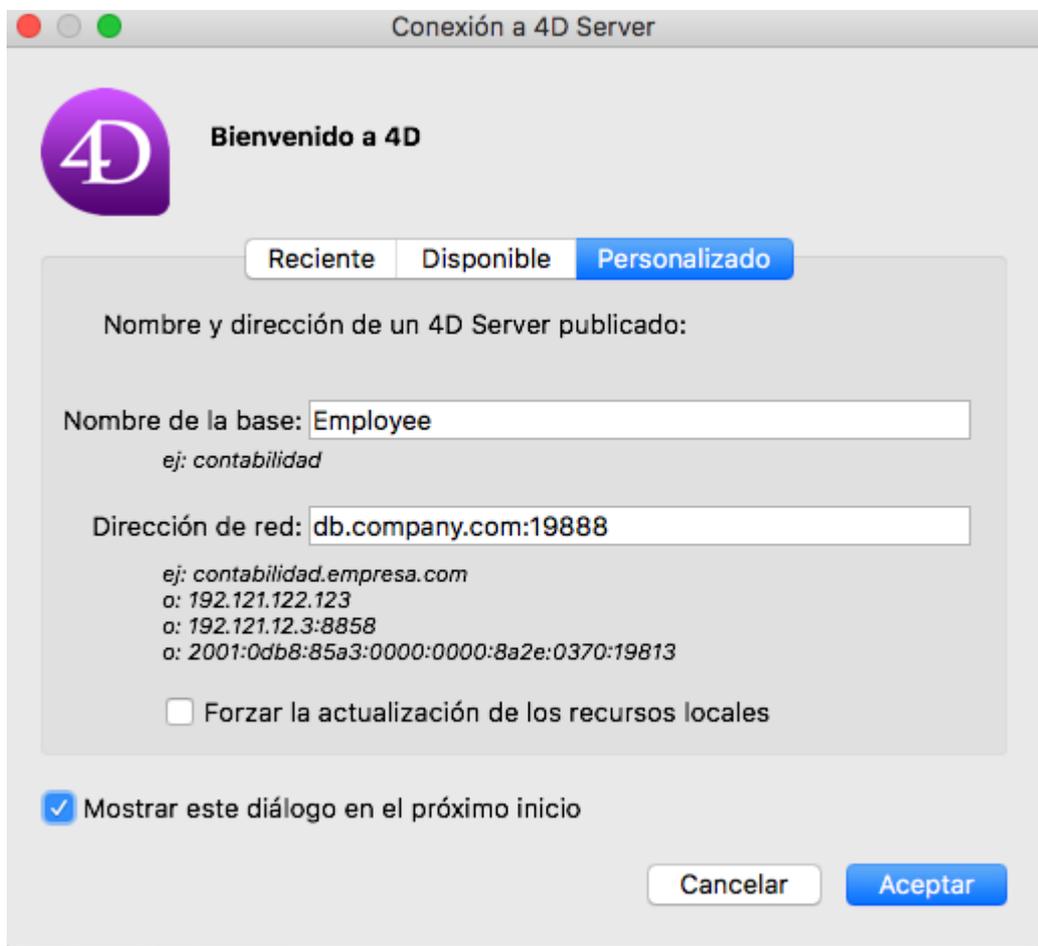
Nota: este parámetro no se tiene en cuenta en las aplicaciones cliente-servidor personalizadas. En teoría, la aplicación cliente se conecta directamente a la aplicación servidor, sin pasar por la caja de diálogo de conexión. Sin embargo, en caso de error, esta caja de diálogo aparecerá, en este caso, el nombre de publicación de la aplicación servidor es el nombre de la base compilada.

Número de puerto

Esta opción le permite cambiar el número de puerto TCP en el que 4D Server publica la base de datos. Esta información se almacena en la estructura de la base y en cada máquina cliente. Por defecto, el número de puerto TCP utilizado por 4D Server y 4D en modo remoto es el 19813.

Personalizar este valor es necesario cuando se desea utilizar varias aplicaciones 4D en el mismo equipo con el protocolo TCP, en este caso, debe especificar un número de puerto diferente para cada aplicación.

Al modificar este valor desde 4D Server o 4D, se transmite automáticamente a todas las máquinas 4D conectadas a la base. Para actualizar las otras máquinas clientes no conectadas, sólo tiene que introducir el nuevo número de puerto (precedido por dos puntos) después de la dirección IP de la máquina servidor en la página Personalizado de la caja de diálogo de conexión en el momento de la próxima conexión. Por ejemplo, si el nuevo número de puerto es 19888:



Nota: cuando 4D Server utiliza IPv4, sólo las bases publicadas en el puerto 19813 son visibles en la página de publicación dinámica "Disponible".

4D Server y los números de puerto

4D Server utiliza varios puertos TCP para las comunicaciones entre los servidores internos y los clientes:

- **Servidor SQL:** 19812 por defecto (modificable vía la página "SQL" de las Preferencias).
- **Servidor de la aplicación:** 19813 por defecto (modificable vía la página "Cliente-Servidor/Publicación" de las Preferencias, ver arriba).
- **Servidor DB4D** (servidor de base de datos): 19814 por defecto. Este número de puerto no se puede modificar directamente, sino que siempre es el número del puerto del servidor de la aplicación + 1.

Cuando un cliente 4D se conecta a 4D Server, se utiliza el puerto TCP del servidor de la aplicación (19813 o el puerto indicado después de los dos puntos ':' en la dirección IP que se muestra en la caja de diálogo de conexión). La conexión a los otros servidores a través de sus respectivos puertos es entonces automática, ya no es necesario especificarla.

Tenga en cuenta que en el caso de acceso a través de un router o un firewall, los tres puertos TCP deben estar abiertos en forma explícita.

Autenticación de usuario con servidor de dominio

Esta opción le permite implementar la función de autenticación única SSO (*Single Sign On*) en su base 4D Server en Windows. Cuando se selecciona esta opción, 4D se conecta de forma transparente al directorio Active del servidor de dominio de Windows y obtiene los *tokens* de autenticación disponibles.

Esta opción se describe en la sección **Autenticación única (Single Sign On - SSO) en Windows**.

Nombre principal del servicio

Cuando Single Sign On (SSO) está activo (ver arriba), debe llenar este campo si desea utilizar Kerberos como su protocolo de autenticación.

Esta opción se describe en la sección **Autenticación única (Single Sign On - SSO) en Windows**.

Tiempo de inactividad cliente-servidor

Este termómetro permite definir el periodo de inactividad después del cual la conexión se cierra entre 4D Server y los equipos clientes conectados a él.

La opción ilimitado elimina ese periodo. Cuando esta opción está seleccionada, el control de inactividad del cliente se desactiva.

Cuando se selecciona un periodo, el servidor cerrará la conexión de un cliente si no recibe ninguna petición del cliente durante el tiempo límite especificado.

Opciones

Inscribir los clientes al inicio para ejecutar en cliente

Cuando esta opción está seleccionada, todos los equipos 4D remotos que se conectan a la base pueden ejecutar los métodos remotamente. Este mecanismo se detalla en la sección **Procedimientos almacenados en los equipos clientes**.

Encriptar las conexiones cliente-servidor

Esta opción permite activar el modo seguro para la comunicación entre el equipo servidor y los equipos 4D remotos. Esta opción se detalla en la sección **Encriptar las conexiones cliente-servidor**.

Actualizar la carpeta "Resources" durante una sesión

Esta configuración permite definir globalmente el modo de actualización de la instancia local de la carpeta en las máquinas 4D conectadas cuando la carpeta de la base ha sido modificada durante la sesión (la carpeta se sincroniza automáticamente en el equipo remoto cada vez que una sesión se abre). Tres opciones están disponibles:

- **Nunca:** la carpeta local no se actualiza durante la sesión. La notificación enviada por el servidor se pasa por alto. La carpeta local se puede actualizar de forma manual utilizando el comando **Actualizar recursos locales** del menú de acción (ver **Utilizar el explorador de recursos**).
- **Siempre:** la sincronización de la carpeta local se lleva a cabo automáticamente durante la sesión cada vez que se envía la notificación por el servidor.
- **Preguntar:** cuando se envía la notificación por el servidor, se muestra una caja de diálogo en las máquinas cliente, indicando la modificación. El usuario puede aceptar o rechazar la sincronización de la carpeta local.

La carpeta centraliza los archivos personalizados necesarios para la interfaz de la base (archivos de traducción, imágenes, etc.) Los mecanismos automáticos o manuales permiten notificar a cada cliente cuando el contenido de esta carpeta se ha modificado. Para mayor información, consulte la sección **Gestión de la carpeta Resources**.

Abrir la estructura en modo

Esta opción permite definir el modo de apertura de la estructura de la base para los equipos clientes. Por defecto, está definido el modo **Lectura/Escritura** pero

también puede configurar la apertura en modo **Lectura únicamente** para evitar modificaciones en la estructura.

Configuración IP

Tabla de configuración Autorizar-Rechazar

Esta tabla permite establecer las reglas de control de acceso a la base en función de la dirección IP de los equipos 4D remotos. Esta opción permite reforzar la seguridad, por ejemplo, para aplicaciones estratégicas.

Nota: esta tabla de configuración no controla las conexiones web.

El funcionamiento de la tabla de configuración es el siguiente:

- La columna "Autorizar-Rechazar" permite seleccionar el tipo de regla a aplicar (Autorizar o Rechazar) utilizando un menú pop-up. Para agregar una regla, haga clic en el botón Añadir. Una nueva línea aparece en la tabla. El botón Eliminar le permite eliminar la línea actual.
- La columna "Dirección IP" permite designar la o las dirección(es) IP, incluidas en la regla. Para especificar una dirección, haga clic en la columna y escriba la dirección de la forma siguiente: 123.45.67.89 (formato IPv4) o 2001:0DB8:0000:85A3:0000:0000:AC1F:8001 (formato IPv6). Puede utilizar un carácter * (asterisco) para especificar las direcciones de tipo "comienza por". Por ejemplo, 192.168 .* indica todas las direcciones que comienzan por 192.168.
- La aplicación de las reglas se efectúa en el orden de visualización de la tabla. Si dos reglas son contradictorias, se da prioridad a la regla ubicada más arriba en la tabla.
Puede reordenar las líneas modificando la ordenación actual (haga clic en el encabezado de la columna para alternar el sentido de ordenación).
También puede mover líneas mediante arrastrar y soltar.
- Por razones de seguridad, sólo las direcciones correspondientes a una regla de autorización explícita podrán conectarse.
En otras palabras, si la tabla sólo contiene una o más reglas Rechazar, todas las direcciones serán rechazadas porque ninguna cumplirá al menos una regla. Si desea rechazar sólo determinadas direcciones (y permitir otras), añada una regla Autorizar * al final de la tabla. Por ejemplo:
 - Rechazar 192.168.* (rechazar todas las direcciones que comiencen por 192.168)
 - Autorizar * (y autorizar todas las demás direcciones)Por defecto, no se aplican restricciones de conexión por 4D Server: la primera línea de la tabla contiene la etiqueta Autorizar y el carácter * (todas las direcciones).

Nueva capa de red ServerNet (compatibilidad)

A partir de 4D v15 (v14 R5), las aplicaciones 4D contienen una nueva capa de red, llamada *ServerNet*, para manejar las comunicaciones entre 4D Server y equipos 4D remotos (clientes). *ServerNet* se basa en una API moderna y robusta. Es fácil de mantener y facilitará la implementación de las últimas tecnologías de red al tiempo que ofrece un alto nivel de rendimiento.

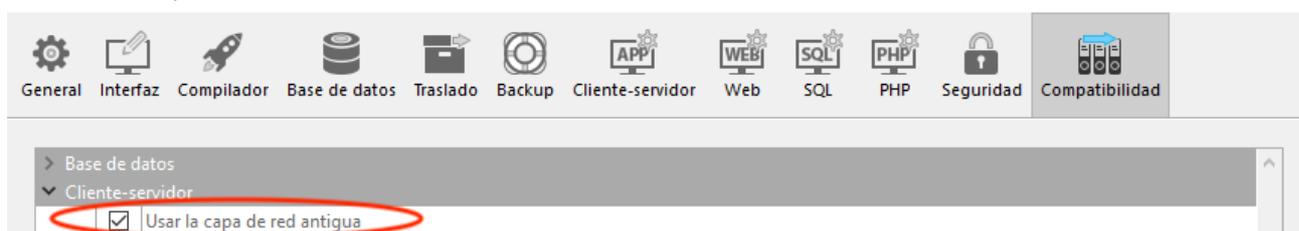
Utilizar *ServerNet* es transparente desde el punto de vista del usuario. Tenga en cuenta, sin embargo, que cuando se utiliza *ServerNet*, los nombres de las bases publicadas en modo seguro ya no están precedidos por un carácter "^", como era el caso de la capa de red antigua (ver **Encriptar las conexiones cliente-servidor**). La capa de red antigua se conserva con el fin de garantizar la compatibilidad de las bases existentes. *ServerNet* se utiliza automáticamente en bases de datos nuevas. Hay opciones que le permiten activar/desactivar el *ServerNet*. Para asegurarse de que sus aplicaciones se beneficiarán de las futuras evoluciones de la red, le recomendamos que habilite *ServerNet* gradualmente en todas sus bases.

Activar o desactivar la capa de red antigua

Puede activar o desactivar en cualquier momento la capa de red antigua en su 4D Server. Puede utilizar:

- la opción **Usar la capa de red antigua** en la caja de diálogo de las Propiedades de la base (ver **Página Compatibilidad**):

Invoices_SotA - Propiedades de la estructura



- la constante Use legacy network layer con el comando **SET DATABASE PARAMETER**.

Nota: como se indica en su título, esta opción se ignora en 4D Server versión 64 bits para OS X; solo *ServerNet* se puede utilizar en esta plataforma.

Por defecto, la capa de red *ServerNet* se:

- activa automáticamente en las nuevas bases, creadas con 4D v14 R5 o versiones posteriores,
- desactiva automáticamente en las bases convertidas.

Migración de los clientes 4D fusionados

Cuando se activa la capa *ServerNet* en su aplicación servidor existente, sólo las aplicaciones cliente 4D sólo compatibles podrán conectarse:

- Las aplicaciones clientes en versión 4D v15 (o a partir de 4D v14 R4) pueden conectarse sin modificación.
- Las aplicaciones clientes en versiones anteriores (v14.x y todos los *releases* v14 'R' anteriores) primero deben actualizarse para poder conectarse al servidor.

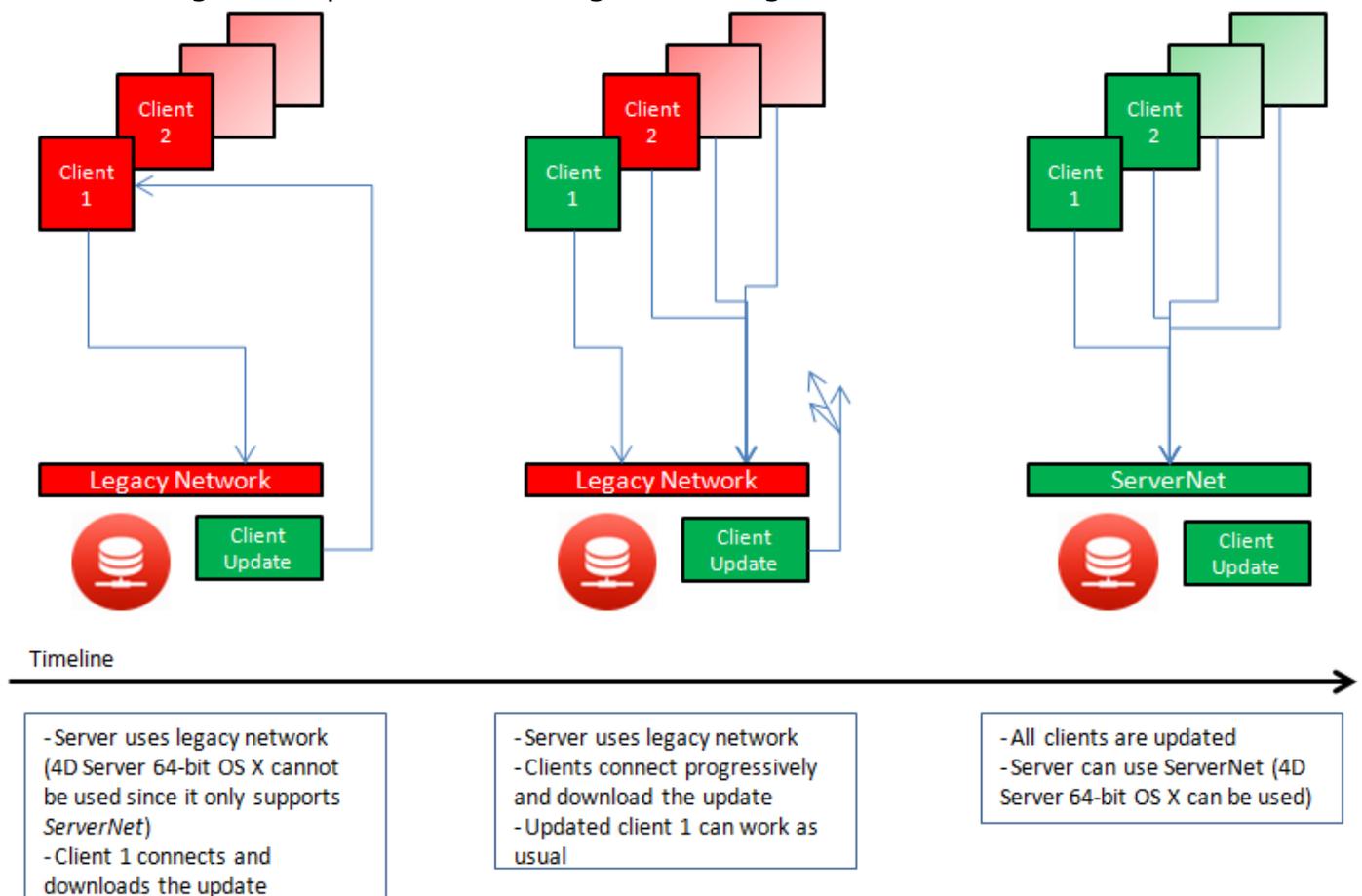
Si su aplicación funciona con clientes 4D Volume Desktop fusionadas en versiones anteriores a v14 R4, y desea utilizar el mecanismo automático de 4D Server para distribuir sus aplicaciones cliente actualizadas través de la red, es necesario establecer una estrategia de migración. Esta estrategia debe basarse en las siguientes principios:

- Los clientes no compatibles sólo pueden conectarse a un 4D Server utilizando la capa de red antigua.
- Los clientes actualizados pueden adaptar su protocolo de forma dinámica para que puedan conectarse a 4D Server v14 R5 y superiores, independientemente de la capa de red que el servidor esté utilizando.

Su estrategia de migración debe seguir estos pasos:

1. Cree una actualización de la aplicación cliente con 4D v15 o superior.
2. Lance 4D Server v15 o superior con el parámetro "Utilizar la capa de red antigua" activado.
Esta configuración permite que todos los clientes se conecten.
3. Espere un período de tiempo determinado hasta que cada cliente se conecte y descargue la nueva versión.
Esto puede durar un día, una semana o incluso más tiempo. Durante este período de transición, los clientes antiguos y actualizados pueden conectarse al servidor de red antiguo..
4. Una vez que todos los clientes se han actualizado, puede desactivar la capa de red antigua y activar definitivamente *ServerNet* en 4D Server.

Esta estrategia se representa en el siguiente diagrama:



Registrar las peticiones de los clientes

Durante el proceso de migración, se recomienda que habilite el archivo "Diagnostic log recording". Cuando se activa este archivo, 4D Server registra cada solicitud de actualización de clientes en este archivo, lo que le permite controlar el proceso. Este registro no se activa por defecto: debe llamar al comando **SET DATABASE PARAMETER** con la constante Diagnostic log recording definida en 1.

Para cada solicitud de actualización, la siguiente información se registra:

- IP del cliente
- versión del cliente

El seguimiento del archivo de historial también es útil después de que haya cambiado el servidor a la capa de red *ServerNet*, para asegurarse de que todos los

clientes se hayan actualizado correctamente. Si un cliente no compatible intenta conectarse, el servidor registra la siguiente entrada:

- IP del cliente
- versión del cliente
- evento "Fail to connect"

En este caso, puede decidir, por ejemplo, actualizar el cliente de forma manual.

Configuración IP

Soporte de IPv6

A partir de v16 R4, el servidor de aplicaciones 4D soporta la notación de direcciones IPv6. El soporte de IPv6 es transparente para los usuarios y para los desarrolladores 4D: 4D Server acepta conexiones IPv6 o IPv4 indiscriminadamente.

Sin embargo, el soporte de las direcciones IPv6 depende de las versiones 4D Server y 4D remotas (32 bits o 64 bits). La siguiente tabla muestra las configuraciones soportadas:

			Client					
			32-bit			64-bit		
			IPv4	IPv6	Both	IPv4	IPv6	Both
Server	32-bit	IPv4	IPv4		IPv4	IPv4		IPv4
		IPv6					IPv6	IPv6
		Both	IPv4		IPv4	IPv4	IPv6	IPv6
	64-bit	IPv4	IPv4		IPv4	IPv4		IPv4
		IPv6					IPv6	IPv6
		Both	IPv4		IPv4	IPv4	IPv6	IPv6

IPv4	Supported (IPv4)
IPv6	Supported (IPv6)
	Not supported

Nota de compatibilidad: el soporte de IPv6 sólo está disponible cuando la capa de red ServerNet está activada en 4D Server (ver [Nueva capa de red ServerNet \(compatibilidad\)](#)).

Para obtener información detallada sobre IPv6, consulte la siguiente especificación: [RFC 2460](#).

Encriptar las conexiones cliente-servidor

Puede configurar sus conexiones cliente/servidor de manera que 4D Server y los equipos 4D remotos se comuniquen en modo seguro. La comunicación cliente/servidor está basada en el protocolo TLS (Transport Layer Security), antiguamente SSL (Secured Socket Layer).

Nota: para una descripción general de las funcionalidades de seguridad de 4D, consulte la [guía de seguridad de 4D](#).

El protocolo TLS y las conexiones cliente/servidor

El protocolo TLS se ha diseñado para dar seguridad a las comunicaciones entre dos aplicaciones, principalmente entre un servidor web y un navegador. El protocolo SSL está diseñado para autenticar al remitente y el receptor y para garantizar la confidencialidad e integridad de la información intercambiada. Para una descripción detallada del protocolo de seguridad, consulte la sección en el manual Lenguaje de 4D.

Entre 4D Server y un 4D remoto, los mecanismos de autenticación son "transparentes" y son gestionados por 4D Server y no requieren de ninguna intervención del usuario.

Nota: encriptar conexiones cliente/servidor ralentiza las conexiones.

Configuración

A nivel de red, el protocolo TLS se inserta entre la capa TCP/IP (nivel bajo) y el protocolo de alto nivel.

Para poder utilizar TLS en una configuración cliente/servidor, los archivos de certificados siguientes deben estar presentes:

- *key.pem* (documento que contiene la llave encriptada privada) y *cert.pem* (documento que contiene el certificado). Estos archivos deben estar en la subcarpeta *Resources* de las aplicaciones 4D Server y 4D. Tienen que estar presentes en la máquina servidor y en cada máquina remota. Por defecto los archivos *key.pem* y *cert.pem* vienen con 4D. Para un mayor nivel de seguridad, se recomienda encarecidamente que cambie estos archivos por sus propios certificados. Para más información sobre cómo crear certificados personalizados, consulte la sección **¿Cómo obtener un certificado?** de la documentación del servidor web de 4D (el procedimiento es idéntico).

Debe activar las conexiones también en modo seguro. Para ello, abra la página "Cliente-Servidor/Opciones de red" de la caja de diálogo de las Propiedades de la base y seleccione la opción **Encriptar las conexiones**

cliente-servidor en el área de "Encriptación" (ver la sección **Opciones red y cliente-servidor**).

Por defecto, la casilla no está seleccionada.

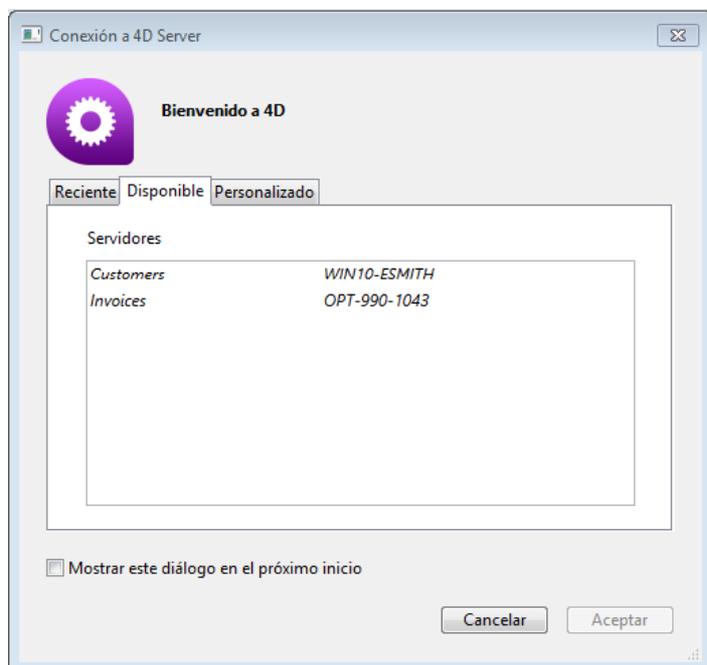
A continuación, debe salir y relanzar 4D Server para que esta configuración se tenga en cuenta.

Todos los equipos remotos 4D se conectarán en modo seguro.

Conexión en modo seguro

- **Uso de ServerNet**

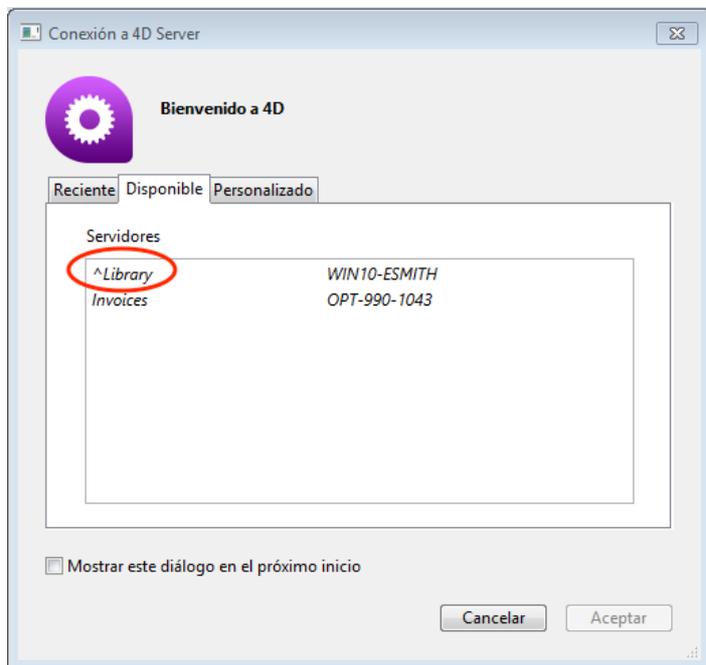
Cuando se utiliza la nueva capa de red ServerNet, la activación del protocolo de seguridad en el servidor es transparente (el modo de publicación no aparece en la caja de diálogo de conexión). La conexión y el paso a modo seguro se efectúa después de una negociación interna entre la aplicación servidor y la aplicación remota.



Para más información acerca de *ServerNet*, consulte la sección **Opciones red y cliente-servidor**.

- **Uso de la antigua capa de red**

Cuando se utiliza la antigua capa de red, el símbolo "acento circunflejo" (^) aparece delante del nombre de las bases publicadas en modo TLS en la página de conexión TCP/IP a 4D Server:



Nota: cuando el nombre de una base no se publica dinámicamente en la página de conexión TCP/IP, el usuario puede introducirlo manualmente en la página Personalizado (ver las secciones **Conexión a una base 4D Server** y **Configuración IP**). En este caso, debe iniciar el nombre con un ^ (acento circunflejo) si la base se publica en modo seguro; de lo contrario la conexión se rechazará.

Autenticación única (Single Sign On - SSO) en Windows

4D Server le permite implementar funcionalidades SSO (*Single Sign On*) en soluciones cliente-servidor en Windows.

La implementación de SSO en sus soluciones 4D permitirá a los usuarios acceder a la aplicación 4D en Windows sin tener que volver a introducir su contraseña cuando ya están registrados en el dominio Windows de su empresa (utilizando Active Directory). Detrás, la aplicación 4D Server puede tener acceso a la autenticación de inicio de sesión de Windows, que se puede utilizar para iniciar la sesión del usuario 4D

en la base utilizando el método de inicio de sesión estándar.

Nota: para una descripción general de las demás funcionalidades de copia de autenticación y seguridad de 4D, consulte la [guía de seguridad de 4D](#).

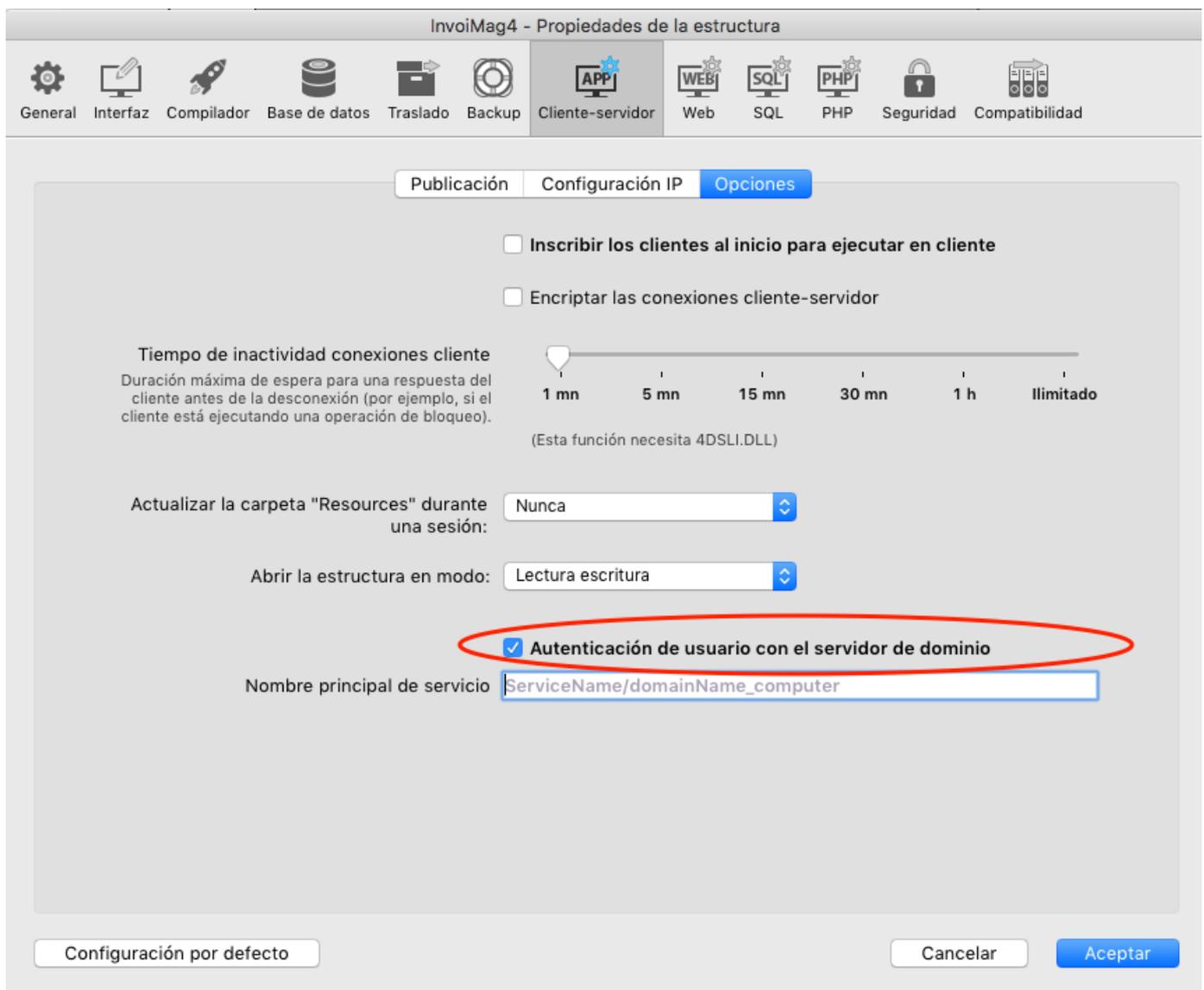
Requerimientos

La funcionalidad SSO está disponible:

- con aplicaciones 4D Server en Windows (las aplicaciones mono usuario 4D no soportan SSO)
- con la capa de red **ServerNet** activada. Para más información, consulte el párrafo **Nueva capa de red ServerNet (compatibilidad)**.

Activación de la funcionalidad SSO

Por defecto, la funcionalidad SSO no está habilitada en 4D Server. Para beneficiarse de esta funcionalidad, es necesario establecer la opción **Autenticación de usuario con el servidor de dominio** en la página Cliente-servidor/Opciones de red del diálogo Propiedades de la base de 4D Server:



Cuando selecciona esta opción, 4D se conecta de forma transparente al directorio Active del servidor dominio de Windows y obtiene los tokens de autenticación disponibles.

Esta opción ofrece autenticación estándar a través del protocolo NTLM. 4D soporta los protocolos NTLM y Kerberos. El protocolo utilizado es seleccionado automáticamente por 4D en función de la configuración actual (ver **Configuraciones soportadas**). Si desea utilizar el protocolo Kerberos, es necesario llenar el campo SPN adicional (ver más adelante).

Activación de Kerberos

Si desea utilizar Kerberos como protocolo de autenticación, también es necesario llenar el campo **Nombre principal de servicio** en la página Cliente-Servidor/Opciones de red del cuadro de diálogo Propiedades de la base:

Autenticación de usuario con el servidor de dominio

Nombre principal de servicio

Esta opción declara el SPN tal como se establece en la configuración Active Directory. Un nombre principal de servicio es un identificador único de una instancia de servicio. Los SPNs son utilizados por la autenticación Kerberos para asociar una instancia de servicio con una cuenta de servicio de inicio de sesión. Esto permite que una aplicación cliente solicite que el servicio autentique una cuenta incluso si el cliente no tiene el nombre de la cuenta. Para más información, por favor consulte [la página SPN en el sitio web de msdn](#).

El identificador SPN debe respetar este patrón:

- "ServiceName/FQDN_user" si el SPN es un atributo del ordenador
- "ServiceName/FQDN_computer" si el SPN es un atributo de usuario

dónde:

- *ServiceName* es el nombre del servicio al que el cliente desea autenticarse.
- El *Fully Qualified Domain Name* (FQDN) es un nombre de dominio que especifica su ubicación exacta en la jerarquía de árbol del Active Directory para ambos equipos y usuarios.

En las bases de datos 4D, el SPN se puede configurar:

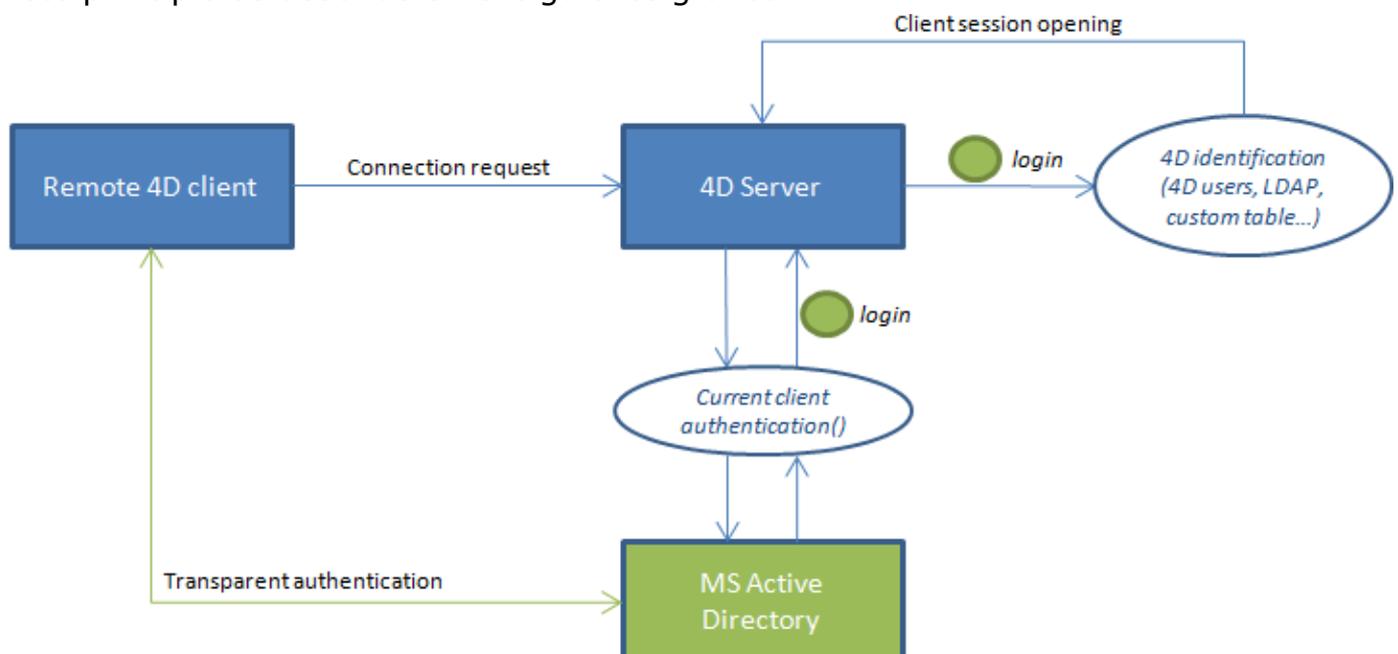
- en la configuración de la estructura de la base, para un uso con 4D Server.
- o en la configuración del usuario (*settings*. archivo *4DSettings* almacenado en la carpeta Preferencias de la base de datos) para las necesidades de despliegue.

Implementar SSO

Cuando las funcionalidades SSO están activas (ver arriba), puede confiar en la autenticación de usuario basada en las credenciales de sesión de Windows para abrir una sesión de usuario en 4D Server.

Tenga en cuenta que la funcionalidad SSO sólo ofrece un inicio de sesión autenticado, debe pasar esta información de acceso a su método de inicio de sesión 4D estándar. Cuando una aplicación 4D remota intenta conectarse al servidor, debe llamar al comando 4D **Current client authentication**, que devolverá el nombre del usuario, tal como se define en el Active Directory. Luego, puede pasar esta información de acceso a su propio sistema de identificación (utilizando el usuario y grupos integrados, los comandos LDAP, o cualquier mecanismo personalizado) para abrir la sesión apropiada para el usuario remoto en su aplicación 4D.

Este principio se describe en el siguiente gráfico:



El comando **Current client authentication** debe ejecutarse en el método base **On Server Open Connection**, que se llama cada vez que un 4D remoto abre una

nueva conexión en la base 4D Server. Si la autenticación falla, puede devolver un valor no nulo en \$0 para rechazar la conexión.

Uso del comando **Current client authentication**

El comando **Current client authentication**, utiliza la siguiente sintaxis:

```
login=Current client authentication(dominio;protocolo)
```

donde:

- *login* es el identificador utilizado por el cliente para iniciar sesión en el Active Directory (texto). Es necesario utilizar este valor para identificar al usuario dentro de su base.
Si el usuario no está autenticado registrado correctamente, una cadena vacía se devuelve y no se devuelve ningún error.
- *dominio* y *protocolo* son parámetros de texto opcionales. Son llenados por el comando y le permiten aceptar o rechazar las conexiones en función de estos valores:
 - *dominio* es el nombre del dominio de Active Directory
 - *protocolo* es el nombre del protocolo utilizado por Windows para autenticar al usuario. Puede ser "Kerberos" o "NTLM", dependiendo de los recursos disponibles.

Para más información sobre este comando, consulte la descripción del comando **Current client authentication**

Configuraciones soportadas

4D Server soporta varias configuraciones SSO, dependiendo de la arquitectura y la configuración actual. El protocolo utilizado para la autenticación (NTLM o Kerberos), así como información devuelta por el comando **CommonComment** dependen de la configuración del sistema. El protocolo utilizado para la autenticación se devuelve en el parámetro *protocolo* del comando **CommonComment**.

La siguiente tabla lista los **requerimientos para la autenticación NTLM o Kerberos**:

	NTLM	Kerberos
4D Server y 4D remoto en máquinas diferentes	sí	sí
El usuario 4D Server esté en el dominio	sí	sí
El 4D remoto esté en el mismo AD que el usuario de 4D Server	sí o no(*)	sí
El SPN esté indicado en 4D Server	no	sí(**)
Información devuelta por CommonComment si las condiciones son respetadas	cliente=login esperado, dominio=dominio esperado, protocolo="NTLM"	cliente=login esperado, dominio=esperado, protocolo="Kerberos"

(*)La siguiente configuración específica es soportada: el usuario remoto 4D es una cuenta local en una máquina que pertenece al mismo AC como 4D Server. En este caso, el parámetro *dominio* se llena con el nombre de la máquina 4D Server. Tenga

en cuenta que el soporte depende de los parámetros usuario: si no está disponible, **Current client authentication** devuelve cadenas vacías.

(**) Si se respetan todos los requisitos de Kerberos pero el comando **CommonComment** devuelve "NTLM" en *protocolo*, esto significa que usted se encuentra en una de las siguientes situaciones:

- La sintaxis SPN no es válida; en otras palabras, no respeta todas las [restricciones impuestas por Microsoft](#).
- O bien, el SPN tiene duplicados en el AD. Este problema sólo puede ser solucionado por el administrador AD en el lado AD.

Nota: una sintaxis válida no significa que la propia declaración SPN es correcta; en particular, si el SPN no existe en el AD, **CommonComment** devuelve cadenas vacías.

Gestión de la carpeta Resources

La carpeta **Resources** de la base se utiliza para compartir los datos personalizados (imágenes, archivos, subcarpetas, etc) entre la máquina servidor y todos los equipos cliente. En el equipo servidor, la carpeta **Resources** simplemente debe estar ubicada junto al archivo de estructura de la base.

Todos los mecanismos de referenciamiento asociados a la carpeta **Resources** son soportados en modo cliente/servidor (carpeta .lproj, XLIFF, imágenes, etc). Para obtener más información sobre este punto, consulte el Manual de Diseño 4D.

Cada cliente tiene en local una copia de esta carpeta. El contenido de la carpeta local se sincroniza automáticamente con la del servidor cada vez que el cliente se conecta.

Por otra parte, los equipos cliente pueden ser "notificados" dinámicamente durante una sesión cuando el contenido de la carpeta **Resources** de la base servidor sea modificado por un desarrollador. Esta notificación puede activarse:

- De forma automática por el servidor, dos minutos después de la última modificación realizada por un cliente (esto ayuda a evitar las notificaciones inoportunas en caso de copia de numerosos archivos).
- O de forma manual a través del comando **Notify clients** en el menú de acción del Explorador de recursos en el equipo cliente en el origen de la modificación.
- O por la programación, a través de un comando **NOTIFY RESOURCES FOLDER MODIFICATION**. Este comando es útil cuando el contenido de la carpeta Resources se modifica en el servidor a través de un procedimiento almacenado.

Del lado del cliente, la forma en que se maneja la notificación de las modificaciones depende de la configuración de la preferencia "Actualización de la carpeta Resources durante una sesión". Esto también se puede definir individualmente mediante el comando **SET DATABASE PARAMETER**. Tres opciones están disponibles: **sin sincronización**, **sincronización automática** o **preguntar**. Para obtener más información, consulte la sección **Opciones red y cliente-servidor** Preferencias de configuración y la descripción del comando **SET DATABASE PARAMETER**. Por último, cada máquina cliente puede sincronizarse con el servidor en cualquier momento a través del comando **Update Local Resources** en el menú de acción del Explorador de recursos. Para obtener más información sobre el Explorador de recursos, consulte el Manual de Diseño 4D.

Nota de compatibilidad: en versiones anteriores de 4D Server, la transferencia de datos personalizados se llevaba a cabo a través de una carpeta llamada "Extras", ubicada al lado del archivo de estructura. Esta carpeta es obsoleta y su uso no es recomendable. Sin embargo, sigue siendo soportada por 4D Server con el fin de preservar la compatibilidad de las aplicaciones existentes.

Registrar una base como servicio

Bajo Windows, 4D Server puede lanzarse como un Servicio

Nota de compatibilidad: esta función no está disponible bajo Mac OS a partir de la versión 12 de 4D Server.

Una aplicación 4D Server registrada como servicio se lanza automáticamente al inicio de la máquina con la base actual, incluso antes de que se abra una sesión de usuario. No se cierra cuando el usuario sale de la sesión.

Este funcionamiento le permite garantizar la disponibilidad de una base 4D Server, incluso en el evento de un incidente que requiera reiniciar la máquina. El mantenimiento puede efectuarse de manera remota.

Notas:

- En una plataforma Windows 64-bits, una aplicación 4D Server registrada como servicio se ejecuta sin interfaz (la ventana de administración del servidor no se muestra más).
- Para mayor información sobre los mecanismos de gestión de los Servicios, consulte la documentación de su sistema, consulte la documentación de su sistema operativo.

Para registrar la base de datos 4D Server como un servicio, seleccione el comando **Dar de alta la base actual como servicio** en el menú **Archivo** de 4D Server. La próxima vez que se inicie la máquina, 4D Server se lanzará automáticamente y la base de datos actual se abrirá. Puede registrar tantas bases de datos como quiera. Cada base puede registrarse sólo una vez.

Nota: bajo Windows, este comando puede volverse gris cuando el acceso a las funciones de gestión de los servicios esté restringido. Para poder utilizar este comando, debe lanzar 4D Server con un nivel administrador (para hacer esto, haga clic derecho en el icono de la aplicación y elija el comando **Ejecutar como administrador** en el menú contextual).

Atención: asegúrese de utilizar una cuenta válida cuando abra la sesión, de lo contrario se mostrará un mensaje de error. Por defecto, 4D Server se ejecuta con la "Cuenta sistema local", la cual puede no necesariamente contar con los parámetros necesarios para utilizar su aplicación. En particular, si desea efectuar impresiones, debe abrir la sesión con una cuenta de usuario que tenga parámetros de impresión por defecto. El mismo problema ocurre si quiere acceder a los volúmenes red. Le recomendamos cambiar de cuenta de usuario: para esto, abra **Panel de control > Sistema y seguridad > Herramientas de administración > Servicios**. En la lista de Servicios, haga clic en 4D Server, elija la opción **Propiedades**, y luego vaya a la pestaña **Conexión** y defina la cuenta con la que el servidor debe ejecutarse (configuración que se utilizará en el próximo lanzamiento).

Para dar de baja su base, seleccione **Dar de baja la base actual** del menú **Archivo** de 4D Server. Este comando está en gris si la base no se ha registrado como un servicio.

Para dar de baja todas las bases de 4D Server a la vez, seleccione **Dar de baja todos los servicios de 4D server** a partir del menú **Archivo** de 4D Server. Este comando está en gris si ningún servicio 4D Server está activo.

No es posible cambiar el estado del registro del servicio de 4D Server desde 4D Server, si la aplicación se ha lanzado como un servicio al inicio. En este caso, los tres elementos del menú se desactivan. Para detener el servicio, utilice el panel de control de Servicios.

Configurar un espejo lógico

4D Server ofrece una solución integrada que permite configurar un sistema de backup vía un espejo lógico. Esta solución está basada en dos comandos: **New log file** y **INTEGRATE MIRROR LOG FILE**.

¿Qué es un espejo lógico?

Un espejo lógico es un modo backup sofisticado, destinado especialmente para bases de datos críticas o de carga alta.

El uso de un espejo lógico consiste en operar una base de datos en una máquina y mantener en una segunda máquina una copia de esta base que se actualiza periódicamente. Ambas máquinas se comunican por la red, la máquina en funcionamiento transmite regularmente a la máquina espejo los cambios de los datos por intermedio del archivo de historial.

De esta manera, cuando hay un incidente que afecta a la base de datos operacional, la base de datos espejo se puede utilizar para conseguir rápidamente volver a trabajar sin pérdida de datos. Por otra parte, la base de datos operacionales nunca está "bloqueada" por las operaciones de backup.

¿Por qué elegir el backup por espejo lógico?

El uso de un espejo lógico corresponde a necesidades concretas. La estrategia estándar basada en copias de seguridad periódicas y el uso de un archivo de historial constituye en la mayoría de los casos una solución simple, fiable y económica. Se hace backup de la base regularmente (cada 24 horas en general). Durante la copia de seguridad, todos los procesos se congelan. Este período de no disponibilidad parcial es muy corto, e incluso en el caso de bases de datos grandes (mayores de 2 GB), no dura más de 5 minutos. Esta operación puede programarse para tener lugar fuera de los períodos de uso de la base.

Sin embargo, para ciertos tipos de organizaciones, tales como hospitales, por ejemplo, las bases de datos críticas debe ser totalmente operacionales las 24 horas del día. La base no puede estar en "modo copia de seguridad", aunque sea por un período de tiempo muy corto. En este caso, la creación de un espejo lógico es una buena solución.

Nota: la base espejo sólo refleja los cambios realizados a los datos. Este modo de backup no es adecuado para bases en proceso de desarrollo, donde las frecuentes modificaciones estructurales hará que el espejo rápidamente sea obsoleto o que requieren múltiples actualizaciones de la estructura de la base espejo.

Principios de funcionamiento

La creación de un sistema de backup por espejo lógico se basa en dos nuevos comandos: **New log file** y **INTEGRATE MIRROR LOG FILE**. Estos comandos se describen en el manual Lenguaje de 4D.

Los principios siguientes aplican:

- La base está instalada en el equipo 4D Server principal (máquina de funcionamiento) y una copia idéntica de la base está instalada en la máquina 4D Server espejo.
- Una prueba al inicio de la aplicación (por ejemplo, para detectar la presencia de un archivo específico en una subcarpeta de la aplicación 4D Server) permite distinguir cada versión (operacional y espejo) y por tanto ejecutar las operaciones apropiadas.
- En la máquina 4D Server en funcionamiento, el archivo de historial es "segmentado" a intervalos regulares con el comando **New log file**. Dado que ninguna copia de seguridad se lleva a cabo en el servidor principal, la base se mantiene disponible permanentemente en modo lectura-escritura.
- Cada "segmento" del archivo de historial se envía a la máquina espejo, donde se integra a la base espejo utilizando el comando **INTEGRATE MIRROR LOG FILE**.

La creación de este sistema necesita la programación de código específico, en particular:

- Un contador de tiempo en el servidor principal para la gestión de los ciclos de ejecución del comando **New log file**,
- Un sistema de transferencia para los "segmentos" del archivo de historial entre la máquina operacional y la máquina espejo (usando 4D Internet Commands para una transferencia vía FTP o por sistemas de mensajería, servicios web, etc),
- Un proceso en la máquina espejo destinado a supervisar la llegada de nuevos "segmentos" del archivo de historial y de integrarlos usando el comando **INTEGRATE MIRROR LOG FILE**,
- Un sistema de comunicación y de gestión de errores entre el servidor principal y el servidor espejo.

ATENCIÓN: un sistema de backup por espejo lógico no es compatible con los backup "estándar" en una base en uso ya que el uso simultáneo de estos dos modos de backup dará lugar a la desincronización de la bases operacional y espejo. Por consiguiente, debe estar seguro de que ningún backup, automático o manual, se efectúe en la base operacional. Por otra parte, es posible hacer backups de la base espejo o establecer un "espejo de espejo" (ver el siguiente párrafo).

Backup de la base espejo y espejo de espejo

4D Server permite efectuar backups de la base en la máquina espejo. Todos los medios convencionales se pueden utilizar para realizar backups en la máquina espejo: backup manual vía el comando de menú **Archivo**, backup periódico definido en las Propiedades de la base o backup programado utilizando los comandos del lenguaje.

Para evitar riesgos de desincronización con la máquina operacional, 4D bloquea automáticamente la máquina espejo cuando se está llevando a cabo una de las dos operaciones básicas: integración del archivo de historial de la máquina operacional y backup de la base espejo.

- Durante la integración del archivo de historial, no es posible llevar a cabo un backup. Si se utiliza el comando **BACKUP**, se genera el error 1417 (ver la sección **Errores de gestión de backup**).
- Cuando un backup está en marcha, todos los procesos se congelan y no es posible poner en marcha la integración de un archivo de historial.

A partir de 4D v14, se puede activar el archivo de historial actual en la máquina espejo, lo que significa que puede configurar un "espejo de espejo" (ver espejos en serie), o incluso una arquitectura de espejos "hub-and-spoke" (varios espejos para la misma base en operación). En el primer caso, el archivo de historial actual del espejo se envía a su vez a otro espejo (el "espejo del espejo") para integración, y así sucesivamente si se utiliza una serie de espejos.

En el segundo caso, el historial actual se envía directamente a varios servidores espejo idénticos. Este tipo de redundancia asegura la disponibilidad continua del servidor, incluso en el caso de fallo simultáneo del servidor y del espejo principal.

Escenario de operación de un espejo lógico

El siguiente escenario ilustra, desde el punto de vista de cada equipo 4D Server, la configuración y el funcionamiento de un sistema de backup con espejo:

Paso	Equipo en operación	Equipo espejo
1	<p>Arranque de la aplicación, back up del archivo de datos. El archivo de historial se activa por defecto; para mayor seguridad, almacenar este archivo en un disco duro separado.</p> <p>4D crea el archivo MyDatabase.journal. Para mayor seguridad, el archivo de historial se guarda en un disco duro separado.</p> <p>Salimos de la aplicación.</p> <p>Copia de todos los archivos de la base (archivo de historial incluido) en el equipo espejo.</p>	<p>Inicio de la aplicación espejo. 4D Server solicita el archivo de historial actual: selección del archivo MyDatabase.journal que fue transferido de la base operacional. Este archivo se utilizará en caso de implementación de un espejo de espejo.</p>
2	<p>Reinicio de la aplicación e inicio de la operación (verificar que no haya un backup total programado).</p>	
3	<p>Decisión de actualizar el espejo (por ejemplo, después de un cierto periodo de operación).</p> <p>Ejecución del método que contiene el comando New log file. El archivo guardado se llama MyDatabase[0001-0001].journal.</p> <p>Envío por programación del archivo MyDatabase[0001-0001].journal al equipo espejo (utilizando 4DIC, Servicios web, etc.).</p> <p>La base está en operación.</p>	
4		<p>Detección de un archivo que está esperando a ser integrado. Ejecución del método que contiene el comando INTEGRATE MIRROR LOG FILE para integrar el archivo MyDatabase[0001-0001].journal. Si utiliza un espejo de espejo, ejecución en el equipo espejo de un procedimiento similar a la etapa 3 (a repetir a cada integración del historial).</p>

- 5 Incidente en el equipo; la base es inutilizable. Decisión de pasar al equipo espejo. Copia del archivo de historial actual MyDatabase.journal al equipo espejo, vía la carpeta de destino habitual.
- 6 Análisis del incidente y reparación. Detección de un archivo que está esperando a ser integrado. Ejecución del método que contiene el comando **INTEGRATE MIRROR LOG FILE** para integrar el archivo MyDatabase.journal. La base está en operación.
- 7 La máquina se repara. Reemplazo de los archivos de la base por los de la base espejo. Inicio de la aplicación. 4D Server solicita el archivo de historial: selección del archivo transferido desde el equipo espejo. Salimos de la base. Volver al paso 2.

Utilizar Servicio Volume Shadow Copy en Windows

Generalidades

4D Server Windows 64 bits viene con una aplicación **VSS writer** dedicada que gestionará automáticamente las solicitudes de instantáneas enviadas a través del Servicio Windows Volume Shadows Copy (VSS).

VSS es ofrecido por Windows Server para permitir que las aplicaciones de copia de seguridad capturen instantáneas (copias sombra) de todos los archivos o discos duros enteros en un momento dado, mientras las aplicaciones se ejecutan. Gracias a esta tecnología, puede recuperar por ejemplo una base de datos 4D Server en el estado exacto que era en el momento de la instantánea. Este mecanismo requiere que los archivos de aplicación en ejecución estén en un estado consistente cuando se realice la instantánea. Por esta razón, una aplicación VSS-aware debe instalar una aplicación VSS writer o servicio. Este componente es "advertido" por el servicio cuando una copia sombra está a punto de ser realizarse y le dice al **VSS requestor** (básicamente la aplicación de copia de seguridad) cómo hacer una copia de seguridad de su archivo y datos.

Nota: para una descripción general de las demás funcionalidades de backup y seguridad de 4D, consulte la [guía de seguridad de 4D](#).

Requerimientos para el virtualizador

Del lado del host, son soportados los siguientes requestors VSS:

- **VMware ESXI** en cualquier plataforma
- **Microsoft Hyper-V Server 2016**

Requerimientos para 4D Server

4D Server soporta Microsoft Windows VSS con la siguiente configuración:

- 4D Server (engined o no)
- Versión 64 bits
- Línea de productos Windows Server versión 2012 R2 o superior.

4D Server instala automáticamente el servicio VSS en la línea de productos Windows Server.

Otras versiones Windows que soportan VSS son compatibles pero no certificadas; en estas versiones, el servicio VSS debe activarse durante la instalación:

- Windows 7,
- Windows 8.1,
- Windows 10 (UA)

Activar el VSS

La funcionalidad VSS se instala/actualiza automáticamente cuando se inicia la aplicación 4D Server. El servicio de aplicación VSS writer se inicia en las siguientes condiciones:

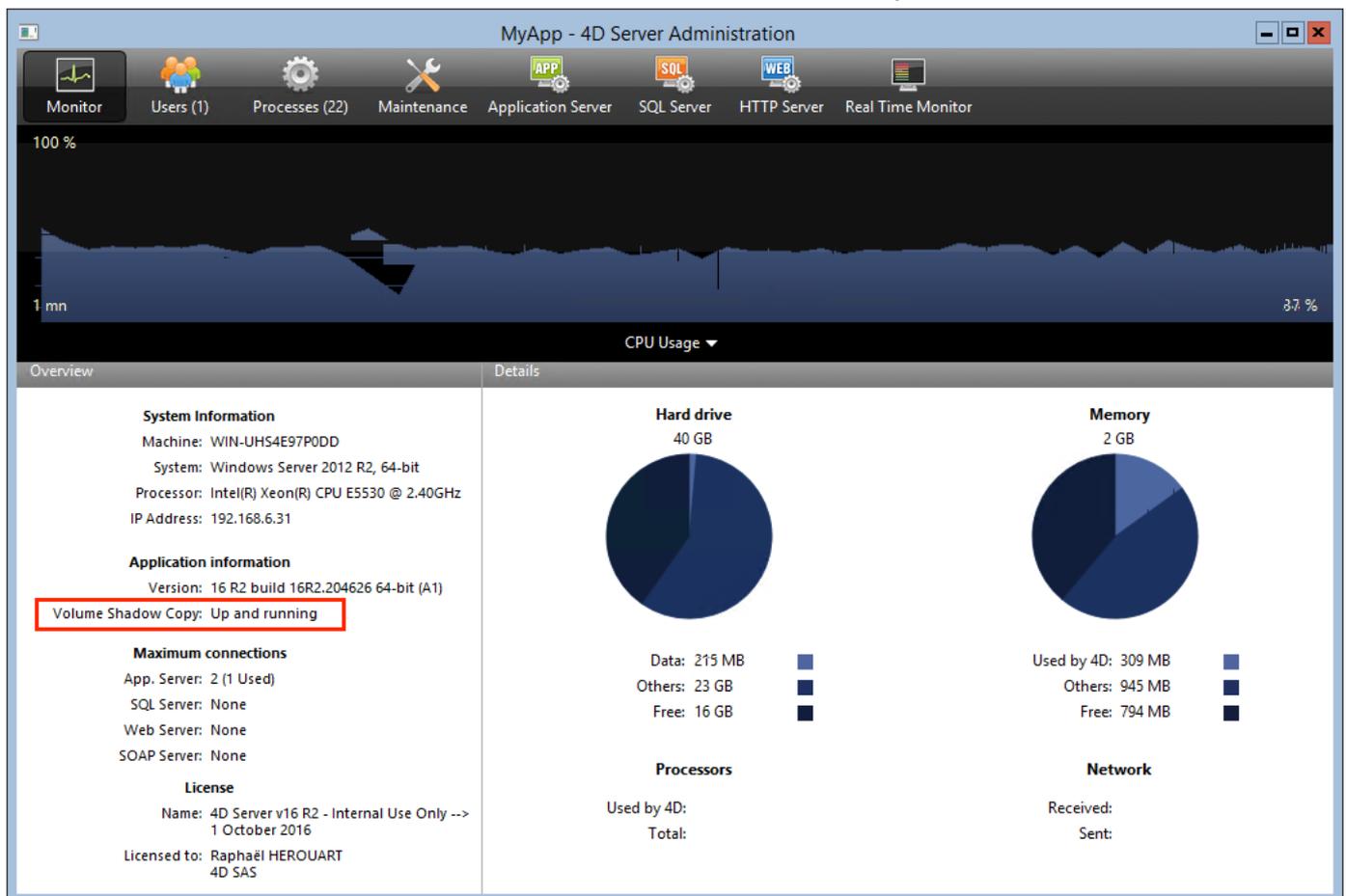
- la configuración de 4D Server es compatible (ver arriba).
- el usuario de la sesión tiene privilegios de administrador.

Por lo general, el escenario inicial será:

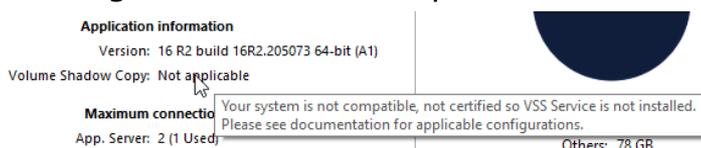
1. 4D Server o Aplicación Server fusionada se lanza por primera vez.
2. Si no se lanza con privilegios de administración, se muestra un icono de advertencia.
3. Cierre y reinicie 4D Server o una Aplicación Server fusionada como administrador.
El servicio 4D VSS se ejecuta automáticamente y se registra en VSS.
4. (Opcional) Reinicie 4D Server o la Aplicación Server fusionada utilizando privilegios estándar.

El ejecutable VSS writer se inicia como un servicio con el nombre "VSS <appName>". Un servicio VSS se ejecutará para todas las instancias de 4D Server. Un servicio VSS se ejecutará para cada aplicación de motor diferente (nombre diferente) que se ejecuta en la máquina (ver abajo).

Página Monitor de la ventana de administración de 4D Server muestra el estado del servicio VSS writer, en el área de información de la Aplicación:



Información adicional sobre el estado Volume Shadow Copy se puede mostrar en una sugerencia cuando se pasa el ratón sobre el área:



Sobre VSS Writer

La aplicación **vss_writer.exe** se ofrece para gestionar el servicio VSS (Volume Shadow Copy Service) para aplicaciones 4D.

Nota: la gestión 4D VSS se maneja a través de una aplicación separada, ya que este programa debe ejecutarse utilizando privilegios de administración.

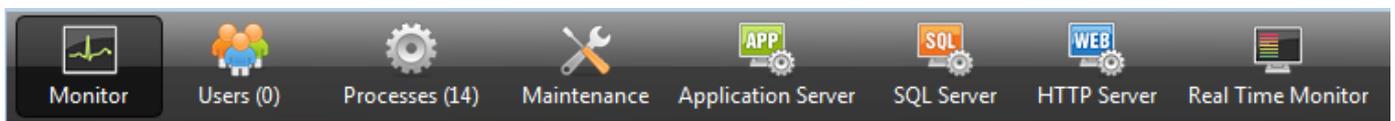
El ejecutable 4D VSS writer se instala automáticamente por 4D Server en el primer lanzamiento

El servicio 4D VSS Writer gestiona y transfiere mensajes VSS a 4D Server. Estos mensajes se registran en el historial de diagnóstico de 4D Server y en el visor de eventos de Windows.

Ventana de administración de 4D Server

4D Server tiene una ventana de administración completa y fácil de usar.

Esta ventana ofrece diferentes herramientas de análisis y control para bases de datos publicadas. La ventana contiene varias páginas que se pueden acceder utilizando los botones en la parte superior:



Cada página se detalla en una sección de este capítulo.

Nota: La ventana de administración se puede acceder desde un equipo 4D remoto. Para obtener más información acerca de este punto, consulte la sección **Administración de máquinas remotas**.

-  Página Monitor
-  Página Usuarios
-  Página Procesos
-  Página Mantenimiento
-  Página Servidor de aplicaciones
-  Página Servidor SQL
-  Página Servidor HTTP
-  Página Monitor tiempo real

Página Monitor

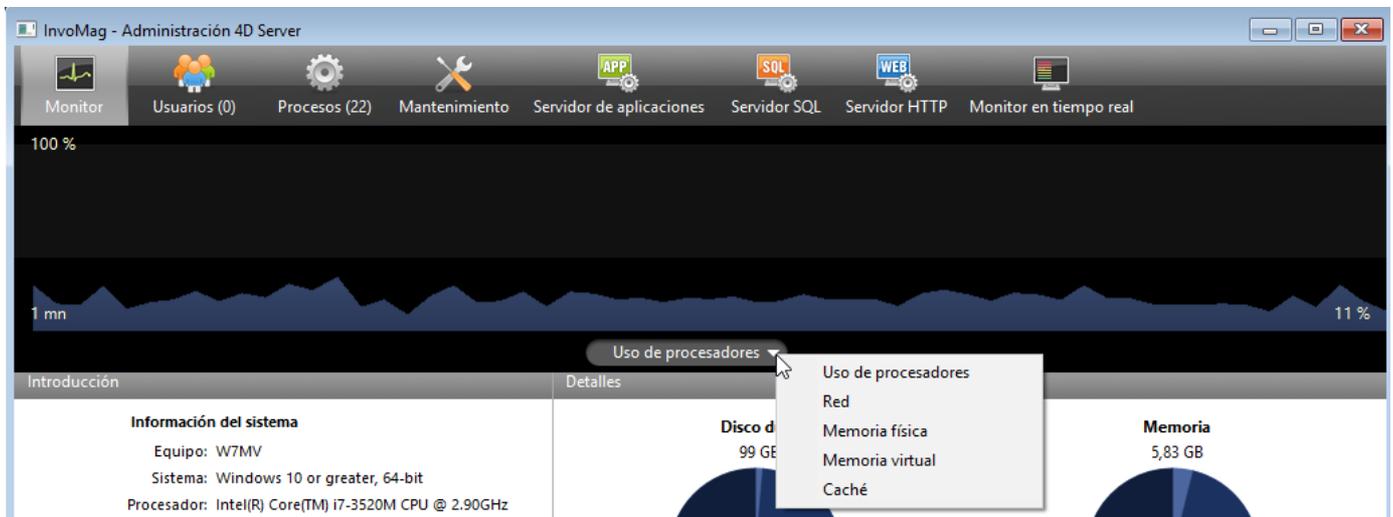
La página Monitor muestra información dinámica sobre el uso de la base así como también información del sistema y la aplicación 4D Server:



Nota: bajo Windows, la visualización de esta información está relacionada con los derechos del usuario que abrió la sesión. Para mayor información, consulte el párrafo “Visualización de información del monitor (Windows)”.

Área gráfica

El área gráfica permite visualizar la evolución en tiempo real de varios parámetros: el uso de los procesadores, el tráfico de red y la ocupación de la memoria. Usted selecciona el parámetro a mostrar utilizando el menú ubicado al centro de la ventana:



- **Uso de procesadores:** uso global de CPU de la máquina, para todas las aplicaciones juntas.
La parte específica de 4D Server en esta tasa de uso está en el área de información "Procesadores".
- **Red:** número de bytes recibidos por segundo por 4D Server.
El número de bytes enviados por 4D Server se ofrece en el área de información "Red".
- **Memoria física:** cantidad de memoria RAM de la máquina utilizada por 4D Server.
Una vista más detallada del uso de la memoria se encuentra en el área de información "Memoria".
- **Memoria Virtual:** muestra en el área gráfica la cantidad de memoria virtual utilizada por la aplicación 4D Server. Esta memoria es asignada por el sistema de acuerdo a las necesidades de la aplicación. El valor en la parte inferior derecha del área indica la cantidad de memoria en uso. El valor en la parte superior izquierda indica la cantidad máxima de memoria virtual utilizable. El valor máximo se calcula dinámicamente en función de los parámetros de memoria generales de la aplicación.
- **Caché:** muestra en el área gráfica la cantidad de memoria caché utilizada por la aplicación 4D Server. El valor en la parte inferior derecha del área indica la cantidad de memoria en uso. El valor en la parte superior izquierda indica el tamaño total de la memoria caché, tal como se definió en las Propiedades de la base.
Note que cuando esta opción está seleccionada, el desplazamiento del área gráfica se ralentiza ya que un análisis eficaz de la caché se efectúa generalmente en un periodo de observación bastante largo.

Área Introducción

El área "Introducción" ofrece diferente información relativa al sistema, la aplicación y las licencias instaladas en la máquina de 4D Server.

Información del sistema

Equipo: W7MV
 Sistema: Windows 10 or greater, 64-bit
 Procesador: Intel(R) Core(TM) i7-3520M CPU @ 2.90GHz
 Dirección IP: Todo

Información de la aplicación

Versión: 16 R6 build 16R6.217210 64 bits (A1)
 Volume Shadow Copy: No aplica

Conexiones máximas

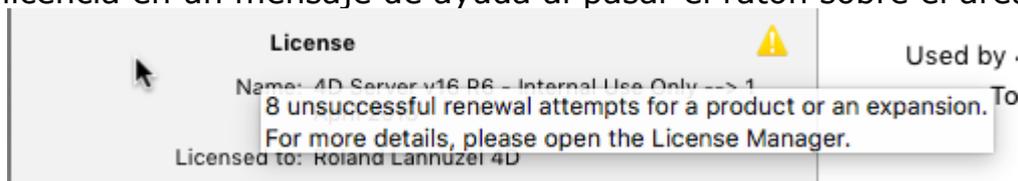
Servidor de aplicaciones: 10(0 Usado(s))
 Servidor SQL: 8
 Servidor Web: Ilimitadas
 Servidor SOAP: Ilimitadas

Licencia

Nombre: 4D Team Developer Professional v16 R6 -
 Licenciado a:

- **Información del sistema:** ordenador, sistema y dirección IP del servidor
- **Información de la aplicación:** número de versión interno del servidor 4D y estado de Volume Shadow Copy (ver la sección **Utilizar Servicio Volume Shadow Copy en Windows**).
- **Máximo de conexiones:** número de conexiones simultáneas permitidas por tipo de servidor
- **Licencia:** descripción de la licencia.

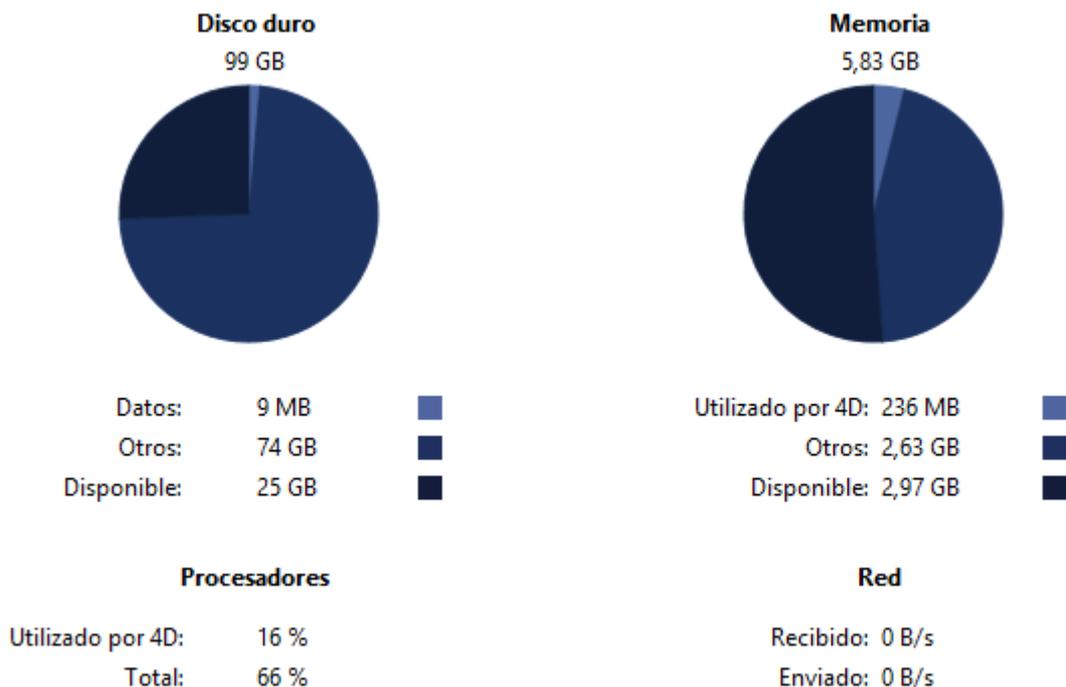
Cuando la licencia del producto o una de sus expansiones expira en menos de 10 días, por ejemplo en caso de una licencia de suscripción, 4D Server intenta renovar automáticamente la licencia desde la cuenta de usuario 4D. En este caso, si la renovación automática falla por algún motivo (error de conexión, estado de cuenta no válido, contrato no prolongado etc.), se muestra un icono de advertencia junto a la licencia para alertar al administrador del servidor. Se puede mostrar información adicional sobre el estado de la renovación de la licencia en un mensaje de ayuda al pasar el ratón sobre el área:



Por lo general, deberá verificar el Administrador de licencias (ver).

Área Detalles

El área "Detalles" repite parte de la información mostrada en el área gráfica y ofrece información adicional.



- **Disco duro:** capacidad global del disco duro y distribución entre el espacio ocupado por los datos de la base (archivo de datos + índice de datos), el espacio utilizado por otros archivos y el espacio disponible.
- **Memoria:** memoria RAM instalada en la máquina y cantidad de memoria ocupada por 4D Server, por las otras aplicaciones y memoria disponible. La memoria ocupada por 4D Server puede igualmente mostrarse dinámicamente en el área gráfica.
- **Procesadores:** tasa instantánea de ocupación de los procesadores de la máquina por 4D Server y por las otras aplicaciones. Esta tasa se recalcula permanentemente. La tasa de ocupación por 4D Server también puede mostrarse dinámicamente en el área gráfica.
- **Red:** número instantáneo de bytes recibidos vía la red por 4D Server y número de bytes enviados por la aplicación. Este valor se actualiza constantemente. El número de bytes recibidos por 4D Server también se pueden mostrar dinámicamente en el área gráfica.

Mostrar información del monitor (Windows)

En Windows, alguna información del sistema mostrada en la página Monitor se recupera vía las herramientas del "Monitor de rendimiento" de Windows. El acceso a estas herramientas sólo está permitido cuando el usuario que abre la sesión donde se lanzó 4D Server tiene las autorizaciones necesarias. Este usuario debe:

- pertenecer al grupo "Administradores",
- bajo Windows Vista: pertenecer al grupo "Usuarios del monitor de rendimiento" (para un usuario que no es Administrador).

Para poner un usuario no administrador en el grupo "Usuarios del analizador de rendimiento" bajo Windows Vista (necesitará utilizar una cuenta Administrador para efectuar estas operaciones):

1. Vaya al Panel de control y abra el panel "Cuentas de usuario".
2. Haga clic en la pestaña "Opciones Avanzadas" y luego en el botón "Avanzado" de la sección "Gestión avanzada de usuarios".
Se ejecuta la aplicación "Usuarios y grupos locales".
3. Doble clic en la carpeta "Grupos".
4. Doble clic en el grupo "Usuarios del monitor de rendimiento".
Aparece una ventana llamada "Propiedades de usuarios del monitor de rendimiento".
5. Haga clic en el botón **Añadir...** para añadir un usuario.
6. En el área de texto titulada "Introduzca los nombres de los objetos a seleccionar," introduzca los nombres de los usuarios a autorizar.
7. Haga clic en **Aceptar** (dos veces).
Cierre la aplicación "Usuarios y grupos locales" luego las "Cuentas de usuarios."

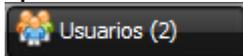
Página Usuarios

La página Usuarios lista los usuarios conectados a la base:

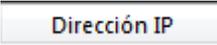


4D Usuario	Equipo	Nombre de sesión	Dirección IP	Conexión	Tiempo CPU	Actividad
 Diseñador	ST1	soporte	localhost	1/06/2016 11:40	00:00:00	0%
 Diseñador	Centaur0	Angelo	192.168.1.117	1/06/2016 12:24	00:00:00	0%

El botón "Usuarios" indica, entre paréntesis, el número total de usuarios conectados a la base (este número no tiene en cuenta los posibles filtros de visualización aplicados a la ventana).



Esta página también contiene un área de búsqueda dinámica y los botones de control.

Puede modificar el orden de las columnas arrastrando y soltando el área de encabezado de las columnas. También puede ordenar la lista de valores de una columna haciendo clic en su encabezado: 

Haga clic varias veces para definir alternadamente un orden ascendente/descendente.

Lista de usuarios

Para cada usuario conectado a la base, la lista ofrece la siguiente información:

- Sistema de la máquina cliente (Mac OS o Windows) forma de icono.
- Usuario 4D: nombre de usuario 4D. Si las contraseñas no están activas, todos los usuarios se llaman "Diseñador."
- Nombre de máquina: nombre del equipo cliente.
- Nombre de sesión: nombre de la sesión abierta en el equipo cliente.
- Dirección IP: dirección IP de la máquina cliente.
- Conexión: fecha y hora de la conexión del equipo cliente.
- Tiempo CPU: tiempo de CPU consumido por este usuario desde la conexión.
- Actividad: radio de tiempo que 4D Server otorga a este usuario (visualización dinámica). "Dormir" si la máquina del cliente ha pasado a modo dormir (ver abajo).

Gestión de usuarios dormidos

4D Server trata específicamente los casos en que un equipo que ejecuta una aplicación remota 4D cambia a modo dormir mientras su conexión a la máquina servidor todavía está activa. En este caso, la aplicación remota 4D conectada notifica automáticamente a 4D Server de su desconexión inminente.

En el servidor, el usuario conectado cambia a un estado de actividad **Dormir**:

4D Usuario	Equipo	Nombre de sesión	Dirección IP	Conexión	Tiempo CPU	Actividad
Diseñador	Centaurus	Angelo	192.168.1.117	5/05/2016 07:47	00:00:00	Dormido
Daniele Namath	ST1	soporte	localhost	5/05/2016 07:43	00:00:00	0%

Este estado libera recursos en el servidor. Además, la aplicación 4D remota vuelve a conectarse a 4D Server de forma automática después de despertar del modo dormir. El siguiente escenario es soportado: un usuario remoto deja de trabajar por un tiempo, por ejemplo durante una pausa para el almuerzo, pero mantiene la conexión con el servidor abierta. La máquina pasa al modo dormir. Cuando el usuario vuelve, despierta la máquina y la aplicación remota 4D recupera automáticamente la conexión con el servidor, así como el contexto de sesión.

Área de búsqueda/filtro

Esta función permite reducir el número de líneas mostradas en la lista a las que corresponden al texto introducido en el área de búsqueda. El área indica las columnas en las cuales se efectuará la búsqueda/filtro. En la página Usuarios, serán las columnas Usuario 4D, Nombre de máquina y Nombre de sesión.

La lista se actualiza en tiempo real a medida que introduce el texto en el área.

Es posible introducir más de un valor a buscar. Utilice punto y coma para separar los valores. El operador utilizado en este caso es de tipo O.

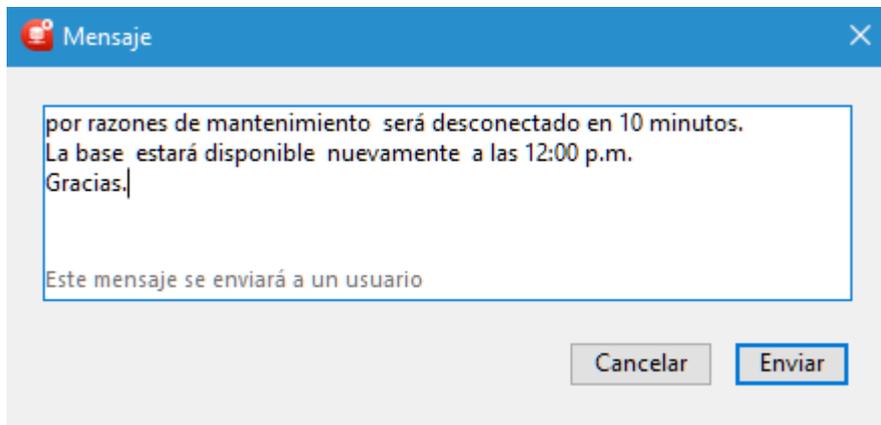
Por ejemplo, si introduce "Juan;María;Pedro," sólo se conservarán las líneas con Juan O María O Pedro en las columnas objetivo.

Botones de administración

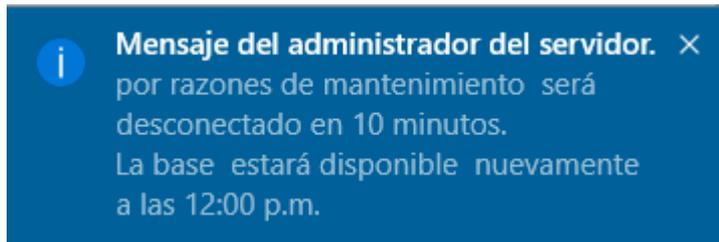
Esta página incluye tres botones de control. Estos botones están activos si al menos una línea está seleccionada. Puede seleccionar varias líneas presionando la tecla Mayús para una selección adyacente o **Ctrl** (Windows) / **Comando** (Mac OS) para una selección no adyacente.

- **Enviar mensaje:** este botón permite enviar un mensaje a los usuarios 4D seleccionados en la ventana. Si ningún usuario 4D está seleccionado, el botón se desactiva.

Al hacer clic en este botón, aparece una caja de diálogo que le permite introducir el mensaje. La caja de diálogo indica el número de usuarios que recibirán el mensaje:



El mensaje se mostrará en forma de alerta en los equipos clientes:

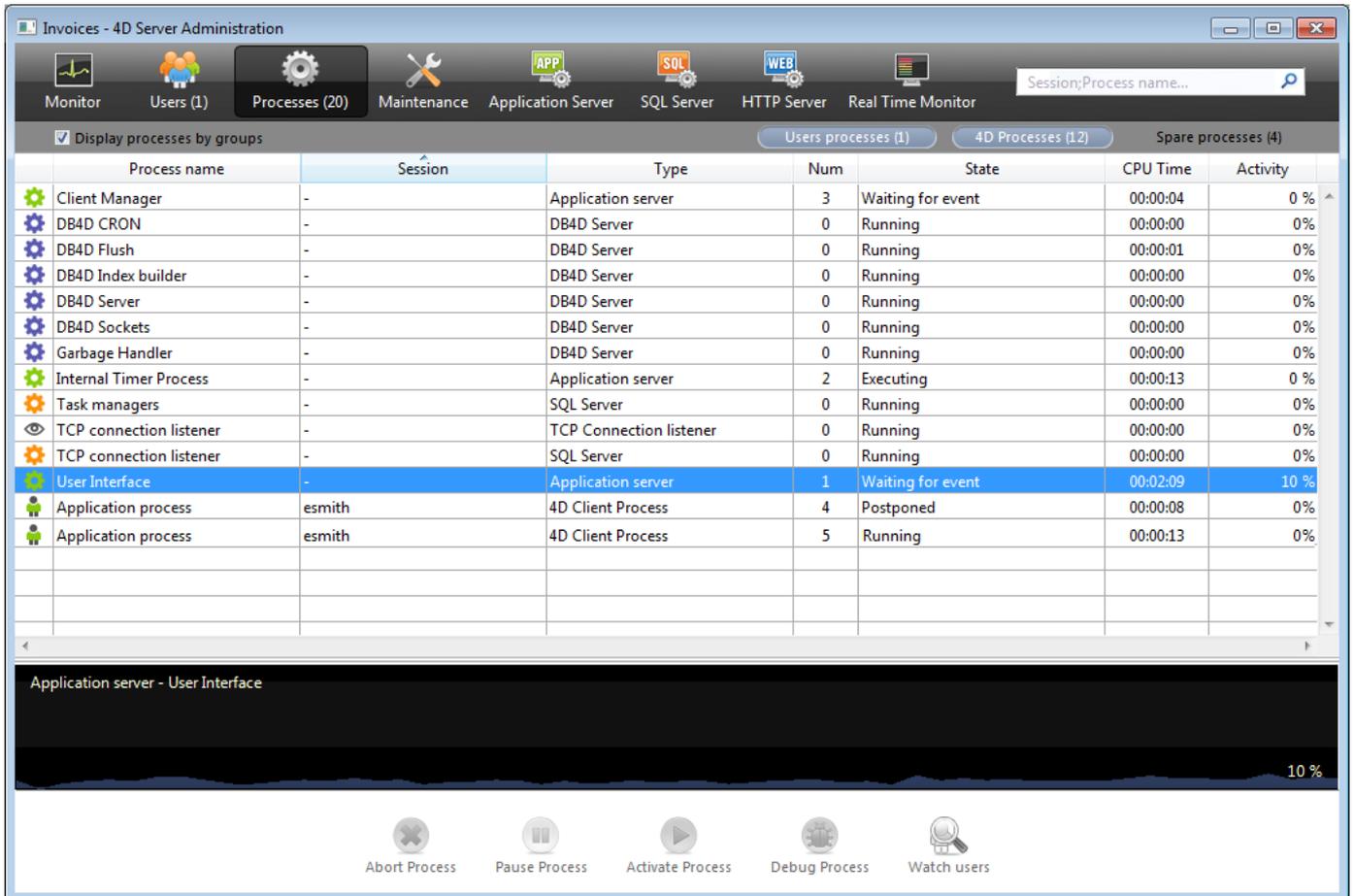


- Visualizar procesos: este botón permite visualizar directamente los procesos de usuario seleccionado en la página Proceso de la ventana de administración. Al hacer clic en el botón, 4D Server pasa a la página Procesos e introduce los nombres de usuarios seleccionados en el área de buscar/filtrar de esta página. Para mayor información, consulte la descripción de esta página.
- Desconectar: este botón puede utilizarse para forzar la desconexión de los usuarios seleccionados. Cuando hace clic en este botón, aparece una caja de diálogo de alerta que le permite confirmar o anular la operación.

Nota: puede también desconectar directamente los usuarios seleccionados sin mostrar la caja de diálogo de confirmación: para hacer esto, presione **Alt** (Windows) u **Opción** (Mac OS) mientras presiona el botón **Desconectar**.

Página Procesos

La página Procesos lista los procesos en ejecución:



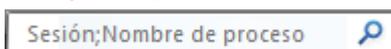
Process name	Session	Type	Num	State	CPU Time	Activity
Client Manager	-	Application server	3	Waiting for event	00:00:04	0 %
DB4D CRON	-	DB4D Server	0	Running	00:00:00	0%
DB4D Flush	-	DB4D Server	0	Running	00:00:01	0%
DB4D Index builder	-	DB4D Server	0	Running	00:00:00	0%
DB4D Server	-	DB4D Server	0	Running	00:00:00	0%
DB4D Sockets	-	DB4D Server	0	Running	00:00:00	0%
Garbage Handler	-	DB4D Server	0	Running	00:00:00	0%
Internal Timer Process	-	Application server	2	Executing	00:00:13	0 %
Task managers	-	SQL Server	0	Running	00:00:00	0%
TCP connection listener	-	TCP Connection listener	0	Running	00:00:00	0%
TCP connection listener	-	SQL Server	0	Running	00:00:00	0%
User Interface	-	Application server	1	Waiting for event	00:02:09	10 %
Application process	esmith	4D Client Process	4	Postponed	00:00:08	0%
Application process	esmith	4D Client Process	5	Running	00:00:13	0%

El botón "Procesos" indica entre paréntesis el número total de procesos en ejecución en la base (este número no tiene en cuenta los posibles filtros de visualización aplicados a la ventana ni el estado de la opción **Mostrar los procesos por grupos**).



Puede cambiar el orden de las columnas simplemente arrastrando y soltando las áreas de encabezado de las columnas. También puede ordenar la lista de los valores de una columna haciendo clic en su encabezado.

Como la **Página Usuarios4D for OCI**, esta página contiene un área de búsqueda/filtrado dinámica, que permite reducir el número de líneas mostradas en la lista al que corresponda al texto introducido en el área de búsqueda. La búsqueda/filtrado se efectúa en las columnas Sesión y Nombre de proceso.



También hay tres botones de atajos que permiten filtrar por tipo los procesos mostrados en la ventana:

- Procesos de usuario: procesos generados por y para las sesiones usuario. Estos procesos están precedidos por un icono en forma de persona.
- Procesos 4D: procesos generados por el motor de 4D Server. Estos procesos están precedidos por un icono en formato de rueda dentada.
- Procesos libres: procesos inactivos pero que se conservan temporalmente y que pueden reutilizarse en cualquier momento. Este mecanismo optimiza la reactividad de 4D Server. Estos procesos están precedidos por un icono gris en forma de persona.

La opción **Mostrar los procesos por grupos** permite reagrupar los procesos internos de 4D Server como también los procesos cliente, para mayor legibilidad. Cuando selecciona esta opción:

- los procesos clientes 4D "twinned" (Procesos cliente 4D principal y Procesos base 4D client, ver el párrafo "Tipo de proceso") se agrupan en uno solo,
- se crea un grupo "Administradores de tareas"; el cual incluye los procesos internos dedicados a la repartición de las tareas (Shared balancer, Net session manager, Exclusive pool worker),
- se crea el grupo "Gestor de clientes"; el cual incluye los diferentes procesos internos clientes.

El área inferior de la ventana permite mostrar la representación gráfica de la actividad de los procesos seleccionados:



Nota: puede seleccionar varias líneas presionando la tecla **Mayús** para una selección adyacente o **Ctrl** (Windows) / **Comando** (Mac OS) para una selección no adyacente.

La actividad del proceso es el porcentaje de tiempo que 4D Server ha otorgado a este proceso (radio).

Como en las versiones anteriores de 4D Server, la ventana ofrece la siguiente información para cada proceso:

- Tipo de proceso (ver a continuación),
- Sesión (vacío en caso de un proceso 4D y nombre de usuario 4D en caso de un proceso usuario),
- Nombre del proceso,
- Número del proceso (tal como lo devuelve la función **New process** por ejemplo). El número del proceso es el número asignado por el servidor. En el caso de un proceso global, este número puede ser diferente del asignado en el equipo cliente.
- Estado actual del proceso,
- Tiempos (en segundos) de ejecución del proceso desde su creación,
- Porcentaje de tiempo que 4D Server ha otorgado a este proceso (radio).

Tipo de proceso

Cada proceso está identificado por un icono y un tipo. El color y la forma del icono indica el tipo de proceso:

- ✿ Servidor de aplicación
- ✿ Servidor SQL
- ✿ Servidor DB4D (motor de base de datos)
- ✿ Servidor web
- ✿ Servidor SOAP
- ✿ Proceso cliente 4D protegido (proceso desarrollo de un 4D conectado)
- ✿ Proceso cliente 4D principal (proceso principal de un 4D conectado). Proceso colaborativo, equivalente al servidor del proceso creado en el equipo cliente)
- ✿ Proceso base 4D client (proceso paralelo a un proceso 4D client. Proceso preemptive responsable de controlar el proceso cliente 4D principal correspondiente)
- ⌚ Proceso en espera (anterior o futuro "Proceso cliente 4D base de datos")
- Worker servidor SQL
- Worker servidor HTTP
- ✿ Proceso 4D client (proceso ejecutado en el 4D conectado)
- ✿ Procedimiento almacenado (proceso lanzado por un 4D conectado y ejecutado en el servidor)
- ⊕ Método web (lanzado por un 4D ACTION por ejemplo)
- ⊕ Método web (apropiativo)
- ⊕ Método SOAP (lanzado por un servicio web)
- ⊕ Método SOAP (proceso apropiativo)
- 📖 Logger
- 👁️ Conexión listener TCP
- 🕒 Manager session TCP
- 🌀 Otro proceso
- Proceso worker (cooperativo)
- ✿ Procedimiento almacenado (proceso apropiativo)
- Proceso worker (apropiativo)

Nota: un proceso cliente 4D principal y su proceso base 4D client "twinned" se agrupan cuando la opción **Mostrar los procesos por grupos** está seleccionada.

Botones de administración

La página tiene cinco botones de control que actúan en los procesos seleccionados. Note que sólo se puede actuar en los procesos usuario.

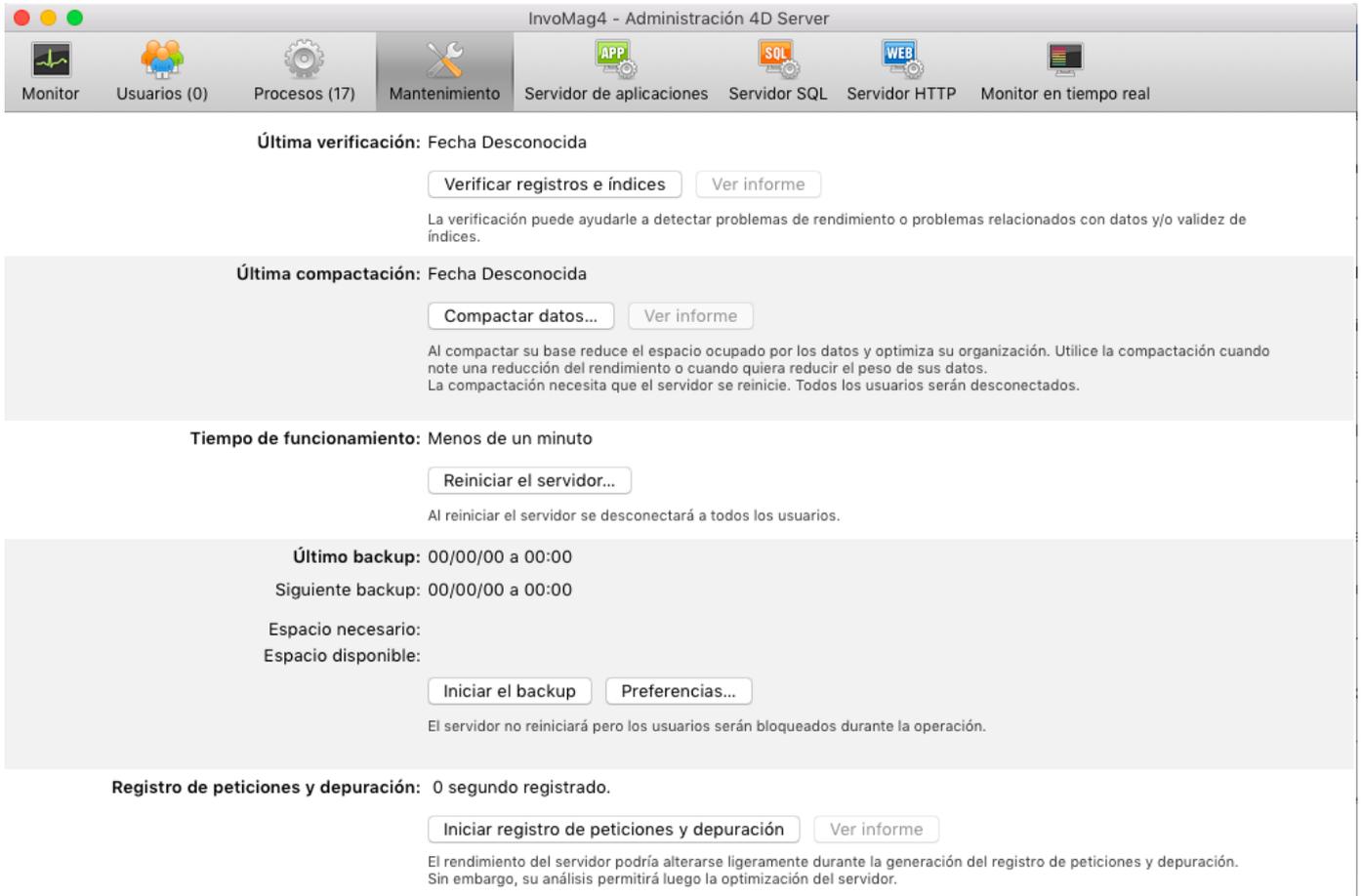


- **Abortar proceso:** permite abortar los procesos seleccionados. Al hacer clic en este botón, aparece una caja de diálogo de alerta de manera que pueda confirmar o cancelar la operación.
Nota: también puede abortar directamente los procesos seleccionados sin mostrar la caja de diálogo de confirmación presionando la tecla **Alt** (Windows) u **Opción** (Mac OS) mientras hace clic en este botón.
- **Pausar proceso:** permite poner pausa a los procesos seleccionados.

- Activar proceso: permite reactivar el o los procesos seleccionados. Los procesos deben haber sido suspendidos anteriormente (utilizando el botón Detener o por programación); de lo contrario, este botón no tiene efecto.
- Depurar proceso: permite abrir en el equipo servidor una o varias ventanas del depurador para los procesos seleccionados. Al hacer clic en este botón, aparece una caja de diálogo de manera que puede confirmar o cancelar la operación. Note que la ventana del depurador no se muestra cuando el código 4D se ejecuta en el equipo servidor (por ejemplo dentro de un trigger o de la ejecución de un método con el atributo "Execute on Server").
Nota: también puede depurar directamente un proceso, sin mostrar la caja de diálogo de confirmación, presionando la tecla **Alt** (Windows) u **Opción** (Mac OS) mientras hace clic en este botón.
- Mirar usuarios: permite mostrar en la página Usuarios todos los procesos de los usuarios seleccionados. Este botón está activo cuando al menos un proceso usuario está seleccionado.

Página Mantenimiento

La página Mantenimiento ofrece información sobre el funcionamiento actual de la base. También ofrece acceso a las funciones de mantenimiento básicas:



InvoMag4 - Administración 4D Server

Monitor Usuarios (0) Procesos (17) **Mantenimiento** Servidor de aplicaciones Servidor SQL Servidor HTTP Monitor en tiempo real

Última verificación: Fecha Desconocida

Verificar registros e índices Ver informe

La verificación puede ayudarle a detectar problemas de rendimiento o problemas relacionados con datos y/o validez de índices.

Última compactación: Fecha Desconocida

Compactar datos... Ver informe

Al compactar su base reduce el espacio ocupado por los datos y optimiza su organización. Utilice la compactación cuando note una reducción del rendimiento o cuando quiera reducir el peso de sus datos. La compactación necesita que el servidor se reinicie. Todos los usuarios serán desconectados.

Tiempo de funcionamiento: Menos de un minuto

Reiniciar el servidor...

Al reiniciar el servidor se desconectará a todos los usuarios.

Último backup: 00/00/00 a 00:00
Siguiente backup: 00/00/00 a 00:00

Espacio necesario:
Espacio disponible:

Iniciar el backup Preferencias...

El servidor no reiniciará pero los usuarios serán bloqueados durante la operación.

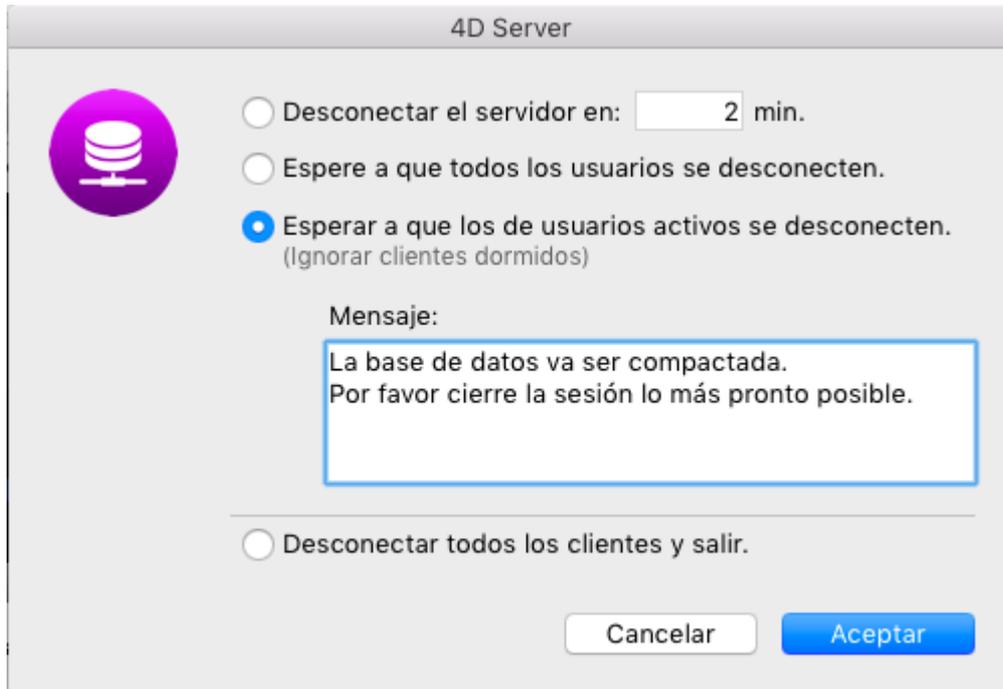
Registro de peticiones y depuración: 0 segundo registrado.

Iniciar registro de peticiones y depuración Ver informe

El rendimiento del servidor podría alterarse ligeramente durante la generación del registro de peticiones y depuración. Sin embargo, su análisis permitirá luego la optimización del servidor.

- **Última verificación:** esta área indica la fecha, la hora y el estado de la última verificación de datos efectuada en la base. Para mayor información sobre la verificación de datos, consulte el manual de Diseño. El botón **Verificar registros e índices** permite lanzar directamente la operación de verificación, sin interrupción del servidor. Note que el servidor puede ralentizarse durante la operación. Todos los registros e índices de la base se verifican. Si quiere definir la verificación o tener opciones adicionales, necesitará el Centro de seguridad y de mantenimiento (CSM). Después de la verificación, se genera un archivo de informe en formato XML y HTML en el servidor en la carpeta Logs ubicada junto al archivo de estructura de la base. El botón **Ver informe** (llamado **Descargar informe** si la operación se ha efectuados desde una máquina cliente) permite mostrar el archivo en su navegador.
- **Última compactación:** esta área indica la fecha, hora y el estado de la última compactación de datos de la base. Para mayor información sobre el procedimiento de compactación de datos, consulte el manual de Diseño. El botón **Compactar los datos...** permite lanzar directamente una operación

de compactación de los datos. Esta operación requiere detener el servidor: cuando hace clic en este botón, aparece la caja de diálogo de cierre de la base 4D Server, permitiéndole elegir el modo de interrupción de la operación:



Para mayor información sobre esta caja de diálogo, consulte la sección **Salir de 4D Server**.

Después de la interrupción de la base, 4D Server efectúa una compactación estándar de los datos de la base. Si quiere opciones adicionales, necesitará el Centro de seguridad y de mantenimiento (CSM).

Una vez termina la compactación, 4D Server reinicia automáticamente la base. Los usuarios 4D pueden entonces reconectarse.

Nota: si la petición de compactación se lleva a cabo desde un equipo 4D client remoto, esta máquina es reconectada automáticamente por 4D Server.

Un archivo de informe se genera en formatos XML y HTML en el servidor en la carpeta Logs ubicada junto al archivo de estructura de la base. El botón **Ver informe** (llamado **Descargar informe** si la operación se efectuó desde un equipo cliente) le permite mostrar el archivo en su navegador.

- **Duración de funcionamiento:** esta área indica la duración del funcionamiento del servidor desde la última vez que se inició (días, horas y minutos). El botón **Reiniciar el servidor...** permite reiniciar de inmediato el servidor. Al hacer clic en este botón, aparece la caja de diálogo de cierre de la base 4D Server de manera que puede elegir el modo de interrumpir la operación (ver la sección **Salir de 4D Server**). Después de reiniciar, 4D Server relanza automáticamente la base. Los usuarios 4D pueden entonces reconectarse. **Nota:** si la petición de reinicio se efectúa desde una máquina 4D client remota, esta máquina se reconecta automáticamente por 4D Server.
- **Último backup:** esta área indica la fecha y la hora del último backup de la base y ofrece información sobre el próximo backup automático programado (si lo hay). Los backups automáticos se configuran utilizando la página "Periodicidad" de las Preferencias de la base.
 - Próximo backup: fecha y hora del próximo backup automático.
 - Espacio necesario: espacio estimado para el backup. El tamaño actual del archivo de backup puede variar de acuerdo a los parámetros (compresión, etc.) y de acuerdo a las variaciones del archivo de datos.

- Espacio disponible: espacio disponible en el volumen de backup.

El botón **Backup de la base** permite iniciar el backup de la base de inmediato utilizando los parámetros de backup actuales (archivos guardados, ubicación de los archivos, opciones, etc.). Puede ver estos parámetros haciendo clic en el botón de **Preferencias....** Durante un backup en el servidor, las máquinas cliente se "bloquean" (pero no se desconectan) y no es posible que nuevos clientes se conecten.

- **Peticiones y depuración:** esta área indica la duración del registro del historial de peticiones y de los eventos de depuración, cuando se activan.
- - El archivo de historial de peticiones guarda diferente información sobre las peticiones recibidas por el servidor (excluyendo las peticiones web): hora, número de proceso, usuario, tamaño de la petición, duración del proceso, etc. que puede utilizarse para analizar la operación del servidor. Este archivo se llama 4DRequestsLog_X (siendo X el número secuencial del archivo) y se guarda en la carpeta Logs de la base. Una vez el archivo alcance un tamaño de 10 MB, se cierra y se genera un nuevo archivo, con un número secuencial incrementado.
 - El archivo de eventos de depuración almacena cada ejecución de un método, comando 4D o comando de plug-in en un archivo llamado "4DDebugLog.txt", ubicado automáticamente en la subcarpeta **Registros** de la base, junto al archivo de estructura. Cada evento se registra de manera sistemática en el archivo antes de su ejecución, lo que asegura su presencia en el archivo incluso si la aplicación se cierra inesperadamente. Tenga en cuenta que este archivo se borra y reescribe cada vez que se inicie la aplicación. Puede configurar este archivo con el comando **SET DATABASE PARAMETER**.

El botón **Iniciar el historial de peticiones** permite generar un nuevo archivo y activar el modo de registro de las peticiones. Como esto puede deteriorar notablemente el desempeño del servidor, está reservado para la fase de desarrollo de la aplicación.

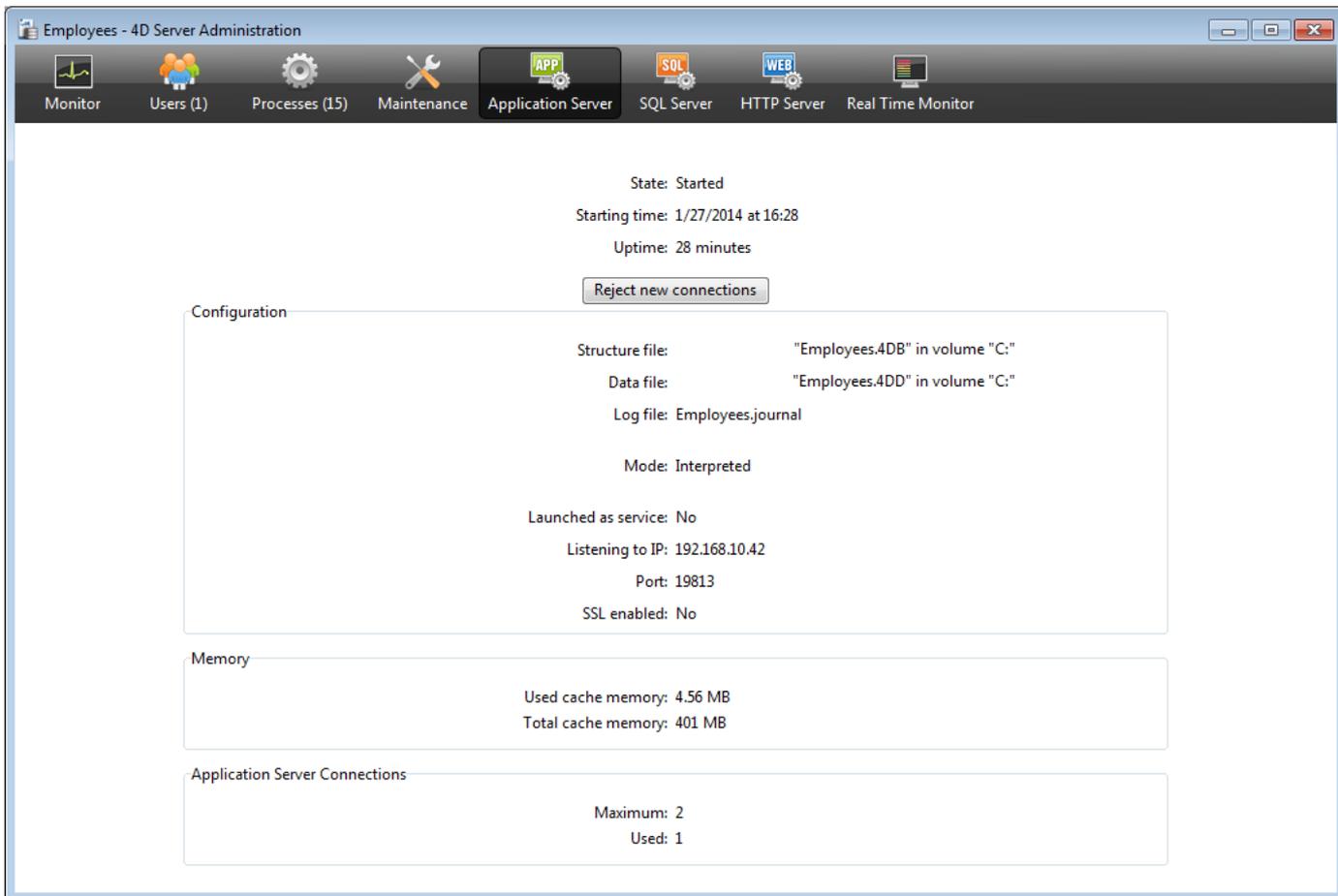
Una vez activado el registro de peticiones, el título del botón cambia a **Detener el historial de peticiones**, de manera que puede detener el registro de las peticiones en cualquier momento. Recuerde que iniciar el historial de peticiones luego de detenerlo "borra" el archivo anterior.

Nota: es posible iniciar y detener el historial de peticiones por programación vía el comando **SET DATABASE PARAMETER**.

El botón **Ver informe** (llamado **Descargar informe** si la operación se efectúa desde un equipo cliente) le permite abrir una ventana sistema mostrando el archivo de historial de las peticiones.

Página Servidor de aplicaciones

La página Servidor de aplicaciones reagrupa la información relativa a la base de datos publicada por 4D Server y permite administrar esta publicación:



The screenshot displays the 'Employees - 4D Server Administration' window. The top navigation bar includes icons for Monitor, Users (1), Processes (15), Maintenance, Application Server (selected), SQL Server, HTTP Server, and Real Time Monitor. The main content area shows the following information:

- State: Started
- Starting time: 1/27/2014 at 16:28
- Uptime: 28 minutes
-
- Configuration**
 - Structure file: "Employees.4DB" in volume "C:"
 - Data file: "Employees.4DD" in volume "C:"
 - Log file: Employees.journal
 - Mode: Interpreted
 - Launched as service: No
 - Listening to IP: 192.168.10.42
 - Port: 19813
 - SSL enabled: No
- Memory**
 - Used cache memory: 4.56 MB
 - Total cache memory: 401 MB
- Application Server Connections**
 - Maximum: 2
 - Used: 1

Información de estado

La parte superior de la página ofrece información sobre el estado actual del servidor de aplicaciones de 4D Server.

- Estado: Iniciado o Detenido.
- Hora de inicio: fecha y hora de lanzamiento de la base servidor. Esta fecha corresponde a la apertura de la base por 4D Server.
- Duración de funcionamiento: tiempo transcurrido desde la última apertura de la base.

Botón Aceptar/Rechazar nuevas conexiones

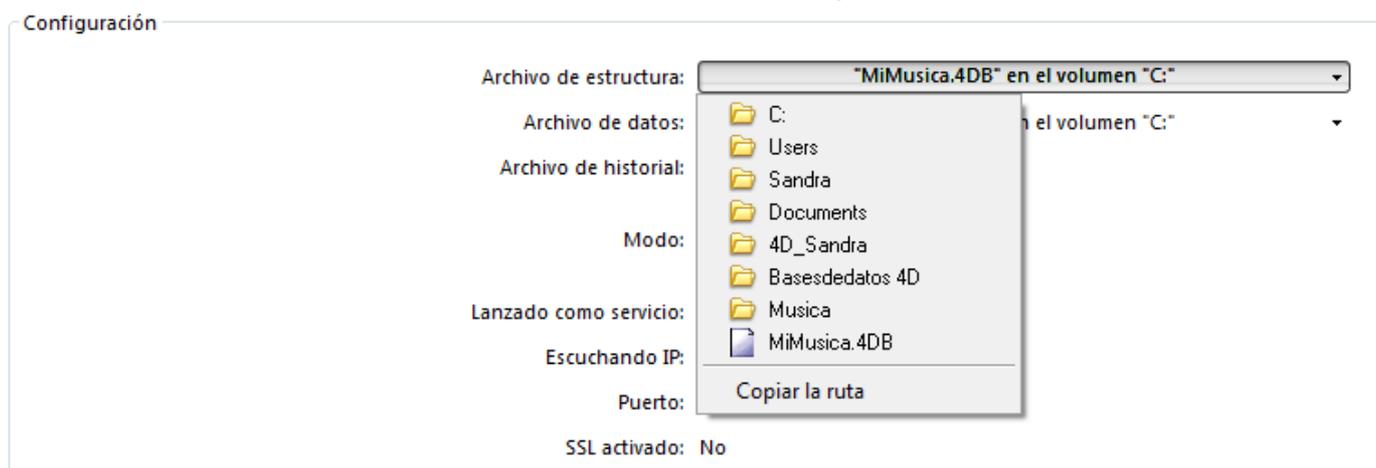
Este botón alterna entre dos estados y permite administrar el acceso de los nuevos equipos clientes a la aplicación servidor.

- Por defecto, cuando se publica la base:
 - El botón se llama "Rechazar nuevas conexiones."
 - Nuevos clientes pueden conectarse libremente (dentro de los límites de las conexiones permitidas por la licencia).
 - El nombre de la base se publica en la caja de diálogo de conexión (si la opción "Publicar el nombre de la base al inicio en el diálogo de conexión" está seleccionada en las Preferencias).
- Si hace clic en el botón **Rechazar nuevas conexiones**:
 - El título del botón cambia a "Aceptar nuevas conexiones."
 - Ningún nuevo cliente puede conectarse.
 - El nombre de la base no se muestra más en la caja de diálogo de conexión.
 - Los clientes ya no conectados no se desconectan y pueden continuar trabajando normalmente.
 - Si hace clic en el botón **Aceptar nuevas conexiones**, la base vuelve a su estado por defecto.

Esta función permite por ejemplo a un administrador llevar a cabo varias operaciones de mantenimiento (verificación, compactación, etc.) justo después de iniciar el servidor. Si el administrador utiliza una conexión cliente, puede estar seguro de ser el único que modifique los datos. También es posible utilizar esta función en preparación de una operación de mantenimiento que requiera que ningún equipo cliente esté conectado.

Configuración

Esta área ofrece información sobre la base de datos 4D publicada por el servidor: nombre y ubicación de los archivos de estructura y de datos y nombre del archivo de historial de la base. Puede hacer clic en el nombre del archivo de estructura o de datos con el fin de visualizar su ruta de acceso completa:



El campo "Modo" indica el modo de ejecución actual de la base: compilado o interpretado.

La parte inferior del área indica los parámetros de configuración del servidor (lanzado como servicio, puerto y dirección IP) y la activación del SSL para las conexiones cliente servidor (no concierne a las conexiones SQL o Web).

Memoria

Esta área indica la **memoria caché total** (parámetro definido en las preferencias de la base) y la **memoria caché utilizada** (ubicación dinámica por 4D Server en función de las necesidades).

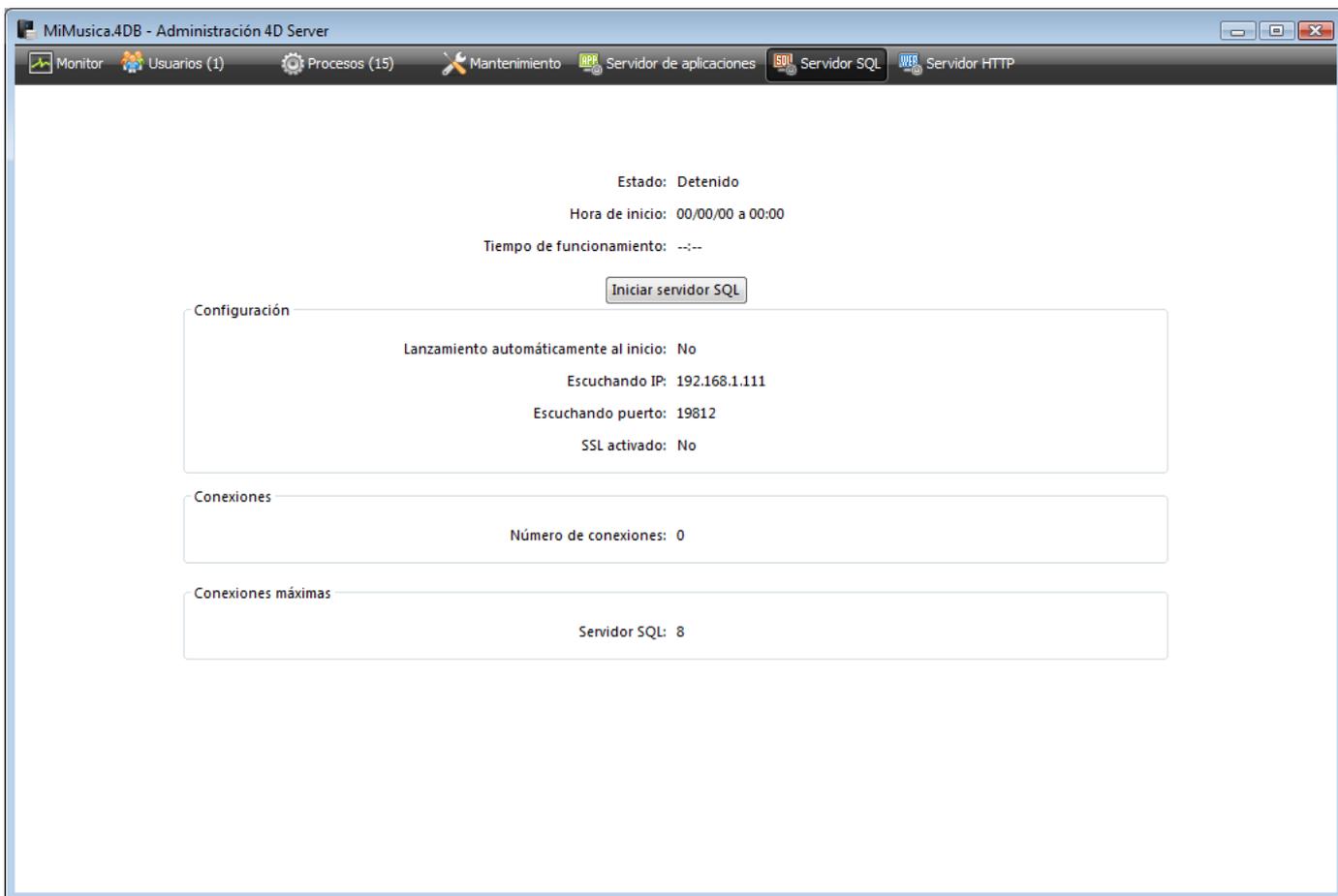
Conexiones máximas

"Máximo:" indica el número máximo de conexiones clientes simultáneas autorizadas por el servidor de la aplicación. Este valor depende de la licencia instalada en el equipo servidor.

"Used:" indica el número de conexiones que actualmente están siendo utilizadas.

Página Servidor SQL

La página Servidor SQL agrupa la información relativa al servidor SQL integrado de 4D Server. Esto también incluye un botón que permite controlar la activación del servidor:



Estado: Detenido
Hora de inicio: 00/00/00 a 00:00
Tiempo de funcionamiento: --:--

Configuración

Lanzamiento automáticamente al inicio: No
Escuchando IP: 192.168.1.111
Escuchando puerto: 19812
SSL activado: No

Conexiones

Número de conexiones: 0

Conexiones máximas

Servidor SQL: 8

Información de estado

La parte superior de la página ofrece información sobre el estado actual del servidor SQL de 4D Server.

- Estado: Iniciado o Detenido
- Hora de inicio: fecha y hora del último lanzamiento del servidor SQL. Este valor puede diferir del valor del servidor de la aplicación si el lanzamiento del servidor SQL no se efectúa "al inicio."
- Tiempo de funcionamiento: tiempo transcurrido desde el último inicio del servidor SQL.

Botón Iniciar/ Detener el servidor SQL

Este botón cambia y permite controlar la activación del servidor SQL de 4D Server.

- Cuando el estado del servidor SQL es "Iniciado," el botón se llama **Detener el servidor SQL**. Si hace clic en este botón, el servidor SQL de 4D Server se detiene de inmediato; no responde más a las peticiones SQL externas recibidas en el puerto TCP designado.
- Cuando el estado del servidor SQL es "Detenido," el botón se llama **Iniciar el servidor SQL**. Si hace clic en este botón, el servidor SQL de 4D Server se inicia de inmediato; responde a las peticiones SQL externas recibidas en el puerto TCP designado. Note que necesitará una licencia adecuada para poder utilizar el servidor SQL de 4D.

Nota: el servidor SQL también puede lanzarse automáticamente al inicio de la aplicación (opción de las Preferencias) o por programación.

Configuración

Esta área ofrece información sobre los parámetros de configuración del servidor SQL: lanzamiento automático al inicio, dirección IP de escucha, puerto TCP (19812 por defecto) y activación del SSL para las conexiones SQL (no concierne a las conexiones 4D ni Web).

Estos parámetros pueden modificarse vía las Preferencias de 4D.

Conexiones

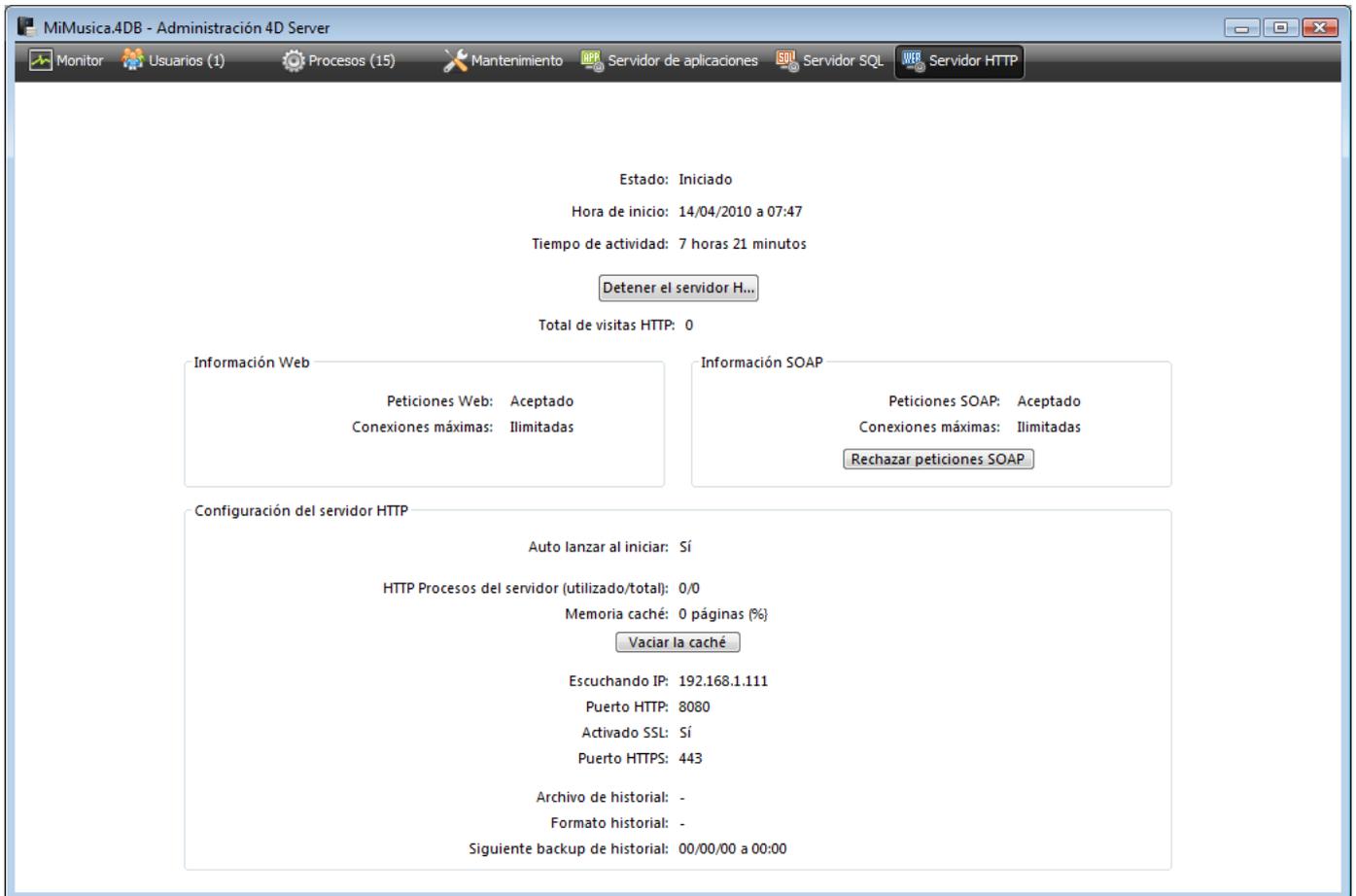
Número de conexiones SQL actualmente abiertas en 4D Server.

Conexiones máximas

Número máximo de conexiones SQL simultáneas autorizadas. Este valor depende de la licencia instalada en el equipo servidor.

Página Servidor HTTP

La página Servidor HTTP agrupa la información relativa al funcionamiento del servidor web y del servidor SOAP de 4D Server. El servidor web permite publicar contenido web tal como páginas HTML o imágenes para los navegadores web. El servidor SOAP administra la publicación de los servicios web. Estos dos servidores se apoyan en el servidor HTTP interno de 4D Server. La página contiene igualmente botones que permiten controlar la activación de los servidores:



The screenshot shows the 'Administración 4D Server' window with the 'Servidor HTTP' tab selected. The page displays the following information:

- Estado:** Iniciado
- Hora de inicio:** 14/04/2010 a 07:47
- Tiempo de actividad:** 7 horas 21 minutos
- Botón:** Detener el servidor H...
- Total de visitas HTTP:** 0
- Información Web:**
 - Peticiones Web: Aceptado
 - Conexiones máximas: Ilimitadas
- Información SOAP:**
 - Peticiones SOAP: Aceptado
 - Conexiones máximas: Ilimitadas
 - Botón:** Rechazar peticiones SOAP
- Configuración del servidor HTTP:**
 - Auto lanzar al iniciar: Sí
 - HTTP Procesos del servidor (utilizado/total): 0/0
 - Memoria caché: 0 páginas (%)
 - Botón:** Vaciar la caché
 - Escuchando IP: 192.168.1.111
 - Puerto HTTP: 8080
 - Activado SSL: Sí
 - Puerto HTTPS: 443
 - Archivo de historial: -
 - Formato historial: -
 - Siguiente backup de historial: 00/00/00 a 00:00

Información de estado

La parte superior de la página ofrece información sobre el estado actual del servidor HTTP de 4D Server.

- Estado: Iniciado o Detenido
- Hora de inicio: fecha y hora del último lanzamiento del servidor HTTP. Este valor puede diferir del valor del servidor de aplicaciones si el lanzamiento del servidor HTTP no se efectúa "al inicio."
- Tiempo de actividad: tiempo transcurrido desde el último inicio del servidor HTTP.
- Número de visitas HTTP: número de visitas HTTP (bajo nivel) recibidos por el servidor HTTP desde su inicio.

Botón Iniciar / Detener el servidor HTTP

Este botón cambia y permite controlar la activación del servidor HTTP de 4D Server.

- Cuando el estado del servidor HTTP es "Iniciado," el botón se llama **Detener el servidor HTTP**. Si hace clic en este botón, el servidor HTTP de 4D Server se detiene de inmediato; el servidor web y el servidor SOAP no aceptan más peticiones.
- Cuando el estado del servidor HTTP es "Detenido," el botón se llama **Iniciar el servidor HTTP**. Si hace clic en este botón, el servidor HTTP de 4D Server se inicia de inmediato; se aceptan las peticiones web y las peticiones SOAP (note que es posible detener por separado el servidor SOAP, ver el párrafo "Información SOAP").

Notas:

- Debe tener una licencia apropiada para poder iniciar el servidor HTTP.
- El servidor HTTP también puede lanzarse automáticamente al inicio de la aplicación (opción de las Preferencias) o por programación.

Información web

Esta área ofrece información específica sobre el servidor web de 4D Server.

- Peticiones web: Aceptadas o Rechazadas. Esta información indica si el servidor web está activo. Como el servidor web está asociado directamente al servidor HTTP, las peticiones web se aceptan cuando se inicia el servidor HTTP y se rechaza cuando se detiene.
- Conexiones máximas: número máximo de conexiones web autorizadas. Este valor depende de la licencia instalada en el equipo servidor.

Información SOAP

Esta área ofrece información específica relativa al servidor SOAP de 4D Server e incluye un botón de control.

- Peticiones SOAP: Aceptadas o Rechazadas. Esta información indica si el servidor SOAP está activo. Para que se acepten las peticiones SOAP, debe iniciarse el servidor HTTP y el servidor SOAP debe aceptar explícitamente las peticiones (ver el botón Aceptar/Rechazar).
- Conexiones máximas: número máximo de conexiones SOAP autorizadas. Este valor depende de la licencia instalada en el equipo servidor.
- Botón **Aceptar/Rechazar las peticiones SOAP**: este botón cambia y permite controlar la activación del servidor SOAP de 4D Server. Este botón modifica el valor de la opción Autorizar peticiones SOAP en la página "Servicios web/SOAP" de las Preferencias (y viceversa).
Si hace clic en el botón **Aceptar peticiones SOAP** y el servidor HTTP se detiene, 4D lo inicia automáticamente.

Configuración servidor HTTP

Esta área ofrece información sobre los parámetros de configuración y el funcionamiento del servidor HTTP:

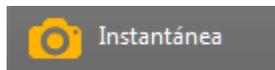
- Lanzamiento automático al iniciar: parámetro definido vía las Preferencias de 4D Server.
- Procesos servidor HTTP (utilizado/total): número de procesos HTTP creados en el servidor (número actual de procesos / total de todos los procesos creados).
- Memoria caché (utilizada/total): tamaño de la memoria caché del servidor HTTP, cuando se activa (tamaño ocupado realmente por la caché / tamaño máximo teórico asignado a la caché en las Preferencias).
- Escuchando dirección IP, Puerto TCP (80 por defecto), SSL activo para las conexiones HTTP (no concierne a las conexiones 4D ni SQL) y Puerto HTTPS utilizado: parámetros de configuración actuales del servidor HTTP, especificados en la página Web/Configuración de las Preferencias (ver la sección en el manual Lenguaje de 4D).
- Información sobre el archivo de historial: ubicación, formato y fecha del próximo backup automático de historial del servidor HTTP (archivo logweb.txt).

En modo avanzado (**Mayús**+ Clic en la operación), muestra información adicional:

Hora de inicio	Duración (ms)	Información	
2014-07-24 12:02:13:609	18	Cargando datos	Creado en el cliente
			Detalles de la operación
			Tipo de operación: Ordenar Tabla: City
			Detalles del proceso
			Número de proceso del cliente: 5 Nombre de proceso: Query_CS_Compute_4 Usuario 4D: Diseñador Nombre de sesión : soporte Equipo: ST1
			OID de cliente: 0D05D5003F8ACE4E863B8D9F9855B6DC
			Versión de cliente: v14 R3 Final

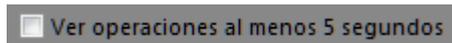
Botón Instantánea

El botón **Instantánea** le permite copiar en el portapapeles todas las operaciones que aparecen en el panel RTM, así como sus detalles relacionados (proceso e info suboperación):



Mostrar operaciones al menos 5 segundos

Si marca la opción **Mostrar operaciones al menos 5 segundos**, toda operación listada se mostrará en la página durante al menos cinco segundos, incluso después de que su ejecución haya finalizado:



Esta funcionalidad es útil para obtener información sobre las operaciones que se ejecutan muy rápidamente.

Métodos base 4D Server

-  Método base On Server Startup
-  Método base On Server Shutdown
-  Método base On Server Open Connection
-  Método base On Server Close Connection

Método base On Server Startup

Método base On Server Startup

Este comando no requiere parámetros

El **Método base On Server Startup** se llama una vez en el equipo servidor cuando abre una base con 4D Server. El **Método base On Server Startup** NO se ejecuta en un entorno diferente a 4D Server.

El **Método base On Server Startup** es la ubicación ideal para:

- Inicializar las variables interproceso utilizadas durante toda la sesión 4D Server.
- Iniciar automáticamente los **Procedimientos almacenados** al abrir la base.
- Cargar preferencias o parámetros guardados durante la sesión anterior de 4D Server.
- Evitar la apertura de la base si no se cumple una condición (ausencia de recursos sistema) para una llamada explícita a **QUIT 4D**.
- Realizar otras acciones que quiera efectuar automáticamente cada vez que se abra la base.

Para ejecutar código automáticamente en un equipo cliente cuando un 4D remoto se conecta al servidor, utilice el **Método base On Server Startup**.

Nota: el **Método base On Server Startup** se ejecuta de manera atómica, lo que significa que ningún 4D remoto puede conectarse mientras la ejecución del método no haya terminado.

Método base On Server Shutdown

Método base On Server Shutdown

Este comando no requiere parámetros

El **Método base On Server Shutdown** se llama una vez en el equipo servidor cuando la base actual se cierra en 4D Server. El **Método base On Server Shutdown** NO es llamado por otro entorno 4D diferente de 4D Server.

Para cerrar la base actual en el servidor, puede seleccionar el comando de menú **Cerrar la base...** en el servidor. También puede elegir el comando **Salir** o llamar al comando **QUIT 4D** dentro de un procedimiento almacenado ejecutado en el servidor.

Cuando se inicia el proceso de cierre de la base, 4D efectúa las siguientes acciones:

- Si no hay un **Método base On Server Shutdown**, 4D Server aborta cada proceso en ejecución uno por uno, sin distinción.
- Si existe un **Método base On Server Shutdown**, 4D Server ejecuta este método en un nuevo proceso local. Por lo tanto puede utilizar este método base para informar los otros procesos, vía la comunicación interproceso, que deben detener su ejecución. Note que 4D Server saldrá finalmente, el **Método base On Server Shutdown** puede efectuar todas las operaciones de limpieza o cierre que usted quiera, pero no puede rehusarse a salir y en algún momento terminará.

El **Método base On Server Shutdown** es el lugar ideal para:

- Detener los procedimientos almacenados lanzados automáticamente cuando se abre la base.
- Guardar (localmente, en disco) las preferencias o los parámetros a reutilizar al inicio de la sesión siguiente en el **Método base On Server Startup**.
- Efectuar cualquier otra acción que quiera activar automáticamente cada vez que salga de la base.

Importante: si utiliza el **Método base On Server Shutdown** para cerrar los procedimientos almacenados, recuerde que el servidor sale una vez se ejecuta el **Método base On Server Shutdown** (y no los procedimientos almacenados). Si los procedimientos almacenados aún están corriendo en este momento, se abortarán. Por lo tanto, si quiere asegurarse de que los procedimientos almacenados se ejecuten completamente antes de ser abortados por el servidor, el **Método base On Server Shutdown** debe indicar a los procedimientos almacenados que deben terminar su ejecución (por ejemplo, utilizando una variable interproceso) y debe permitirles cerrar (por medio de un bucle de x segundos u otra variable interproceso).

Si quiere que el código se ejecute automáticamente en un equipo cliente cuando un 4D remoto deja de conectarse al servidor, utilice el **Semaphore**.

⚙️ Método base On Server Open Connection

\$1, \$2, \$3 -> Método base On Server Open Connection -> \$0

Parámetro	Tipo	Descripción
\$1	Entero largo	← Número de usuario utilizado internamente por 4D Server para identificar los usuarios
\$2	Entero largo	← Número de conexión utilizado internamente por 4D Server para identificar una conexión
\$3	Entero largo	← Obsoleto: devuelve siempre 0 (pero debe declararse)
\$0	Entero largo	➡ 0 o se omite = conexión aceptada; otro valor = conexión rechazada

¿Cuándo se llama el método base On Server Open Connection?

El **Método base On Server Open Connection** se llama una vez en el equipo servidor cada vez que un equipo 4D remoto inicia un proceso de conexión. El **Método base On Server Open Connection** NO se invoca por otro entorno 4D diferente de 4D Server.

El **Método base On Server Open Connection** se llama cada vez que:

- un 4D remoto se conecta (inicio del proceso principal)
- un 4D remoto abre el entorno Diseño (inicio del proceso de Diseño)
- un 4D remoto inicia un proceso global, (cuyo nombre o comienza por "\$") lo cual necesita de la creación de un proceso cooperativo en el servidor (*). Este proceso puede crearse utilizando el comando **New process**, un comando de menú o la caja de diálogo "Ejecutar un método".

En cada caso con un 4D remoto, se inician tres procesos. Uno en la máquina cliente y otros dos en el equipo servidor. En la máquina cliente, el proceso ejecuta el código y envía las peticiones a 4D Server. En el equipo servidor, el **proceso 4D Client** mantiene el entorno de la base de datos del proceso cliente (las selecciones actuales y el bloqueo de registros para el proceso usuario) y responde a las peticiones enviadas por el proceso ejecutado en la máquina cliente. El **proceso base 4D Client** está a cargo de controlar el proceso 4D Client correspondiente.

(*) A partir de 4D v13, por razones de optimización los procesos servidores (proceso apropiativo para los accesos al motor de la base y proceso cooperativo para el acceso al lenguaje) sólo se crean durante la ejecución del código del lado del cliente. Por ejemplo, estos son los detalles de una secuencia de código 4D que se ejecuta en un nuevo proceso cliente:

```
// el proceso global comienza sin un nuevo proceso en el servidor, como un proceso local.
```

```
CREATE RECORD([Table_1])
```

```
[Table_1]field1_1:="Hello world"
```

```
SAVE RECORD([Table_1]) // creación aquí del proceso apropiativo en el servidor
```

```
$serverTime:=Current time(*) // creación aquí del proceso cooperativo en el servidor  
// llamada de On Server Open Connection
```

Importante: las conexiones web y las conexiones SQL no provocan la ejecución del **Método base On Server Open Connection**. Cuando un navegador web se conecta a 4D Server, se llaman el **Método de base On Web Authentication** (si lo hay) y/o la **Método base On Web Connection**.

Cuando 4D Server recibe una petición SQL, se llama el **Método base On SQL Authentication** (si existe). Para mayor información, consulte la descripción de este método base en el manual de Lenguaje 4D.

Importante: cuando se inicia un procedimiento almacenado, el **Método base On Server Open Connection** NO se llama. Los **Procedimientos almacenados** son procesos servidor y no procesos 4D Client. Ellos ejecutan el código en el equipo servidor, pero no responden a las peticiones intercambiadas por 4D client (u otros clientes) y 4D Server.

¿Cómo se llama al método base On Server Open Connection?

El **Método base On Server Open Connection** se ejecuta en el equipo servidor en el proceso 4D Client que provocó la llamada del método.

Por ejemplo, si un 4D remoto se conecta a una base 4D Server interpretada, se inicia el proceso usuario, el proceso de diseño y el proceso de registro del cliente (por defecto). El **Método base On Server Open Connection** se ejecuta tres veces seguidas. La primera vez dentro del proceso principal, la segunda vez en el proceso de inscripción del cliente y la tercera vez en el proceso de diseño. Si los tres procesos son respectivamente el sexto, séptimo y octavo proceso a iniciar en el equipo servidor, y si llama **Current process** desde el **Método base On Server Open Connection**, la primera vez **Current process** devuelve 6, la segunda vez 7 y la tercera 8.

Note que el **Método base On Server Open Connection** se ejecuta en el equipo servidor, al interior del proceso 4D Client en el servidor, independiente del proceso ejecutado en el cliente. Adicionalmente, en el momento en que se invoca el método, el proceso 4D Client no se ha nombrado aún (**PROCESS PROPERTIES** no devolverá en este momento el nombre del proceso 4D Client).

El **Método base On Server Open Connection** no tiene acceso a la tabla de las variables proceso del proceso ejecutado en el cliente. Esta tabla reside en el equipo cliente, no en el equipo servidor.

Cuando el **Método base On Server Open Connection** accede a una variable proceso, trabaja con una tabla de variables proceso particular, creada dinámicamente por el proceso 4D Client.

4D Server pasa tres parámetros de tipo Entero largo al **Método base On Server Open Connection** y espera un resultado Entero largo. El método debe por lo tanto ser declarado explícitamente con tres parámetros Entero largo así como también con un resultado de función Entero largo:

```
C_LONGINT($0;$1;$2;$3)
```

Si no devuelve un valor en \$0, por consiguiente deja la variable indefinida o inicializada en cero, 4D Server estima que el método base acepta la conexión. Si no acepta la conexión, devuelve un valor no nulo en \$0.

Esta tabla detalla la información ofrecida por los tres parámetros pasados en el método base:

Parámetro	Descripción
\$1	Número de usuario utilizado internamente por 4D Server para identificar los usuarios
\$2	Número de conexión utilizado internamente por 4D Server para identificar una conexión
\$3	Obsoleto: siempre devuelve 0 pero debe declararse

Estos números de referencia no son utilizables directamente como fuentes de información a pasar, por ejemplo, como parámetros a un comando 4D. Sin embargo, ofrecen una manera única de identificar un proceso 4D Client entre el **Método base On Server Open Connection** y el **Método base On Server Close Connection**. La combinación de estos valores es única en cualquier momento de una sesión 4D Server. Al guardar esta información en una tabla o en un array interproceso, los dos métodos base pueden intercambiar información. En el ejemplo al final de esta sección, los dos métodos base utilizan esta información para almacenar la fecha y hora de inicio y fin de una conexión en el mismo registro de una tabla.

Ejemplo 1

El siguiente ejemplo muestra cómo mantener un historial de las conexiones a la base de datos utilizando el **Método base On Server Open Connection** y utilizando el **Método base On Server Close Connection**. La tabla *[Server Log]* (mostrada a continuación) se utiliza para hacer seguimiento a los procesos de conexión:

Historial_Servidor	
ID	2 ³²
Reg_Fecha	17
Reg_Hora	🕒
Fecha_Salida	17
Hora_Salida	🕒
ID_Usuario	2 ³²
ID_Conexion	2 ³²
ID_Proceso	2 ³²
Nombre_Proceso	A

La información almacenada en esta tabla es administrada por el **Método base On Server Open Connection** y el **Método base On Server Close Connection** listado a continuación:

` Método base On Server Open Connection

C_LONGINT(\$0;\$1;\$2;\$3)

` Crear un registro [Server Log]

CREATE RECORD([Server Log])

[Server Log]Log ID:=**Sequence number**([Server Log])

` Guardar el historial Fecha y Hora

[Server Log]Log Date:=**Current date**

[Server Log]Log Time:=**Current time**

` Guarda la información de conexión

[Server Log]User ID:=\$1

[Server Log]Connection ID:=\$2

SAVE RECORD([Server Log])

‣ No devuelve error de manera que la conexión puede continuar

\$0:=0

‣ Método base On Server Close Connection

C_LONGINT(\$1;\$2;\$3)

‣ Recuperar el registro [Server Log]

QUERY([Server Log];[Server Log]User ID=\$1;*)

QUERY([Server Log];&[Server Log]Connection ID=\$2;*)

QUERY([Server Log];&[Server Log]Process ID=0)

‣ Guardar fecha y hora de desconexión

[Server Log]Exit Date:=**Current date**

[Server Log]Exit Time:=**Current time**

‣ Guardar información proceso

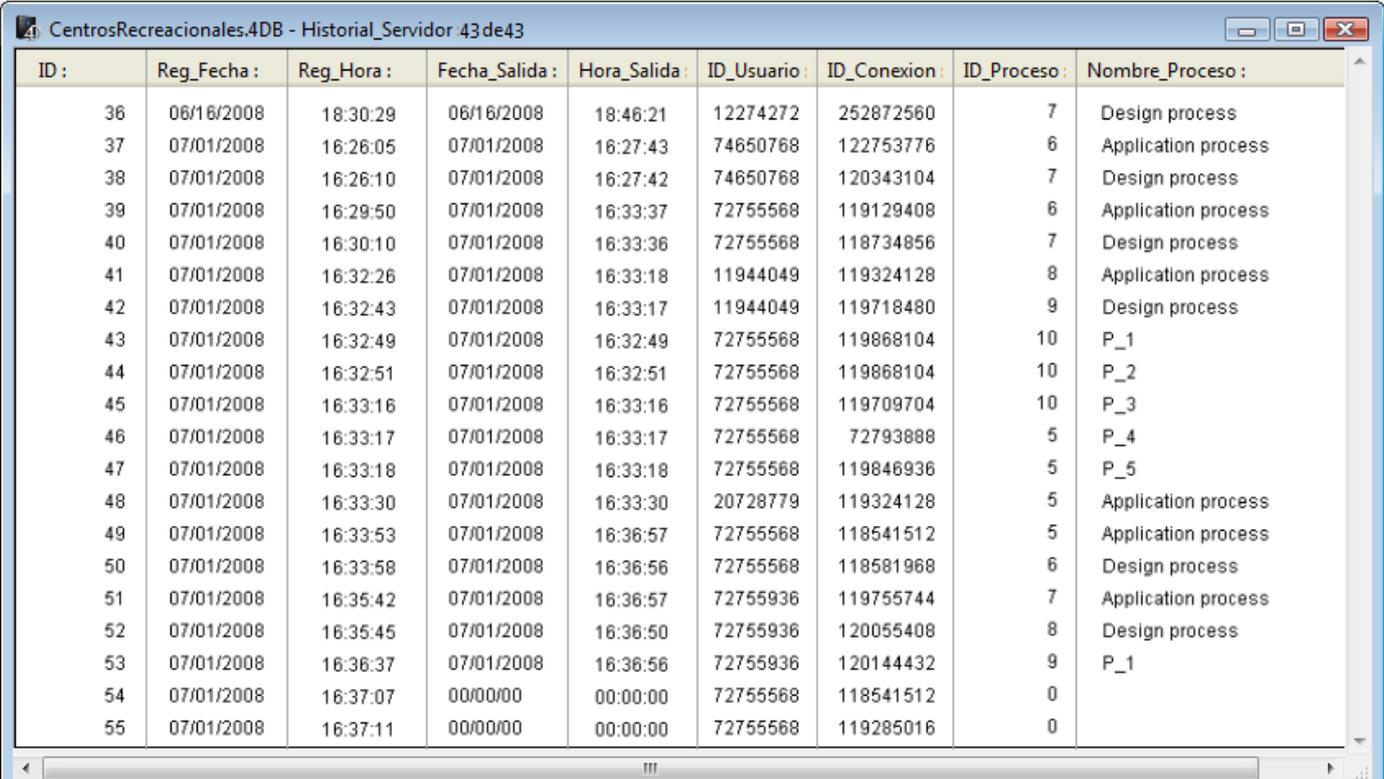
[Server Log]Process ID:=**Current process**

PROCESS PROPERTIES([Server Log]Process ID;\$vsProcName;\$vlProcState;\$vlProcTime)

[Server Log]Process Name:=\$vsProcName

SAVE RECORD([Server Log])

Estas son algunas entradas en [Server Log] mostrando varias conexiones remotas:



ID :	Reg_Fecha :	Reg_Hora :	Fecha_Salida :	Hora_Salida :	ID_Usuario :	ID_Conexion :	ID_Proceso :	Nombre_Proceso :
36	06/16/2008	18:30:29	06/16/2008	18:46:21	12274272	252872560	7	Design process
37	07/01/2008	16:26:05	07/01/2008	16:27:43	74650768	122753776	6	Application process
38	07/01/2008	16:26:10	07/01/2008	16:27:42	74650768	120343104	7	Design process
39	07/01/2008	16:29:50	07/01/2008	16:33:37	72755568	119129408	6	Application process
40	07/01/2008	16:30:10	07/01/2008	16:33:36	72755568	118734856	7	Design process
41	07/01/2008	16:32:26	07/01/2008	16:33:18	11944049	119324128	8	Application process
42	07/01/2008	16:32:43	07/01/2008	16:33:17	11944049	119718480	9	Design process
43	07/01/2008	16:32:49	07/01/2008	16:32:49	72755568	119868104	10	P_1
44	07/01/2008	16:32:51	07/01/2008	16:32:51	72755568	119868104	10	P_2
45	07/01/2008	16:33:16	07/01/2008	16:33:16	72755568	119709704	10	P_3
46	07/01/2008	16:33:17	07/01/2008	16:33:17	72755568	72793888	5	P_4
47	07/01/2008	16:33:18	07/01/2008	16:33:18	72755568	119846936	5	P_5
48	07/01/2008	16:33:30	07/01/2008	16:33:30	20728779	119324128	5	Application process
49	07/01/2008	16:33:53	07/01/2008	16:36:57	72755568	118541512	5	Application process
50	07/01/2008	16:33:58	07/01/2008	16:36:56	72755568	118581968	6	Design process
51	07/01/2008	16:35:42	07/01/2008	16:36:57	72755936	119755744	7	Application process
52	07/01/2008	16:35:45	07/01/2008	16:36:50	72755936	120055408	8	Design process
53	07/01/2008	16:36:37	07/01/2008	16:36:56	72755936	120144432	9	P_1
54	07/01/2008	16:37:07	00/00/00	00:00:00	72755568	118541512	0	
55	07/01/2008	16:37:11	00/00/00	00:00:00	72755568	119285016	0	

Ejemplo 2

El siguiente ejemplo evita una nueva conexión entre las 2 y 4 A.M.

‣ Método base On Server Open Connection

C_LONGINT(\$0;\$1;\$2;\$3)

```
If((?02:00:00?<=Current time)&(Current time<?04:00:00?))
```

```
    $0:=22000
```

```
Else
```

```
    $0:=0
```

```
End if
```

⚙️ Método base On Server Close Connection

\$1, \$2, \$3 -> Método base On Server Close Connection

Parámetro	Tipo	Descripción
\$1	Entero largo	← Número de usuario utilizado internamente por 4D Server para identificar los usuarios
\$2	Entero largo	← Número de conexión utilizada internamente por 4D Server para identificar una conexión
\$3	Entero largo	← Obsoleto: devuelve siempre 0 pero debe declararse

Descripción

El **Método base On Server Close Connection** se llama en el equipo servidor cada vez que termina un proceso 4D Client.

Como para el **Método base On Server Open Connection**, 4D Server pasa tres parámetros de tipo entero largo al **Método base On Server Close Connection**. Por otra parte, 4D Server no espera un resultado en retorno.

El método debe contener la declaración explícita de tres parámetros Entero largo:

```
C_LONGINT($1;$2;$3)
```

Esta tabla detalla la información ofrecida por los tres parámetros pasados al método base:

Parámetro	Descripción
\$1	Número de usuario utilizado internamente por 4D Server para identificar usuarios
\$2	Número de conexión utilizado internamente por 4D Server para identificar una conexión
\$3	Obsoleto: devuelve siempre 0 pero debe declararse

El **Método base On Server Close Connection** es el inverso exacto del **Método base On Server Open Connection**. Para mayor información y una descripción de este método base, así como para la descripción de los **procesos 4D Client**, ver la descripción de este método base.

Ejemplo

Ver el primer ejemplo para **Método base On Server Open Connection**.

Uso de un 4D Remoto

-  Conexión a una base 4D Server
-  Administración de máquinas remotas
-  Compilación de máquinas remotas

Conexión a una base 4D Server

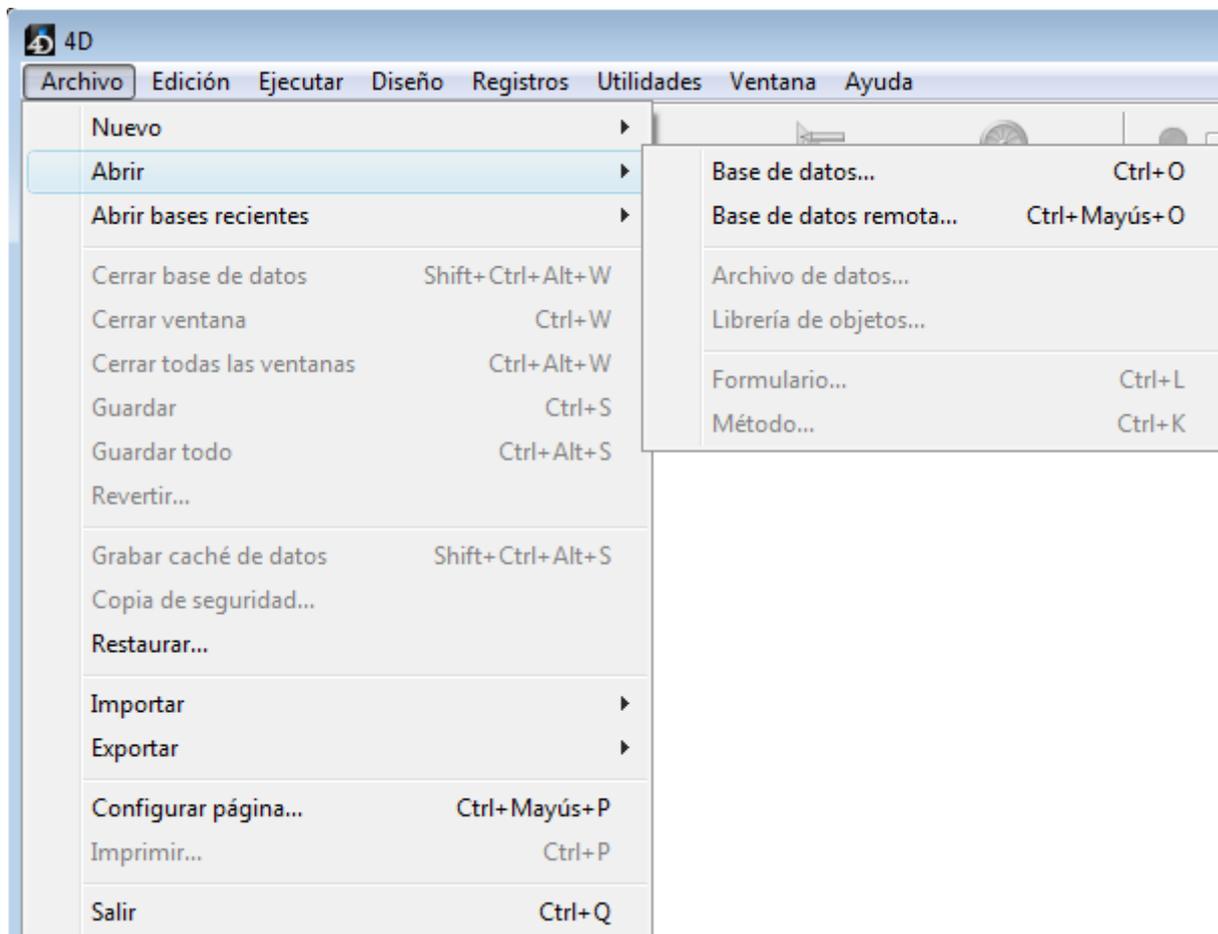
Hay tres formas para conectarse a una base 4D Server vía un 4D remoto:

- Utilizar la caja de diálogo de conexión
- Utilizar el menú **Abrir bases recientes**
- Utilizar un archivo 4DLink de atajo con los parámetros de acceso a la base.

Utilizar la caja de diálogo de conexión

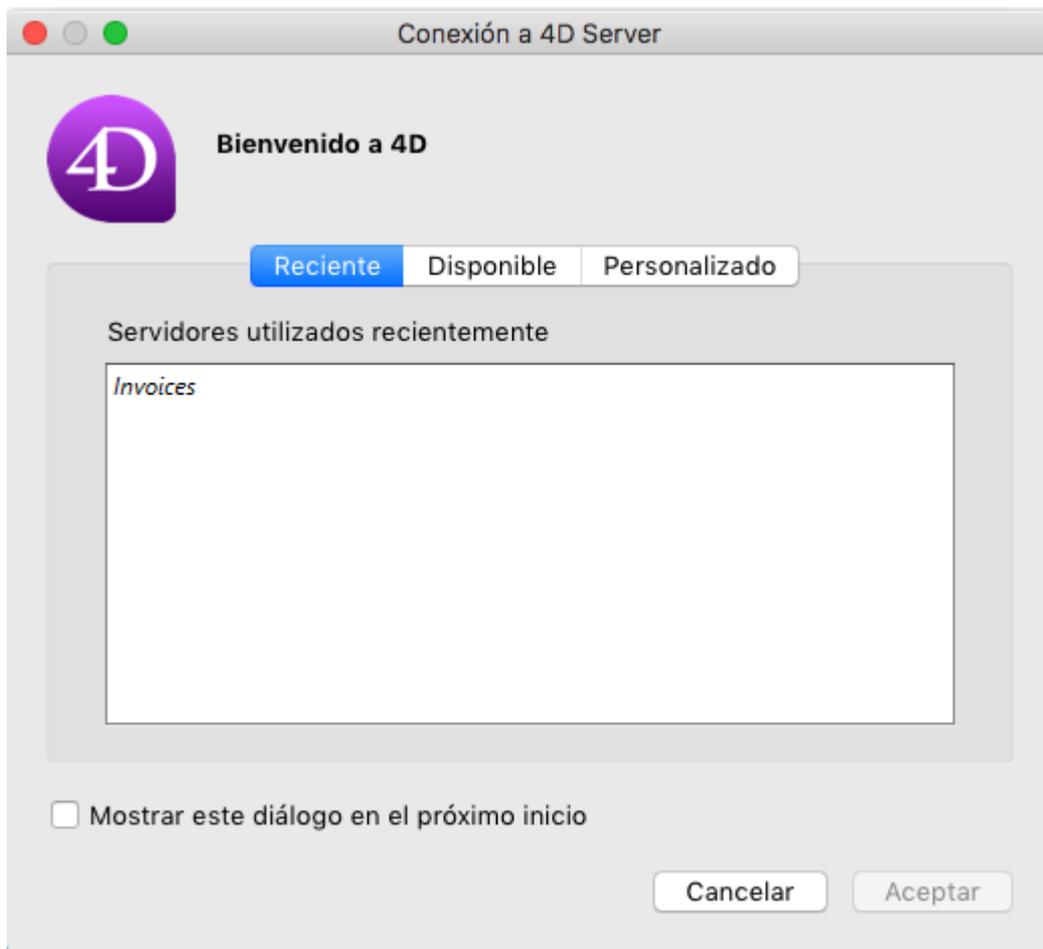
Para mostrar la caja de diálogo de conexión a 4D Server, primero lance la aplicación 4D.

El comando **Abrir** del menú **Archivo** (o el botón correspondiente en la barra de herramientas de 4D) permite seleccionar el modo de apertura de la base:



Elija el comando **Abrir>Base de datos remota...**

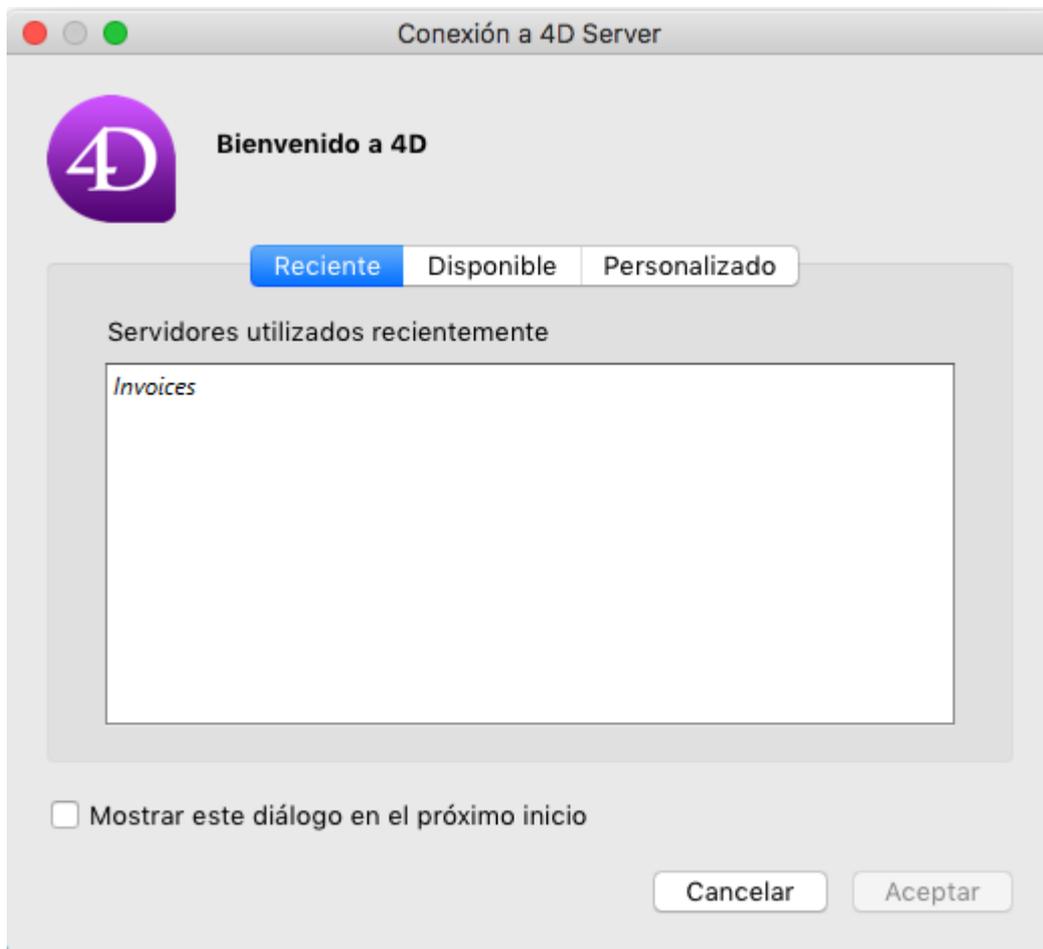
Aparece la caja de diálogo de conexión a 4D Server. Esta caja de diálogo tiene tres páginas, accesibles por medio de pestañas: Reciente, TCP/IP y Personalizado:



Si selecciona la opción **Mostrar este diálogo en el próximo inicio**, esta caja de diálogo se mostrará automáticamente al inicio de la aplicación 4D.

Nota: puede igualmente mostrar esta caja de diálogo haciendo clic en el enlace **Conectarse a 4D Server** en el diálogo de bienvenida de 4D.

Página "Reciente"



Esta página memoriza la lista de bases 4D server utilizadas recientemente. La lista está en orden alfabético. Para conectarse a un servidor desde esta lista, haga doble clic en su nombre o selecciónelo y haga clic en el botón **Aceptar**.

Página "Disponible"



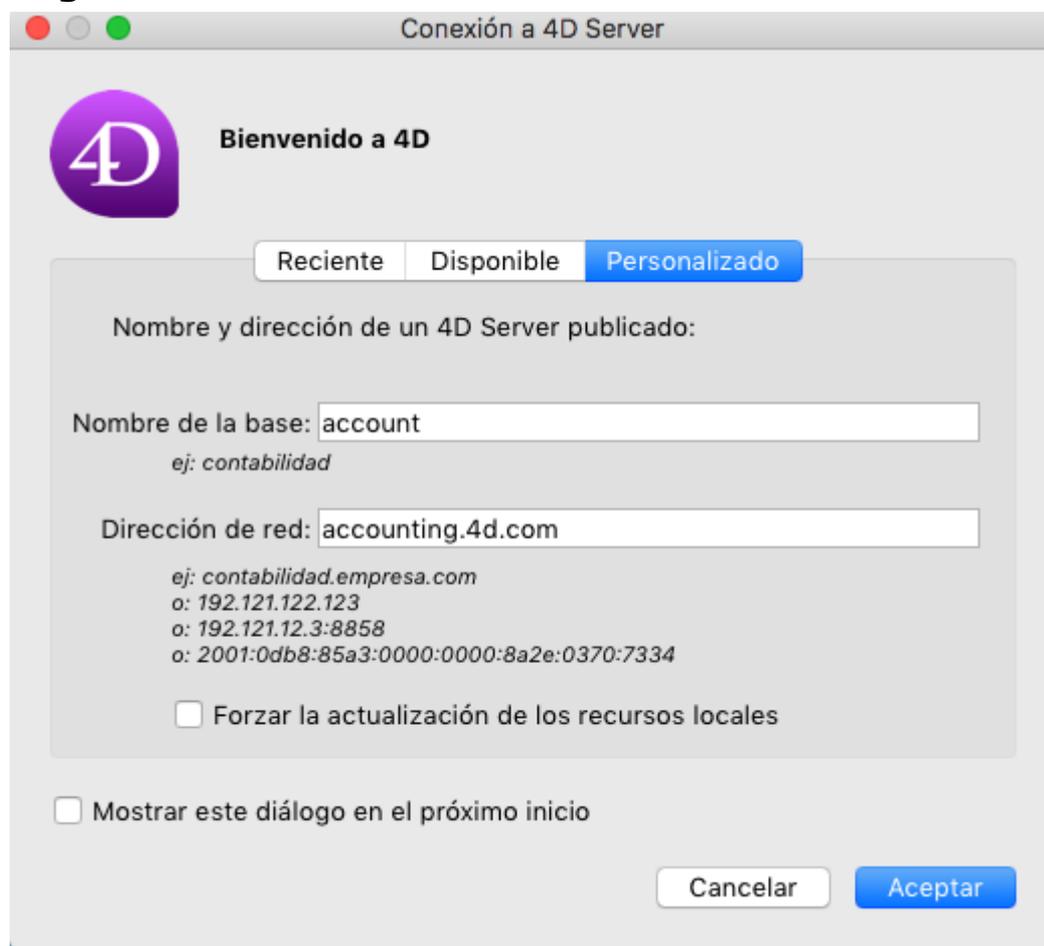
4D Server incluye un sistema integrado de publicación, que permite mostrar automáticamente el nombre de la bases 4D Server disponibles en la red. Estos nombres aparecen en la página **Disponible** de la caja de diálogo de conexión. La lista está por orden de aparición y se actualiza dinámicamente. Para conectarse a un servidor de esta lista, haga doble clic sobre el nombre o selecciónelo y haga clic en el botón **Aceptar**.

Notas:

- Un acento circunflejo (^) aparece delante del nombre de bases publicadas con la opción de cifrado. Para obtener más información, consulte la sección **Encriptar las conexiones cliente-servidor**.
- Es posible no publicar el nombre de la base en la red deseleccionando la opción **Publicar la base al inicio** en el equipo servidor. En este caso, la conexión sólo puede efectuarse manualmente en la página "Personalizado".

Nota de compatibilidad: en modo IPv4, sólo las bases publicadas en el puerto por defecto (19813) están visibles en la pestaña **Disponible**. Cuando se activa IPv6 (ver **Soporte de IPv6**), todas las bases de datos 4D Server publicadas en la red están visibles en la pestaña **Disponible**. Si no desea que aparezca una base de datos, debe anular la selección de la opción de publicación (consulte la nota anterior).

Página "Personalizado"



The screenshot shows a window titled "Conexión a 4D Server" with a purple 4D logo and the text "Bienvenido a 4D". There are three tabs: "Reciente", "Disponible", and "Personalizado" (which is selected). The main area contains the following fields and options:

- Label: "Nombre y dirección de un 4D Server publicado:"
- Field: "Nombre de la base:" with the value "account" and an example "ej: contabilidad" below it.
- Field: "Dirección de red:" with the value "accounting.4d.com" and examples "ej: contabilidad.empresa.com", "o: 192.121.122.123", "o: 192.121.12.3:8858", and "o: 2001:0db8:85a3:0000:0000:8a2e:0370:7334" below it.
- Checkbox: "Forzar la actualización de los recursos locales" (unchecked).
- Checkbox: "Mostrar este diálogo en el próximo inicio" (unchecked).
- Buttons: "Cancelar" and "Aceptar".

La página **Personalizado** permite asociar un nombre personalizado a una base 4D Server y conectarse vía la dirección de red del servidor. Como 4D Server puede configurarse con el fin de que el nombre de la base no se publique automáticamente en la red (ver la sección **Publicar la base al inicio**) y en este caso, este nombre no aparece en la página "Disponible", puede ser necesario introducir manualmente la dirección del servidor.

- **Nombre de base:** permite definir el nombre de la base 4D Server. Este nombre se utiliza en la página Recientes al referirse a la base.
- **Dirección de red:** permite introducir la dirección red de la máquina donde se lanzó 4D Server. Puede introducir el nombre del dominio (por ejemplo "mydb.mycompany.com") o la dirección IP (se soportan los formatos IPv6 e IPv4).
- Por defecto, el puerto de publicación de un 4D Server es el 19813, en cuyo caso no es necesario especifica la dirección de red. Sin embargo, especialmente cuando varios servidores se ejecutan simultáneamente en la misma máquina, este puerto puede haber cambiado en las Propiedades de la base (ver la sección **Opciones red y cliente-servidor**). En este caso, la dirección debe estar seguida por dos puntos y el número del puerto, por ejemplo:
192.168.92.104:19814 (formato IPv4) o
[2001:0db8:0000:85a3:0000:0000:ac1f:8001]:19814 (formato IPv6).

Nota: si una base fue seleccionada en las páginas **Reciente** o **Disponible** en el momento en que hizo clic en la página **Personalizado**, los dos campos muestran la información correspondiente.

Una vez que esta página designe un servidor, hacer clic en el botón **Aceptar** le permitirá conectarse al servidor. El servidor se mostrará en la página **Recientes**.

Nota: si la base se publica utilizando la opción de cifrado, debe agregar un acento circunflejo (^) delante del nombre, de lo contrario la conexión será rechazada. Para obtener más información, consulte la sección **Encriptar las conexiones cliente-servidor**.

Forzar la actualización de los recursos locales

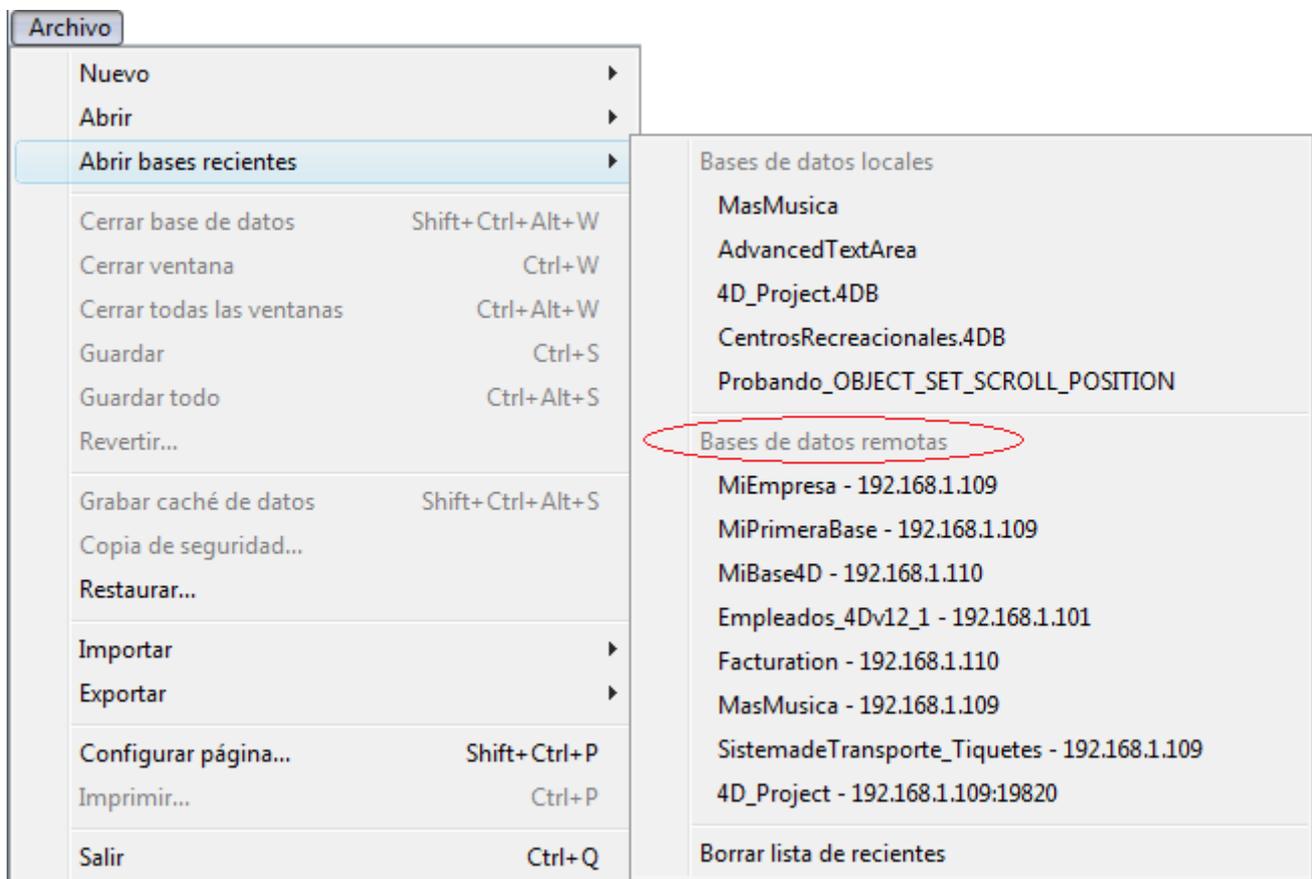
Esta opción causa la actualización sistemática de los recursos locales en el equipo cliente al momento de la conexión. Los recursos locales son la información estructural relacionada con la base que está almacenada en cada máquina cliente.

Por regla general, la actualización de los recursos locales es automática en el equipo remoto cada vez que se conecta, cuando la estructura de la base se ha modificado entre dos conexiones. La mayoría de las veces, esta opción no es necesaria. No obstante, en determinados casos, puede ser necesaria para forzar la actualización.

Utilizar el menú Abrir bases recientes

El elemento de menú **Abrir bases recientes** permite conectarse directamente a una base 4D Server a la cual se ha conectado anteriormente.

Este comando se encuentra en el menú **Archivo** de 4D. Si utiliza la aplicación 4D para abrir las bases locales y para conectarse a las bases remotas, este menú lista los dos tipos de bases. Las bases remotas se ubican en la parte inferior del menú:



El comando **Borrar lista de recientes** permite reinicializar el menú.

Usar un archivo 4DLink

Puede generar archivos de acceso a las bases que contienen parámetros destinados a automatizar y simplificar la apertura o la conexión a las bases 4D. Por lo general, un archivo de acceso puede guardar la dirección de una base 4D Server remota así como también los identificadores de conexión, evitando así varias operaciones a los usuarios.

Los archivos de acceso pueden igualmente utilizarse para la apertura de bases locales.

Uso de los archivos

Un archivo de acceso .4DLink permite lanzar la aplicación 4D y abrir la base 4D Server objetivo. Hay dos formas de usarlo:

- Vía un doble clic o un arrastrar y soltar en la aplicación 4D,
- Via el submenú **Abrir bases recientes** (archivo ubicado en la carpeta de preferencias local).
Un mismo archivo .4DLink de tipo "base remota" puede copiarse y utilizarse en los diferentes equipos.

Nota: también es posible seleccionar un archivo 4DLink en la caja de diálogo de apertura de 4D y de 4D Server (apertura de bases locales únicamente).

Creación de archivos

Los archivos de acceso a las bases de 4D son los archivos XML que tienen la extensión ".4DLink". 4D genera y utiliza este tipo de archivo para construir el submenú de "bases recientes": un archivo .4DLink es generado automáticamente por 4D cuando una base local se abre por primera vez o cuando se conecta a un

servidor por primera vez.

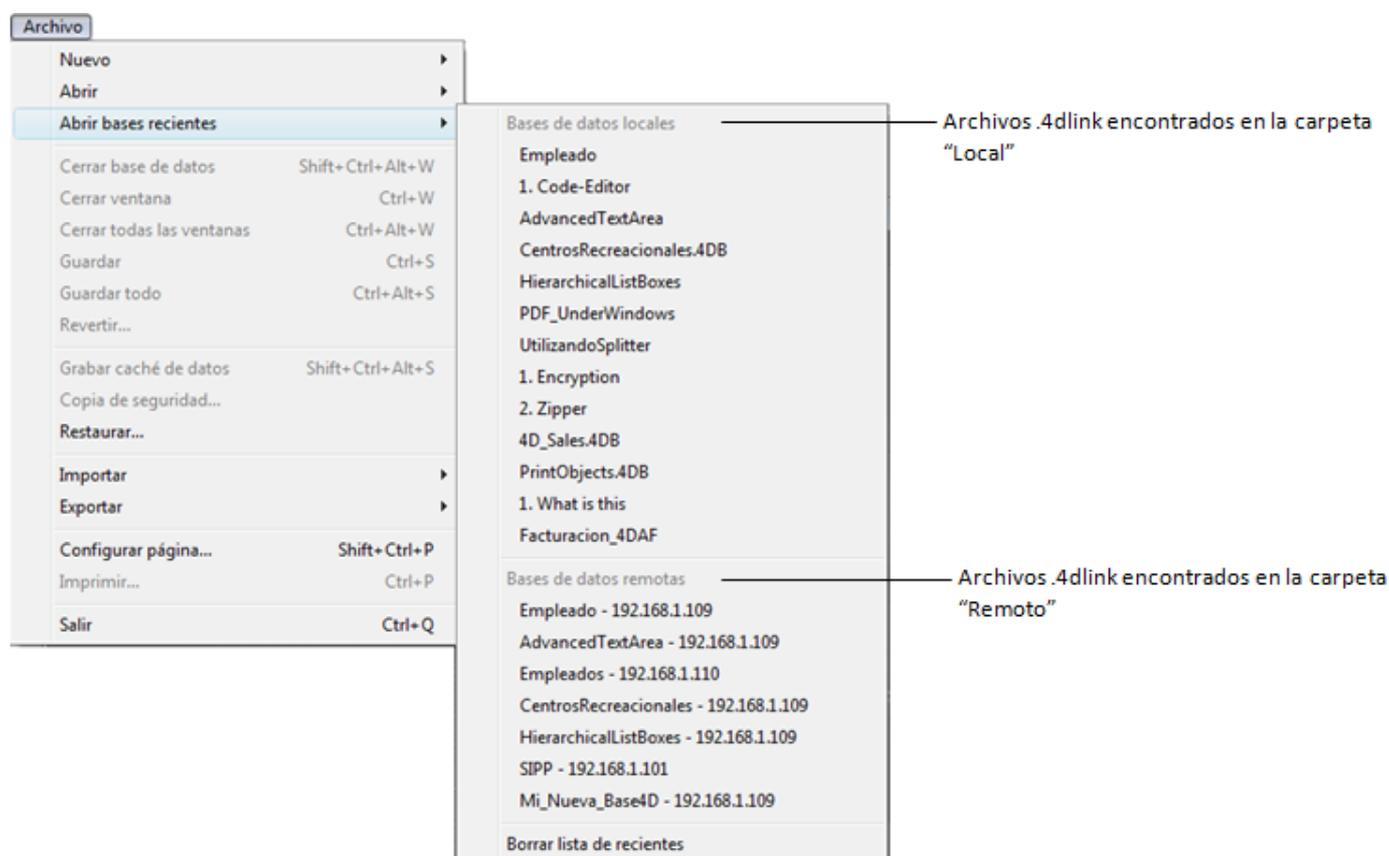
Los archivos .4DLink creados automáticamente por 4D se ponen en la carpeta de las preferencias locales del usuario. En esta carpeta, se crean dos directorios: Local y Remoto. La carpeta Local contiene los archivos ".4DLink" que se pueden utilizar para conectarse a bases locales y la carpeta Remota contiene los archivos ".4DLink" que se pueden utilizar para conectarse a bases remotas.

Las carpetas de preferencias locales se encuentran en:

- Windows 7 y posteriores:
C:\Users\NomUsuario\AppData\Roaming\4D\Favorites vXX
- OS X: *Users/NomUsuario/Library/Application Support/4D/Favorites vXX/*

... donde *XX* representa el número de versión de la aplicación (por ejemplo, "Favorites v14" para 4D v14).

Los archivos que se encuentran en estos directorios son mostrados por 4D en el submenú **Abrir bases recientes** del menú **Archivo**:



Los archivos ".4DLink" también pueden crearse con un editor XML y contener información personalizada, como los identificadores de conexión (nombre de usuario y contraseña) o el modo de apertura de la base.

4D ofrece una DTD describiendo las llaves XML que se pueden utilizar para construir un archivo ".4DLink". Este DTD se denomina *database_link.dtd* y se encuentra en la subcarpeta *\Resources\DTD* de la aplicación 4D.

Administración de máquinas remotas

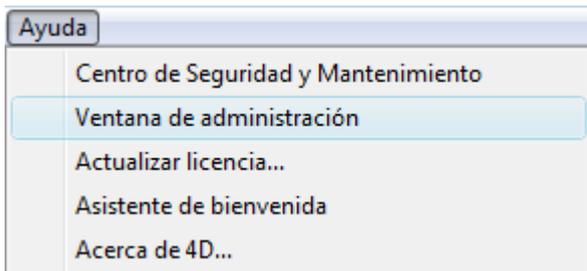
Puede administrar el equipo 4D Server desde un 4D remoto (equipo cliente) abriendo la ventana de administración de 4D Server (ver la sección **Página Monitor**) en el equipo cliente.

Abrir la ventana de administración en un equipo 4D remoto

Para abrir una ventana de administración del servidor desde un equipo cliente, debe conectarse a la base remota como Diseñador o Administrador. De lo contrario, cuando intente abrir la ventana de administración, se generará un error de privilegio (-9991).

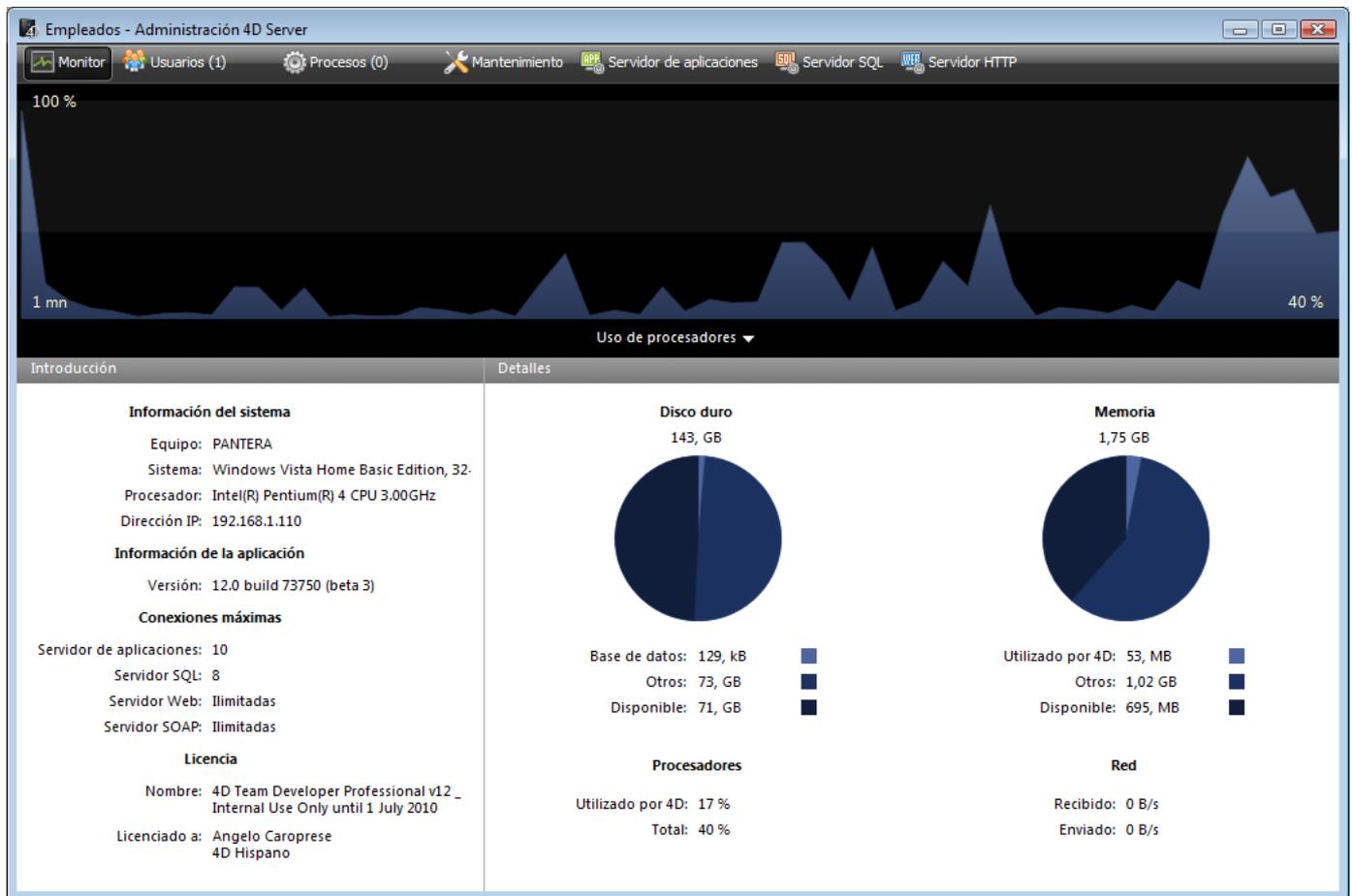
El acceso a la ventana se efectúa por una de estas dos formas:

- Seleccione el comando **Ventana de administración** del menú **Ayuda** o haga clic en el botón correspondiente en la barra de herramientas de 4D:



- Ejecute el comando **OPEN ADMINISTRATION WINDOW**.

Se muestra una ventana de administración del servidor en el equipo cliente:



Especificaciones de la administración vía un equipo cliente

Un equipo cliente que muestra la ventana de administración del servidor tiene acceso a toda la información disponible y puede actuar sobre los procesos y el inicio de los servidores. Cuando se muestra la ventana de administración del servidor en una máquina cliente, hay ciertas restricciones y particularidades del funcionamiento:

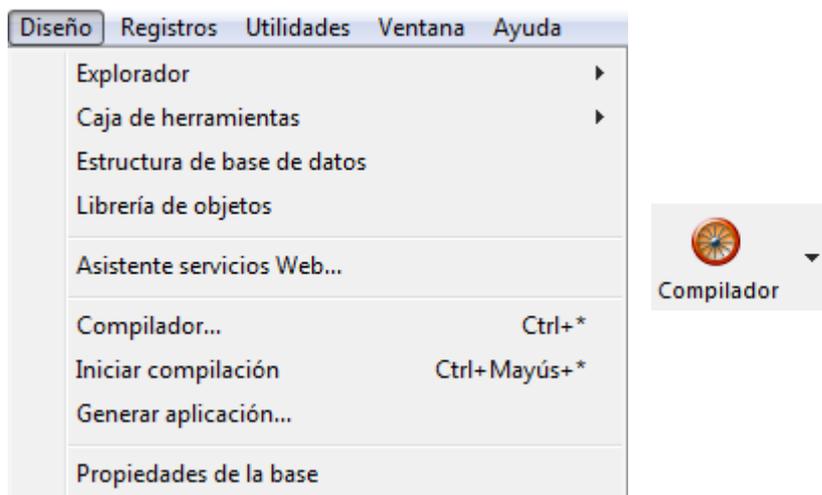
- En la **Página Proceso**, no es posible depurar un proceso de usuario (la ventana de depuración aparece en la máquina servidor).
- En la **Página Mantenimiento**, es posible ejecutar acciones que permitan que todos los clientes se desconecten y el servidor se reinicie (compactación y reinicio). En este caso, la máquina cliente que solicita la operación se vuelve a conectar automáticamente al reiniciar.
- En la **Página Mantenimiento**, los botones **Ver informe** cambian de nombre por **Descargar informe** después de la ejecución de una operación de mantenimiento. Estos archivos se descargan en la carpeta de base de datos en el equipo cliente antes de mostrarse.

Compilación de máquinas remotas

Es posible compilar una aplicación 4D desde una conexión remota. En otras palabras, es posible compilar desde un equipo cliente 4D. En las versiones de 4D Server anteriores a la v11 SQL, la compilación sólo podía llevarse a cabo desde la aplicación monousuario.

Nota: no es posible construir una aplicación 4D personalizada (monousuario o cliente/servidor) desde una conexión remota. El Generador de aplicaciones no es accesible en este entorno (el comando de menú está atenuada).

Del lado del cliente, la interfaz y los principios de compilación son los mismos que los de las versiones monousuario. La compilación puede activarse desde el menú **Diseño** o en la barra de herramientas, o desde la ventana del compilador:



Nota: la licencia "4D Team Server" se requiere del lado del 4D Server para que los equipos clientes puedan acceder a la función de compilación.

Sólo una máquina cliente puede compilar una base en un momento dado. La compilación por un cliente bloquea la función de las otras máquinas remotas. Si otro equipo cliente intenta compilar la base al mismo tiempo, aparece un mensaje de advertencia.

Mientras que una máquina cliente efectúa una compilación, las otras máquinas clientes pueden seguir trabajando y modificar los métodos o cualquier otro elemento estructural. El código compilado y el código interpretado serán entonces diferentes, lo que significa que será necesario volver a compilar la base de datos posteriormente.

El código compilado se envía en el archivo .DB en el servidor gradualmente a medida que la compilación se lleva a cabo.

Del lado del cliente, después de la finalización de la operación, es posible reiniciar el

servidor en modo interpretado o en modo compilado utilizando los comandos correspondientes en el menú Ejecutar. Cuando un equipo 4D remoto solicita el reinicio del servidor en compilado/interpretado, aparece la caja de diálogo estándar de cierre del servidor permitiéndole definir un tiempo de espera o enviar un mensaje de advertencia a los demás clientes (ver la sección **Salir de 4D Server**). Cuando el servidor se haya reiniciado, el cliente en el origen de la operación de reinicio se conecta de nuevo automáticamente.

Del lado del servidor, el reinicio en compilado/interpretado requiere del uso de la caja de diálogo estándar de apertura de archivos (menú pop up asociado al botón **Abrir**)

Nota: la compilación en la red WAN, no se recomienda por razones de rendimiento (especialmente en el caso de bases con muchos métodos), ya que la operación genera una gran cantidad de intercambios en red.

4D Server y el lenguaje 4D

-  4D Server y el lenguaje 4D
-  4D Server, conjuntos y selecciones temporales
-  Procedimientos almacenados
-  Procedimientos almacenados en los equipos clientes
-  Importación con procedimientos almacenados (ejemplo)
-  Servicios basados en los procedimientos almacenados (ejemplo)
-  Atributo Ejecutar en servidor

4D Server y el lenguaje 4D

Con 4D Server, hay tres situaciones en las que puede ejecutar código 4D en el equipo servidor:

- Triggers
- Procedimientos almacenados
- Métodos de proyecto con el atributo "Ejecutar en el servidor"
- Métodos base

Triggers

Un trigger es un método asociado a una tabla. Los triggers pueden evitar operaciones "ilegales" en los registros de su base de datos. Los triggers son una herramienta muy poderosa que permite restringir las operaciones en una tabla, como también evitar pérdidas accidentales de datos o sabotaje. Por ejemplo, en un sistema de facturación, puede evitar que añadan una factura sin especificar el nombre del cliente.

Los triggers se ejecutan en la máquina en la cual está el motor de la base de datos. Con 4D Server, los triggers se ejecutan en el contexto de los procesos que corren en el equipo servidor y no en el equipo cliente. Más precisamente, se ejecutan en el contexto de los procesos "gemelos" de los procesos usuarios que llaman la operación de la base de datos. Estos procesos gemelos comparten el contexto de la base con los procesos usuario en el equipo cliente (en particular, el estado de las transacciones y el bloqueo de los registros) pero no comparten el contexto del lenguaje (variables, procesos, conjuntos, selecciones actuales). Note sin embargo que el registro actual de la tabla del trigger es el mismo en todos los contextos.

Para mayor información sobre triggers, consulte la sección **Triggers** del manual de Lenguaje 4D.

Procedimientos almacenados

Un procedimiento almacenado 4D es un método de proyecto ejecutado en un proceso separado en el equipo servidor (o en cualquier máquina cliente), en lugar de la máquina cliente que lanzó el método. Ver la sección **Procedimientos almacenados**.

Métodos con atributo "Ejecutar en servidor"

Los métodos de proyecto que tienen el atributo "Execute on Server" también se ejecutan en el servidor. Sin embargo, a diferencia de los procedimientos almacenados, se ejecutan en los procesos "gemelos" del proceso cliente y se benefician de su contexto de base de datos. Para mayor información, consulte la sección **Atributo Ejecutar en servidor**.

Métodos base

Cuatro métodos base se ejecutan únicamente en la máquina servidor:

- **Método base On Server Startup**
- **Método base On Server Shutdown**
- **Método base On Server Open Connection**
- **Método base On Server Close Connection**

Cinco métodos base pueden ejecutarse en el equipo servidor o en un equipo cliente en función del contexto:

-
-
-
-
-

Tres métodos base pueden ejecutarse únicamente en un equipo cliente:

-
-
-

Ver las secciones correspondientes en este manual y en el manual de Lenguaje de 4D para mayor información sobre los métodos base.

4D Server y las variables

- 4D Server mantiene una tabla de variables interproceso. El alcance de estas variables es el equipo servidor. Al correr una base compilada, la definición de la tabla de las variables interproceso es común entre el servidor y todos los equipos cliente, cada máquina tiene su propia instancia.
- Como todos los procesos, cada procedimiento almacenado, método base y trigger tiene su propia tabla de variables proceso. Estas variables proceso pueden crearse y utilizarse de manera dinámica durante cada fase de la ejecución.

4D Server, conjuntos y selecciones temporales

Con 4D Server, la visibilidad de los conjuntos y de las selecciones temporales depende del origen de la creación (proceso servidor o proceso cliente) y el tipo de estos objetos (objetos locales, proceso o interproceso). Para mayor información, consulte la sección **4D Server, conjuntos y selecciones temporales**.

4D Server, conjuntos y selecciones temporales

Como se explica en las secciones y del manual Lenguaje de 4D, puede crear y utilizar conjuntos y selecciones temporales interproceso, proceso y locales:

- **Conjuntos/selecciones temporales proceso:** sólo se puede acceder a un objeto proceso por el proceso en el cual fue creado, si fue creado en un proceso cliente, por el proceso "gemelo" creado en el servidor. Los conjuntos proceso se borran tan pronto como termina la ejecución del método de proceso. Los objetos proceso no necesitan un prefijo particular en el nombre.
- **Conjuntos/selecciones temporales interproceso:** un objeto interproceso es visible para todos los procesos en el equipo (cliente o servidor) donde se creó. Un conjunto o selección temporal es un objeto interproceso si el nombre del conjunto está precedido por los símbolos (<>) — un signo "menor que" seguido por un signo "mayor que". **Nota:** esta sintaxis puede utilizarse en Windows y Macintosh. Además, en Mac OS únicamente, puede utilizar el símbolo diamante (Opción-Mayús-V).
- **Conjuntos/Selecciones temporales/clientes:** un objeto local/cliente es visible únicamente en el proceso donde fue creado. El nombre de un objeto local/cliente está precedido por el signo dólar (\$). **Nota:** aunque su nombre no comienza por \$, el conjunto sistema **UserSet** es un conjunto local/cliente.

La siguiente tabla indica los principios de visibilidad de las selecciones y conjuntos en función de donde fueron creados (la tabla es idéntica para ambos tipos de objetos):

	Proceso cliente	Otros procesos del cliente	Otros clientes	Proceso servidor	Otros procesos del servidor
Creación en un proceso cliente					
\$test	x				
test	x			X (Trigger)	
<>test	x	x			
Creación en un proceso servidor					
\$test				x	
test				x	
<>test				x	x

x = visible

Debe recordar esta matriz de visibilidad en función de las operaciones que quiera efectuar. Por ejemplo, si quiere efectuar una operación de tipo **DIFFERENCE**, **INTERSECTION** o **UNION**, asegúrese de que todos los conjuntos estén visibles en la máquina que efectúa la operación.

Por razones de optimización, se recomienda elegir la ubicación de creación y el alcance de los objetos en función de sus necesidades de visibilidad.

Procedimientos almacenados

¿Qué es un procedimiento almacenado en un sistema SQL?

La expresión "Procedimiento almacenado" proviene del mundo de los servidores SQL. Cuando una estación cliente envía una petición a un servidor SQL, envía en realidad texto en lenguaje SQL al servidor SQL. Esta petición se analiza sintácticamente (parsing) y es interpretada en el servidor SQL antes de ejecutarse. Obviamente, si el texto de la petición es importante y si la petición se envía varias veces durante una sesión, puede tomar mucho tiempo el envío del código en la red, el análisis y la interpretación de la petición. Así que la idea era encontrar una forma de enviar la petición por la red, analizada e interpretada de una vez y luego ejecutarla únicamente cada vez que se reciba de una estación cliente. La solución era conservar el código fuente de la petición (en otras palabras, un procedimiento) en el servidor y que el cliente envíe una petición con únicamente el nombre del procedimiento a ejecutar. El procedimiento es por lo tanto "guardado" en el servidor y de ahí viene el término "procedimiento almacenado."

Note que un procedimiento almacenado SQL es un procedimiento que puede recibir parámetros de una estación cliente, ejecutar las tareas para las que fue creado (de manera sincrónica o asincrónica) y posiblemente devolver un resultado al cliente. Cuando un cliente solicita la ejecución de un procedimiento almacenado, en cierta medida, delega la ejecución del código en el equipo servidor.

¿Qué es un procedimiento almacenado en 4D Server?

Aunque utilizamos el término utilizado en la industria, las funcionalidades de los procedimientos almacenados de 4D Server superan significativamente el concepto regular de procedimientos almacenados.

Con 4D en modo local, cuando utiliza un comando como **New process**, puede abrir un proceso usuario en el cual puede ejecutar un método. Este método se llama un **método proceso** (ver la sección **Métodos de proyecto** en el manual Lenguaje de 4D).

Puede hacer lo mismo con 4D Server, en un equipo cliente. Además, utilizando el comando **Execute on server** en el equipo servidor, puede iniciar un proceso de usuario en el cual ejecutar un método. Además, utilizando el comando **EXECUTE ON CLIENT**, puede correr un método en otro proceso en un cliente diferente.

En ambos casos, el método se llama un **procedimiento almacenado** y por extensión, el proceso iniciado en el equipo servidor u otro equipo cliente también llamado **procedimiento almacenado**.

Importante: la diferencia principal entre un procedimiento almacenado SQL y un procedimiento almacenado 4D Server es que en el primer caso usted ejecuta un procedimiento SQL, en el segundo caso, ejecute un proceso 4D autónomo.

Arquitectura de los procesos almacenados de 4D

Como un proceso 4D normal, un procedimiento almacenado tiene su propio entorno:

- **Una selección actual por tabla:** cada procedimiento almacenado tiene su propia selección actual. Una tabla puede tener una selección actual diferente en cada procedimiento almacenado.
- **Un registro actual por tabla:** cada tabla puede tener un registro actual diferente en cada procedimiento almacenado.
- **Variables:** cada procedimiento almacenado tiene sus propias variables proceso. Las variables proceso son reconocidas únicamente en el contexto del procedimiento almacenado al cual pertenecen.
- **Tabla por defecto:** cada procedimiento almacenado tiene su propia tabla por defecto.
- **Conjuntos proceso:** cada procedimiento almacenado tiene su propio conjunto procesos.
- **On Error Call:** cada procedimiento almacenado tiene su propio método de gestión de errores.
- **Ventana de depuración:** cada procedimiento almacenado tiene su propia ventana de depuración.

En términos de interfaz de usuario, un procedimiento almacenado puede abrir ventanas y mostrar datos (**DISPLAY RECORD**).

Un procedimiento almacenado ejecutado en un equipo cliente 4D permite la entrada de datos.

Por otra parte, un procedimiento almacenado ejecutado en el servidor no permite la entrada de datos.

Puede iniciar tantos procedimientos almacenados como lo autorice el sistema (hardware y memoria). De hecho, la máquina servidor debe considerarse como una máquina que no sólo responde a los clientes 4D y a los navegadores web, sino también es capaz de ejecutar procesos que interactúan con otros procesos que corren en el equipo servidor y en las máquinas remotas.

De la misma forma que 4D ofrece un entorno multitareas a los procesos usuario que corren en la máquina, 4D Server ofrece un entorno multitareas a los procedimientos almacenados. Por ejemplo, 4D Server mantiene una tabla de las variables interproceso que pueden ser utilizadas por los procesos almacenados para comunicarse entre ellos.

Nota: la propiedad de método "Ejecutar en servidor" permite ejecutar un método en un proceso en el servidor, pero el método utiliza un proceso "twinned" del proceso cliente, que le permite en particular beneficiarse del entorno de este proceso cliente. En este caso, no es un procedimiento almacenado 4D. Para mayor información, consulte la sección **Atributo Ejecutar en servidor**.

¿Qué puede hacer un procedimiento almacenado?

Gran parte de las funcionalidades de los procesos y comandos descritas en el manual Lenguaje 4D aplican también a los procedimientos almacenados, excepto por la entrada de datos para los procedimientos almacenados ejecutados en el servidor.

Un procedimiento almacenado puede añadir, buscar, ordenar, actualizar o borrar registros. Un procedimiento almacenado puede utilizar conjuntos y selecciones temporales, acceder a documentos en el disco, trabajar con BLOBs, imprimir registros, etc. Piense simplemente que en lugar de hacer algo en la máquina 4D local, lo hace en el equipo servidor o en varios equipos clientes.

Una ventaja evidente de los procedimientos almacenados ejecutados en el servidor es que precisamente un procedimiento almacenado se ejecuta en el equipo servidor, donde se encuentra el motor de la base de datos. Por ejemplo, un **APPLY TO**

SELECTION no es eficiente en la red, pero lo es al interior de un procedimiento almacenado. El ejemplo propuesto en la sección **Importación con procedimientos almacenados (ejemplo)** muestra la importante optimización del rendimiento que puede alcanzar implementando un procedimiento almacenado.

Los procedimientos almacenados ejecutados en uno o varios equipos clientes permiten optimizar la optimización de la repartición de tareas entre clientes y la comunicación entre varios equipos clientes. Consulte el comando **REGISTER CLIENT** en el manual Lenguaje para ver un ejemplo de procedimientos almacenados ejecutados en varios equipos clientes.

Sin embargo, la ventaja principal de la arquitectura de los procedimientos almacenados es la dimensión adicional que da a 4D Server. Utilizando procedimientos almacenados puede implementar sus propios servicios 4D Server. El único límite es su imaginación. El ejemplo en la sección **Importación con procedimientos almacenados (ejemplo)** muestra un procedimiento almacenado que ofrece información sobre 4D Server a sus clientes. Puede, por ejemplo, listar los volúmenes del equipo servidor. Este ejemplo puede expandirse fácilmente para reenviar la información sobre los directorios o documentos al cliente.

¿Qué no puede hacer un procedimiento almacenado (ejecutado en el servidor)?

De manera general, los procedimientos almacenados ejecutados en el servidor no deben efectuar operaciones que impliquen elementos de interfaz (tal como menús, ventanas, formularios...). Los mecanismos de gestión de interfaz no se manejan en el servidor.

Deben evitarse los comandos que provocan la aparición de cajas de diálogo en el equipo servidor así como también las cajas de diálogo con entrada de datos.

Esta es la lista de comandos que NO debe utilizar en procedimientos almacenados ejecutados en el servidor. Estos comandos se clasifican en tres categorías:

- **Comandos prohibidos en el servidor**

La presencia de uno de estos comandos en un procedimiento almacenado provoca la aparición de una caja de diálogo de alerta que indica que el comando no puede ejecutarse en 4D Server. Se devuelve el error #67, el cual puede interceptarse por medio de un método instalado por el comando **ON ERR CALL**.

ACCUMULATE
ADD RECORD
_o_ADD SUBRECORD
APPEND MENU ITEM
BREAK LEVEL
POST OUTSIDE CALL
CHANGE LICENSES
Count menu items
Count menus
CREATE USER FORM
DELETE MENU ITEM
DELETE USER FORM
DISABLE MENU ITEM
DISPLAY SELECTION
EDIT ACCESS
EDIT FORM

ENABLE MENU ITEM
FILTER EVENT
Get menu item
Get menu item key
Get menu item mark
Get menu item style
Get menu title
SET PICTURE TO LIBRARY
_o_GRAPH TABLE
INSERT MENU ITEM
Level
LIST USER FORMS
Menu selected
MODIFY RECORD
MODIFY SELECTION
_o_MODIFY SUBRECORD
ON EVENT CALL
_o_Open external window
PAGE BREAK
PAGE SETUP
PRINT SETTINGS
QUERY BY EXAMPLE
QR REPORT
Printing page
REMOVE PICTURE FROM LIBRARY
SET MENU ITEM
SET MENU ITEM SHORTCUT
SET MENU ITEM MARK
SET MENU ITEM STYLE
SET PICTURE TO LIBRARY
SHOW MENU BAR
Subtotal

● **Comando inapropiados en el servidor**

La utilización de estos comandos en procedimientos almacenados no es recomendable ya que su funcionamiento no se adapta a una ejecución en el servidor. Estos comandos pueden bloquear el servidor, provocar errores y no producen los efectos esperados. No se devuelve un código de error específico.

ACCEPT
Activated
_o_ADD DATA SEGMENT
After
APPEND DATA TO PASTEBOARD
APPEND TO LIST
Before
BLOB TO DOCUMENT
BLOB to list
BRING TO FRONT
_o_C_GRAPH
CANCEL
CHANGE CURRENT USER
CHANGE PASSWORD
CLEAR LIST
CLEAR PASTEBOARD

Copy list
Count list items
Count screens
Create document(1)
_o_Create resource file(1)
Current form table
Current user
Deactivated
DELETE FROM LIST
DELETE USER
DIALOG
_o_DISABLE BUTTON
DRAG AND DROP PROPERTIES
DRAG WINDOW
Drop position
_o_During
_o_ENABLE BUTTON
ERASE WINDOW
EXPORT DATA(1)
FILTER KEYSTROKE
Find window
Focus object
FONT LIST
_o_Font name
_o_Font number
Evento formulario
FORM FIRST PAGE
FORM Get current page
FORM GET PROPERTIES
FORM GOTO PAGE
FORM LAST PAGE
FORM NEXT PAGE
FORM PREVIOUS PAGE
FORM SET INPUT
FORM SET OUTPUT
Frontmost process
Frontmost window
Get edited text
GET GROUP LIST
GET GROUP PROPERTIES
GET HIGHLIGHT
GET LIST ITEM
GET LIST ITEM PROPERTIES
GET LIST PROPERTIES
GET MOUSE
GET PASTEBBOARD DATA
GET PICTURE FROM PASTEBBOARD
Get text from pasteboard
GET USER LIST
GET USER PROPERTIES
GET WINDOW RECT
Get window title
GOTO OBJECT
GRAPH SETTINGS
HIDE PROCESS

HIDE TOOL BAR
HIDE WINDOW
HIGHLIGHT RECORDS
HIGHLIGHT TEXT
IMPORT DATA(1)
In break
In footer
In header
INSERT IN LIST
_o_INVERT BACKGROUND
Is a list
Is user deleted
Keystroke
List item parent
List item position
LIST TO BLOB
Load list
MAXIMIZE WINDOW
Menu bar height
Menu bar screen
MINIMIZE WINDOW
Modified
New list
Next window
OBJECT GET COORDINATES
OBJECT MOVE
OBJECT SET LIST BY NAME
OBJECT SET COLOR
OBJECT SET ENTERABLE
OBJECT SET FILTER
OBJECT SET FORMAT
OBJECT SET RGB COLORS
OBJECT SET TITLE
OBJECT SET VISIBLE
Old
Open document(1)
Open resource file(1)
ORDER BY(2)
Outside call
Pasteboard data size
Pop up menu
POST CLICK
POST EVENT
POST KEY
QUERY BY FORMULA(2)
QUERY(2)
REDRAW
_o_REDRAW LIST
REDRAW WINDOW
REGISTER CLIENT
REJECT
SAVE LIST
SCREEN COORDINATES
SCREEN DEPTH
Screen height

Screen width
Select folder
SELECT LIST ITEMS BY POSITION
SELECT LIST ITEMS BY REFERENCE
SELECT LOG FILE
Selected list items
Self
SET CURSOR
SET FIELD TITLES
Set group properties
SET LIST ITEM
SET LIST ITEM PROPERTIES
SET LIST PROPERTIES
SET PICTURE TO PASTEBOARD
SET SCREEN DEPTH
SET TABLE TITLES
SET TEXT TO PASTEBOARD
SET TIMER
Set user properties
SET WINDOW RECT
Shift down
SHOW PROCESS
SHOW WINDOW
SORT LIST
User in group
Validate password
Window kind
WINDOW LIST
Window process

(1) Únicamente cuando el primer parámetro es una cadena vacía.

(2) Únicamente cuando la sintaxis utilizada provoca la aparición de la caja de diálogo (ej.: **ORDER BY**([Table])).

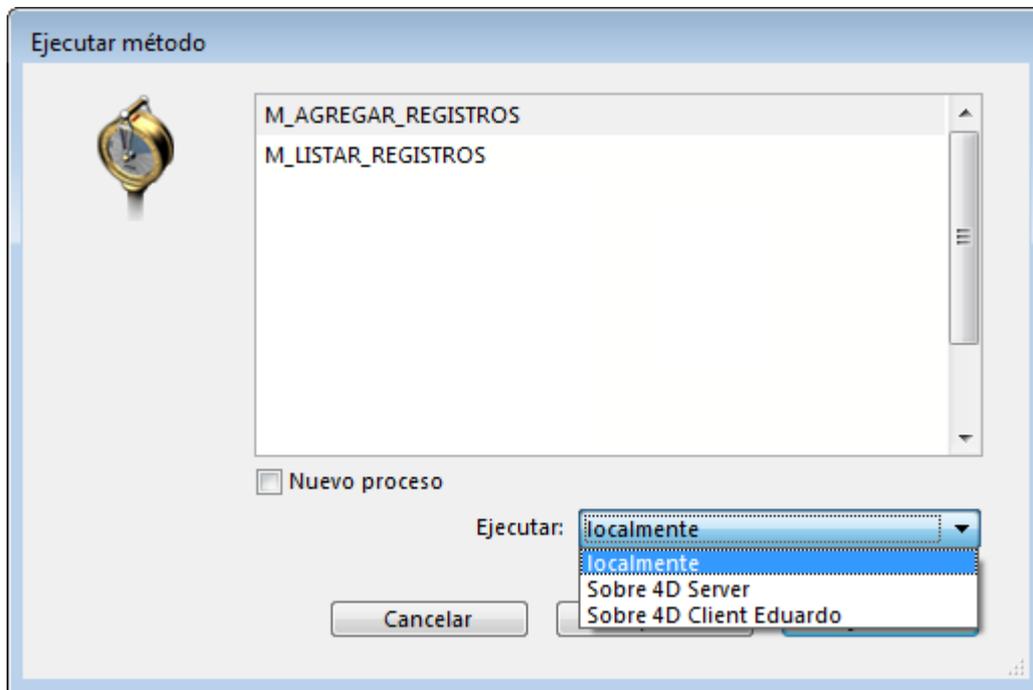
- **Comandos sin efecto en el servidor**

Los siguientes comandos no tienen efecto cuando se ejecutan en un procedimiento almacenado en el servidor. No se devuelve ningún código de error específico.

GRAPH
MESSAGES OFF
MESSAGES ON
SET MENU BAR
SHOW TOOL BAR

¿Cómo lanzar un procedimiento almacenado?

- Desde 4D, puede lanzar manualmente un procedimiento almacenado en la caja de diálogo de ejecución del método:



Puede ejecutar en 4D Server o en otro equipo 4D client. Note que para los equipos clientes 4D en esta lista, deben haber sido registrados (ver la sección **Procedimientos almacenados en los equipos clientes** y el comando **REGISTER CLIENT**).

- En 4D, puede lanzar un procedimiento almacenado por programación con la ayuda de los comandos **Execute on server** o **EXECUTE ON CLIENT**.

Nota: no es posible utilizar los comandos de gestión de procesos **DELAY PROCESS**, **PAUSE PROCESS** y **RESUME PROCESS** desde un 4D remoto con procedimientos almacenados en el servidor.

- Un método ejecutado en 4D Server (método base del servidor, método con atributo Ejecutar en servidor, trigger o procedimiento almacenado) puede lanzar un procedimiento almacenado con la ayuda de los comandos **Execute on server**, **New process** o **EXECUTE ON CLIENT**.

Comunicación interproceso entre los procedimientos almacenados y los procesos usuario

Los procedimientos almacenados pueden comunicarse entre ellos utilizando:

- variables interproceso
- semáforos globales o locales
- registros
- conjuntos interproceso y selecciones temporales interproceso
- los comandos **GET PROCESS VARIABLE**, **SET PROCESS VARIABLE** y **VARIABLE TO VARIABLE**.

Consulte las secciones correspondientes del manual Lenguaje de 4D. Nuevamente, recuerde que los comandos 4D actúan dentro del alcance del equipo que ejecuta el procedimiento almacenado (servidor o clientes) de la misma forma que en local en un equipo cliente.

Nota: los mecanismos **POST OUTSIDE CALL** y **Outside call** no tienen significado en el equipo servidor, porque los procedimientos almacenados no tienen una interfaz usuario con entrada de datos.

También hay otra funcionalidad importante: los procesos usuario de los clientes (procesos que corren en un equipo cliente) pueden leer y escribir las variables proceso (*) de un procedimiento almacenado, utilizando los comandos **GET PROCESS VARIABLE**, **SET PROCESS VARIABLE** y **VARIABLE TO VARIABLE**.

(*) Como también las variables interproceso del equipo servidor.

Importante: la comunicación proceso "Intermáquina", ofrecida por los comandos **GET PROCESS VARIABLE**, **SET PROCESS VARIABLE** y **VARIABLE TO VARIABLE**, sólo es posible del cliente al servidor. Siempre es un proceso cliente el que lee o escribe las variables de un procedimiento almacenado.

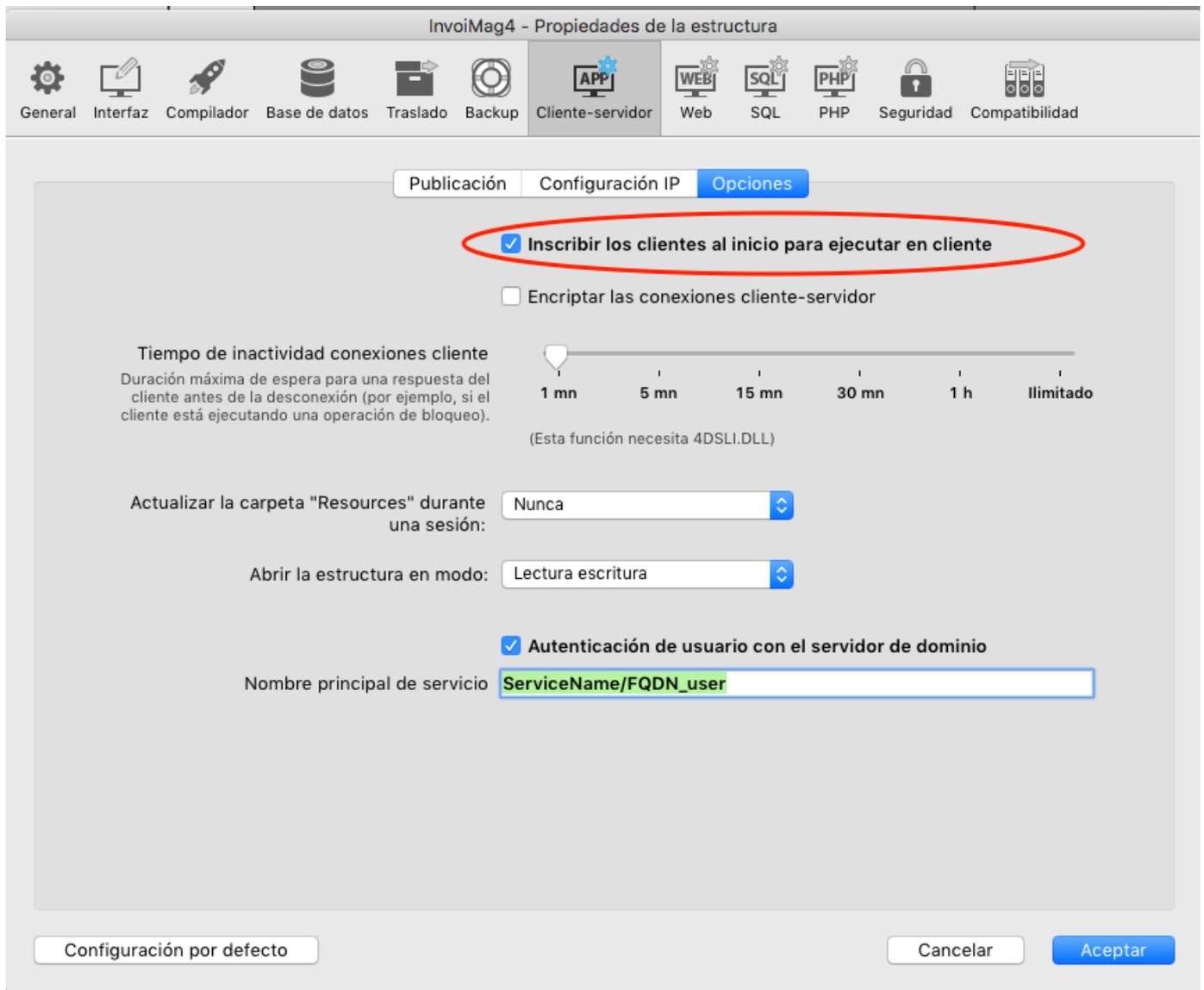
Procedimientos almacenados en los equipos clientes

Puede ejecutar los procedimientos almacenados en uno o varios clientes 4D. El funcionamiento general de los procedimientos almacenados ejecutados en los clientes es idéntico al de los procedimientos almacenados ejecutados en el servidor, excepto que en el cliente pueden invocar la entrada de datos. Para mayor información consulte la sección **Procedimientos almacenados**.

Todo equipo cliente que ejecute procedimientos almacenados por petición del servidor o de otro equipo cliente, debe estar explícitamente inscrito para la sesión. Hay dos métodos para registrar un cliente: puede registrar automáticamente al conectarse o por programación.

Registro automático de cada cliente 4D que se conecta a 4D Server

La casilla de selección "Inscribir los clientes al inicio para ejecutar en cliente" está disponible en los parámetros de la base, en la pestaña "Opciones de red" de la página "Cliente-Servidor":



Cuando esta opción está seleccionada, cada equipo cliente 4D que se conecta a la base es referenciado automáticamente con 4D Server como que puede ejecutar los procedimientos almacenados. Se crea un proceso de tipo 4D Client con el nombre de la máquina del equipo cliente en el servidor. Se crea también un proceso equivalente sobre cada equipo cliente.

Registro de 4D Client por programación

Puede registrar uno o varios equipos clientes 4D por programación. Esta opción permite seleccionar los equipos clientes que deben inscribirse y definir su nombre de registro. El tema "Proceso" contiene el comando **REGISTER CLIENT** que permite inscribir un equipo cliente con el nombre que prefiera.

Dar de baja un 4D Client

Sin importar cómo se hayan registrado los equipos cliente, puede darles de baja para la sesión actual utilizando el comando **UNREGISTER CLIENT** (Tema "Proceso") para un cliente dado. El proceso de registro (llamado de acuerdo al cliente) desaparece del grupo de procesos de usuario en el equipo servidor, así como también en el equipo cliente.

Nota: el comando **GET REGISTERED CLIENTS** permite obtener la lista y la carga de trabajo (el número de métodos pendientes por ejecutar) de los clientes

registrados en la sesión.

Para mayor información sobre estos comandos, consulte el manual de lenguaje 4D.

Importación con procedimientos almacenados (ejemplo)

El siguiente ejemplo muestra cómo la importación de datos puede acelerarse dramáticamente en entorno cliente/servidor. El método **Regular Import** permite medir cuanto tiempo toma importar los registros utilizando el comando **IMPORT TEXT**:

```
` Método de proyecto Import clásico
$vhDocRef:=Open document("")
If(OK=1)
    CLOSE DOCUMENT($vhDocRef)
    INPUT FORM([Tabla1];"Import")
    $vhStartTime:=Current time
    IMPORT TEXT([Tabla1];Document)
    $vhEndTime:=Current time
    ALERT("La operación toma "+String(0+($vhEndTime-$vhStartTime))+ " segundos.")
End if
```

Con la importación de datos clásica, 4D analiza el archivo de texto, luego para cada registro, crea un nuevo registro, llena los campos con los valores importados y envía el registro al equipo servidor para ser añadido a la base. Por consiguiente circulan numerosas peticiones por la red. Una manera de optimizar la operación es utilizar un procedimiento almacenado para efectuar la importación localmente en el equipo servidor. El equipo cliente carga el documento en un BLOB, luego inicia un procedimiento almacenado que pasa el BLOB como parámetro. El procedimiento almacenado guarda el BLOB en un documento en el disco, luego importa el documento localmente. Por lo tanto, la importación de los datos se realiza localmente (a una velocidad comparable con la de una versión local de 4D) porque se eliminan la mayoría de las peticiones que transitan por la red.

Este es el método de proyecto **CLIENT IMPORT**. Ejecutado en el equipo cliente, produce la ejecución del procedimiento almacenado **SERVER IMPORT** que se muestra a continuación:

```
` Método de proyecto CLIENT IMPORT
` CLIENT IMPORT ( Puntero ; Texto)
` CLIENT IMPORT ( -> [Tabla] ; Formulario de entrada )
```

```
C_POINTER($1)
C_TEXT($2)
C_TIME($vhDocRef)
C_BLOB($vxDatos)
C_LONGINT(spErrCode)
```

```
` Seleccione el documento a importar
```

```

$vhDocRef:=Open document("")
If(OK=1)
` Si se seleccionó un documento, no lo conserve abierto
  CLOSE DOCUMENT($vhDocRef)
  $vhStartTime:=Current time
` Trate de cargarlo en memoria
  DOCUMENT TO BLOB(Document;$vxDatos)
  If(OK=1)
` Si el documento puede cargarse en el BLOB,
` Inicie el procedimiento almacenado que importará los datos en el equipo servidor
    $spProcesoID:=Execute on server("SERVER IMPORT";32*1024;
    "Servidor Import Services";Table($1);$2;$vxDatos)
` En este punto, ya no necesitamos el BLOB en este proceso
    CLEAR VARIABLE($vxDatos)
` Espere a que termine la operación realizada por el procedimiento almacenado
    Repeat
      DELAY PROCESS(Current process;300)
      GET PROCESS VARIABLE($spProcesosID;spErrCode;spErrCode)
      If(Undefined(spErrCode))
` Nota: si el procedimiento almacenado no ha inicializado su propia instancia
` de la variable spErrCode, puede que se devuelva una variable indefinida
        spErrCode:=1
      End if
    Until(spErrCode<=0)
` Envíamos un acuse de recibo al procedimiento almacenado
    spErrCode:=1
    SET PROCESS VARIABLE($spProcesosID;spErrCode;spErrCode)
    $vhEndTime:=Current time
    ALERT("Tomó "+String(0+($vhEndTime-$vhStartTime))+ " segundos.")
  Else
    ALERT("No hay suficiente memoria para cargar el documento.")
  End if
End if

```

Este es el método de proyecto **SERVER IMPORT** ejecutado como un procedimiento almacenado:

- ` Método de proyecto SERVER IMPORT
- ` SERVER IMPORT (Entero largo ; Texto; BLOB)
- ` SERVER IMPORT (Número de Tabla ; Formulario de entrada ; Datos importados)

```

C_LONGINT($1)
C_TEXT($2)
C_BLOB($3)
C_LONGINT(spErrCode)

```

- ` La operación no ha terminado, asigne 1 a spErrCode

```

spErrCode:=1
$vpTabla:=Table($1)
INPUT FORM($vpTabla->,$2)
$vsDocName:="Archivo Import "+String(1+Random)
If(On Windows)
    $vsDocName:=$vsDocName+".txt" ` En Windows, la extensión es obligatoria
End if
DELETE DOCUMENT($vsDocName)
BLOB TO DOCUMENT($vsDocName;$3)
IMPORT TEXT($vpTabla->,$vsDocName)
DELETE DOCUMENT($vsDocName)
    ` La operación terminó, asigne 0 a spErrCode
spErrCode:=0
    ` Espere a que el equipo cliente que originó la petición haya recibido el resultado
Repeat
    DELAY PROCESS(Current process;1)
Until(spErrCode>0)

```

Nota: el método de proyecto *On Windows* se lista en la sección del manual Lenguaje de 4D.

Una vez que estos dos métodos se hayan implementado en una base, puede efectuar una importación basada en un procedimiento almacenado, escribiendo por ejemplo:

```
CLIENT IMPORT(->[Tabla1];"Import")
```

Si realiza algunas pruebas comparativas, descubrirá que utilizando este método puede importar registros hasta 60 veces más rápido que con una importación regular.

Servicios basados en los procedimientos almacenados (ejemplo)

En el ejemplo de la sección **Importación con procedimientos almacenados (ejemplo)**, un procedimiento almacenado se inicia y termina cada vez que se solicita una operación de importación de datos. En este ejemplo, un procedimiento almacenado se lanza automáticamente cuando se inicia la base del servidor y puede iniciarse y detenerse a voluntad por cualquier 4D conectado a la base. Tan pronto como se ejecuta, el procedimiento almacenado puede responder de manera asincrónica a múltiples peticiones enviadas por los clientes conectados a la base.

Mientras que la sección **Importación con procedimientos almacenados (ejemplo)**, muestra cómo optimizar un servicio 4D Server existente, este ejemplo muestra cómo implementar servicios novedosos y personalizados disponibles para todos los equipos 4D client conectados. Además, este ejemplo puede utilizarse como modelo para la creación de sus propios servicios.

Inicio automático del procedimiento almacenado

El procedimiento almacenado es lanzado automáticamente por el **Método base On Server Startup**:

```
` Método de base On Server Startup  
START SP SERVICES
```

Como el **Método base On Server Startup** lanza al método de proyecto **SP SERVICES** como un procedimiento almacenado, **SP SERVICES** comienza a correr tan pronto como la base se abre con 4D Server, bien sea que haya o no clientes conectados a la base. En la siguiente imagen, la ventana de administración de 4D Server muestra el procedimiento almacenado en ejecución cuando ningún cliente está conectado.

Nombre del proceso	Sesión	Tipo	Num	Estado	Tiempo CPU	Actividad
DB4D Flush		SQL Server	0	Running	00:00:00	0.00 %
DB4D Index builder		DB4D Server	0	Running	00:00:00	0.00 %
DB4D Server		SQL Server	0	Running	00:00:00	0.00 %
EXCLUSIVE pool worker 0		SQL Server	0	Running	00:00:00	0.00 %
EXCLUSIVE pool worker 1		SQL Server	0	Running	00:00:00	0.00 %
Shared Load Balancer		SQL Server	0	Running	00:00:00	0.00 %
SHARED pool worker 0		SQL Server	0	Running	00:00:00	0.00 %
SHARED pool worker 1		SQL Server	0	Running	00:00:00	0.00 %
SQL Net Session Manager Thread		SQL Server	0	Running	00:00:00	0.00 %
Client Manager	-	Application server	3	Waiting for flag	00:00:03	0.00 %
Internal Bridge Process	-	Application server	4	Waiting for flag	00:00:01	0.00 %
Internal Timer Process	-	Application server	2	Executing	00:00:01	0.92 %
User Interface	-	Application server	1	Waiting for event	00:00:11	1.85 %

Iniciar y detener voluntariamente el procedimiento almacenado

Este es el método de proyecto ***START SP SERVICES***:

```

` START SP SERVICES Project Method
◇vISPServices:=Execute on server("SP SERVICES";32*1024;"SP SERVICES";*)

```

Como el comando **Execute on server** actúa como **New process** al llamarse en la máquina servidor, el mismo método (***START SP SERVICES***) puede utilizarse en el equipo servidor o en una máquina cliente para iniciar a voluntad el método ***SP SERVICES*** como un procedimiento almacenado en el equipo servidor.

El método de proyecto ***STOP SP SERVICES*** "ordena" detenerse al método de proyecto ***SP SERVICES***.

```

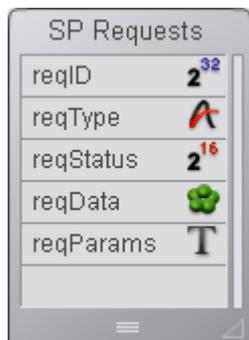
` Método de proyecto STOP SP SERVICES
SET PROCESS VARIABLE(◇vISPServices;vbStopSPServices;True)

```

Cuando se inicia el método de proyecto ***SP SERVICES***, da a la variable proceso *vbStopSPServices* el valor False y luego ejecuta un bucle hasta que esta variable Booleana se convierte en True. El comando , permite a todo proceso usuario ejecutado en el servidor o en un equipo cliente modificar el valor de la variable *vbStopSPServices*, y por lo tanto detener el procedimiento almacenado a voluntad.

Comunicación con el procedimiento almacenado

El procedimiento almacenado debe poder recibir peticiones de clientes y responder de manera asincrónica en cualquier momento y sin importar el orden. Una manera directa de asegurar esta comunicación es utilizar una tabla.



La tabla *[SP Requests]* contiene los siguientes campos:

- *[SP Requests]reqID* es asignado utilizando el comando **Sequence number**. Este campo identifica cada petición de manera única.
- *[SP Requests]reqType* describe el tipo de la petición.
- *[SP Requests]reqStatus* puede tomar uno de los siguientes valores:

Valor Descripción

- | | |
|-----|-------------------------------------------------|
| 1 | se publicó la petición pero no se ha procesado. |
| 0 | se ha procesado con éxito la petición. |
| < 0 | se procesó la petición pero ocurrió un error. |

Nota: estos valores se eligen arbitrariamente para este ejemplo, no son impuestos por 4D.

- *[SP Requests]reqData* es un BLOB que contiene los datos de la petición. Puede contener los datos enviados por el solicitante o los datos devueltos por el procedimiento almacenado al solicitante.
- *[SP Requests]reqParams* contiene eventualmente los valores de los parámetros enviados por el solicitante al procedimiento almacenado.

¿Por qué utilizar una a tabla?

La comunicación entre un proceso cliente y un procedimiento almacenado puede implementarse utilizando los comandos **GET PROCESS VARIABLE**, **SET PROCESS VARIABLE** y **VARIABLE TO VARIABLE**. Por ejemplo, esta solución utilizada en la sección **Importación con procedimientos almacenados (ejemplo)**, como también en el método de proyecto **STOP SP SERVICES** listado anteriormente.

Acá, el sistema debe permitir al procedimiento almacenado recibir y reenviar las cantidades variables de datos. Puede utilizar arrays, incluyendo arrays de texto e imagen), pero hay dos razones principales para preferir el empleo de una tabla:

- El algoritmo de gestión de peticiones vía los registros es más fácil de implementar. Enviar una petición desde un equipo cliente consiste simplemente en añadir una petición en la tabla. Responder a la petición desde el procedimiento almacenado consiste simplemente en modificar esta petición.
- Como las peticiones se almacenan en una tabla, se guardan en el disco. Por lo tanto, si el tamaño de una petición es importante, no será un problema en la medida en que puede suprimirse de la memoria (a diferencia de los datos almacenados en los arrays).

Enviar una petición desde el equipo cliente

El método de proyecto **Client post request** es un método genérico para enviar una petición:

- ` Método de proyecto Client post request
- ` Client post request (Cadena { ; Texto }) -> Entero largo
- ` Client post request (Tipo de petición { ; Parámetros }) -> Número de petición

CREATE RECORD([SP Requests])

[SP Requests]reqID:=**Sequence number**([SP Requests])

[SP Requests]reqType:=\$1

[SP Requests]reqStatus:=1

If(**Count parameters**>=2)

[SP Requests]reqParams:=\$2

End if

SAVE RECORD([SP Requests])

\$0:=[SP Requests]reqID

El método devuelve el número de la petición, cuya unicidad está garantizada por el uso del comando **Sequence number**. Una vez añadido el registro a la tabla [SP Requests], el cliente no tiene que interrogar regularmente el campo [SP Requets]redStatus hasta que el procedimiento almacenado termine de procesar la petición.

Probar el estado de la petición y recuperar el resultado en el equipo cliente

El método de proyecto **Client get result** es un método genérico para probar el estado de la petición. Como se explicó anteriormente, tan pronto como el campo [SP Requets]redStatus se vuelve diferente de 1, el cliente sabe que el procedimiento almacenado ha tratado (con éxito o no) la petición.

- ` Método de proyecto Client get result
- ` Client get result (Entero largo ; ->BLOB { ; Entero largo }) -> Entero largo
- ` Client get result (Número de petición ; ->Datos { ; Duración }) -> Código del error

C_LONGINT(\$0;\$1;\$vIDelay)

\$0: =1

\$vIDelay: =0

If(**Count parameters**>=3)

\$vIDelay: =**\$3**

End if

READ ONLY([SP Requests])

Repeat

QUERY([SP Requests];[SP Requests]reqID=**\$1**)

If(**Records in selection**([SP Requests])>0)

If([SP Requests]reqStatus&NBS#&NBS#1)

\$2->:=[SP Requests]reqData

READ WRITE([SP Requests])

```

While(Locked([SP Requests]))
    WAITING LOOP($vIDelay)
    LOAD RECORD([SP Requests])
End while
DELETE RECORD([SP Requests])
$0:=[SP Requests]reqStatus
End if
Else
    ` Se perdió el registro de la petición
    ` Esto no debería suceder. Pero de todas formas establece el código de error -2 (valor arbitrario)
    $0:=-2
End if
    ` La petición aún no ha sido procesada
    If($0=1)
        WAITING LOOP($vIDelay)
    End if
Until($0&#x0000;#&#x0000;1)
READ ONLY([SP Requests])

```

Si la petición fue administrada con éxito por el procedimiento almacenado, el método copia el resultado (si lo hay) del registro al BLOB cuyo puntero se pasó como parámetro. El método llamante luego analiza y utiliza los datos del BLOB en función del tipo de petición. Note que el cliente está encargado de borrar el registro [SP Requests] una vez termina la petición.

El pequeño método de proyecto **WAITING LOOP** hace un bucle hasta un cierto número de tics:

```

    ` Método de proyecto WAITING LOOP
    ` WAITING LOOP ( Entero largo )
    ` WAITING LOOP ( Duración en tics )
C_LONGINT($1)
$vIStartTicks:=Tickcount
Repeat
    IDLE
Until((Tickcount-$vIStartTicks)>=$1)

```

Recordatorio: **DELAY PROCESS** no tiene efecto en el proceso principal. Si utiliza el método de proyecto **WAITING LOOP**, el proceso esperará la cantidad de tiempo necesaria, aunque la petición se origine desde el proceso del entorno usuario de un equipo cliente.

El procedimiento almacenado y sus subrutinas

El método de proyecto **SP SERVICES** es el método ejecutado como procedimiento almacenado en el equipo servidor. La estructura general de este método, a continuación, es muy simple:

Inicialización de una variable “stop”

Repetir

 Búsqueda de las peticiones cuyo campo [SP Requests]reqStatus es igual a 1

 Para cada petición

 En función del tipo de petición, llamar una subrutina

 que guarde el resultado en el campo [SP Requests]reqData

 Cambiar el estado de la petición para que el cliente sepa lo que sucedió

 End for

 “Dormir” un poco antes de volver a comenzar

Hasta que la variable “stop” se vuelva true

Este es el código fuente real:

 ` Método de proyecto SP SERVICES

 ` El procedimiento almacenado comienza

vbStopSPServices:=False

 ` El procedimiento almacenado no necesita acceso en lectura escritura a las tablas...

READ ONLY(*)

 ` ...excepto la tabla [SP Requests]

READ WRITE([SP Requests])

Repeat

 ` Búsqueda de las peticiones no procesadas aún

QUERY([SP Requests];[SP Requests]reqStatus=1)

 ` Proceso de estas peticiones una por una

For(\$vlRecord;1;Records in selection([SP Requests]))

 ` Si el registro de la petición está bloqueado, esperar hasta que esté desbloqueado

While(Locked([SP Requests]))

 ` Espera un segundo antes de intentar nuevamente

DELAY PROCESS(Current process;60)

 ` Trata de obtener acceso lectura-escritura

LOAD RECORD([SP Requests])

End while

 ` Asume que la petición se procesará con éxito

 [SP Requests]reqStatus:=0

Case of

 :([SP Requests]reqType="Server Information")

SP DO SERVER INFORMATION

 :([SP Requests]reqType="Volume List")

SP DO VOLUME LIST

 :([SP Requests]reqType="Browse Directory")

SP DO BROWSE DIRECTORY([SP Requests]reqParams)

 ` ...

 ` OTROS TIPOS DE PETICIONES PODRÍAN AÑADIRSE AQUÍ

 ` ...

Else

 ` El tipo de petición es desconocido, devolver el error -1 (valor arbitrario)

```

        [SP Requests]reqStatus:=-1
    End case
    ` Forzar el estado de la petición a otro diferente de 1
    ` (en caso de una subrutina dar el valor 1)
        If([SP Requests]reqStatus=1)
            [SP Requests]reqStatus:=-3
        End if
    ` Actualizar el registro de la petición
        SAVE RECORD([SP Requests])
    ` Ir a la siguiente petición no procesada
        NEXT RECORD([SP Requests])
    End for
    ` Liberar el último registro procesado
        UNLOAD RECORD([SP Requests])
    ` Espere un segundo antes de continuar respondiendo peticiones
        DELAY PROCESS(Current process;60)
    ` Bucle hasta que se le ordene al procedimiento almacenado detener su ejecución
    Until(vbStopSPServices)

```

El método de proyecto **SP SERVICES** puede utilizarse como modelo para implementar novedosos servicios en una base. En esta sección, detallamos las subrutinas **SP DO SERVER INFORMATION** y **SP DO VOLUME LIST**. La subrutina **SP DO BROWSE DIRECTORY** (que toma como parámetro un parámetro enviado por el cliente en el campo *[SP Requests]reqParams*) no se trata en este documento. Dependiendo del tipo de la petición, el método de proyecto **SP SERVICES** llama a una subrutina cuya tarea es guardar los datos resultantes en el campo *[SP Requests]reqData*. El almacenamiento del registro y el cambio del estado lo realiza el método de proyecto **SP SERVICES**.

Esta es la subrutina **SP DO SERVER INFORMATION** que guarda la información relativa al servidor en el BLOB. Otro método de proyecto extraerá los datos del BLOB en función del equipo cliente.

```

    ` Método de proyecto SP DO SERVER INFORMATION
    TEXT TO BLOB(Application version(*) ; [SP Requests]reqData; UTF8 C string)
    TEXT TO BLOB(Structure file; [SP Requests]reqData; UTF8 C string; *)
    TEXT TO BLOB(Data file; [SP Requests]reqData; UTF8 C string; *)
    PLATFORM PROPERTIES($vIPlatform; $vISystem; $vIMachine)
    VARIABLE TO BLOB($vIPlatform; [SP Requests]reqData; *)
    VARIABLE TO BLOB($vISystem; [SP Requests]reqData; *)
    VARIABLE TO BLOB($vIMachine; [SP Requests]reqData; *)

```

Esta es la subrutina **SP DO VOLUME LIST**, que guarda la información relativa a los volúmenes en el BLOB. Otro método de proyecto extraerá los datos del BLOB en función del equipo cliente.

```

    ` Método de proyecto SP DO VOLUME LIST
    VOLUME LIST($asVName)
    $vISize:=Size of array($asVName)
    ARRAY REAL($arVSize; $vISize)

```

```

ARRAY REAL($arVUsedSpace;$vISize)
ARRAY REAL($arVFreeSpace;$vISize)
For($vIElem;1;$vISize)
    VOLUME
ATTRIBUTES($asVName{$vIElem};$arVSize{$vIElem};$arVUsedSpace{$vIElem}
    ;$arVFreeSpace{$vIElem})
End for
VARIABLE TO BLOB($asVName;[SP Requests]reqData)
VARIABLE TO BLOB($arVSize;[SP Requests]reqData;*)
VARIABLE TO BLOB($arVUsedSpace;[SP Requests]reqData;*)
VARIABLE TO BLOB($arVFreeSpace;[SP Requests]reqData;*)

```

Mostrar la información del servidor en un equipo cliente

Con los métodos de proyecto genéricos *Client post request* y *Client get result*, el método de proyecto **M_SERVER_INFORMATION** muestra, en el equipo cliente, la información devuelta por el procedimiento almacenado. Este método puede estar asociado a un comando de menú o ser llamado, por ejemplo, desde el método de objeto de un botón:

```

` M_SERVER_INFORMATION
C_BLOB(vxData)
C_LONGINT($vIReqID;$vIErrCode;$vIOffset)
` Envío de la petición
$vIReqID:=Client post request("Server Information")
` Prueba del estado de la petición y recepción del resultado
$vIErrCode:=Client get result($vIReqID;->vxData;60)
` Si la petición termina con éxito, muestra el resultado
If($vIErrCode=0)
` Extracción de la información resultante del BLOB
    $vIOffset:=0
    vsServerVersion:=BLOB to text(vxData;UTF8 C string;$vIOffset)
    vsStructureFile:=BLOB to text(vxData;UTF8 C string;$vIOffset)
    vsDataFile:=BLOB to text(vxData;UTF8 C string;$vIOffset)
    BLOB TO VARIABLE(vxData;$vIPlatform;$vIOffset)
    BLOB TO VARIABLE(vxData;$vISystem;$vIOffset)
    BLOB TO VARIABLE(vxData;$vIMachine;$vIOffset)
` Análisis de las propiedades de la plataforma
    vs4DPlatform:="Versión de 4D Server desconocida"
    vsSystem:="Versión del sistema desconocida"
    vsMachine:="Equipo desconocido"
` ...
` Este es el código (no listado) que analiza $vISystem y $vIMachine
` ( ver el ejemplo del comando PLATFORM PROPERTIES)
` ...
` Visualización de la información resultante
    DIALOG([SP Requests];"SERVER INFORMATION")

```

Else

```
ALERT("Error de petición "+String($vErrCode))
```

End if

```
` Ya no se necesita el BLOB
```

```
CLEAR VARIABLE(vxData)
```

Este es el formulario [SP Requests];"SERVER INFORMATION" en ejecución:

Número de versión 4D Server :
F0031120

Ruta del archivo de estructura :
C:\4DFP\base\basePro\MyMusic\MyMusic.4DB

Ruta del archivo de datos :
C:\4DFP\base\basePro\MyMusic\MyMusic.4DD

Versión de la plataforma 4D Server :
4D Server Windows version 5.1

Versión Sistema Operativo:
Windows XP

Hardware de la máquina servidor :
INTEL Compatible

Cancelar Aceptar

Mostrar la lista de volúmenes del servidor en un equipo cliente

Con los métodos de proyecto genéricos *Client post request* y *Client get result*, el método de proyecto **M_SERVER_VOLUMES** muestra, en el equipo cliente, la información en la lista de los volúmenes del servidor devuelta por el procedimiento almacenado. Este método puede estar asociado a un comando de menú o llamado, por ejemplo, desde el método de objeto de un botón:

```
` M_SERVER_VOLUMES
```

```
C_BLOB(vxData)
```

```
` Envío de la petición
```

```
$vReqID:=Client post request("Volume List")
```

```
` Prueba del estado de la petición y recepción del resultado
```

```
$vErrCode:=Client get result($vReqID;->vxData;120)
```

```
` Si la petición termina con éxito, muestra el resultado
```

```
If($vErrCode=0)
```

```
` Extracción de la información resultante del BLOB
```

`$vIOffset:=0`

`BLOB TO VARIABLE(vxDat;asVName;$vIOffset)`

`BLOB TO VARIABLE(vxDat;arVSize;$vIOffset)`

`BLOB TO VARIABLE(vxDat;arVUsedSpace;$vIOffset)`

`BLOB TO VARIABLE(vxDat;arVFreeSpace;$vIOffset)`

`For($vIElem;1;Size of array(arVSize))`

 ` Conversión de bytes en MB

`arVSize{$vIElem}:=arVSize{$vIElem}/1048576`

`arVUsedSpace{$vIElem}:=arVUsedSpace{$vIElem}/1048576`

`arVFreeSpace{$vIElem}:=arVFreeSpace{$vIElem}/1048576`

`End for`

 ` Visualización de la información resultante

`DIALOG([SP Requests];"VOLUME LIST")`

`Else`

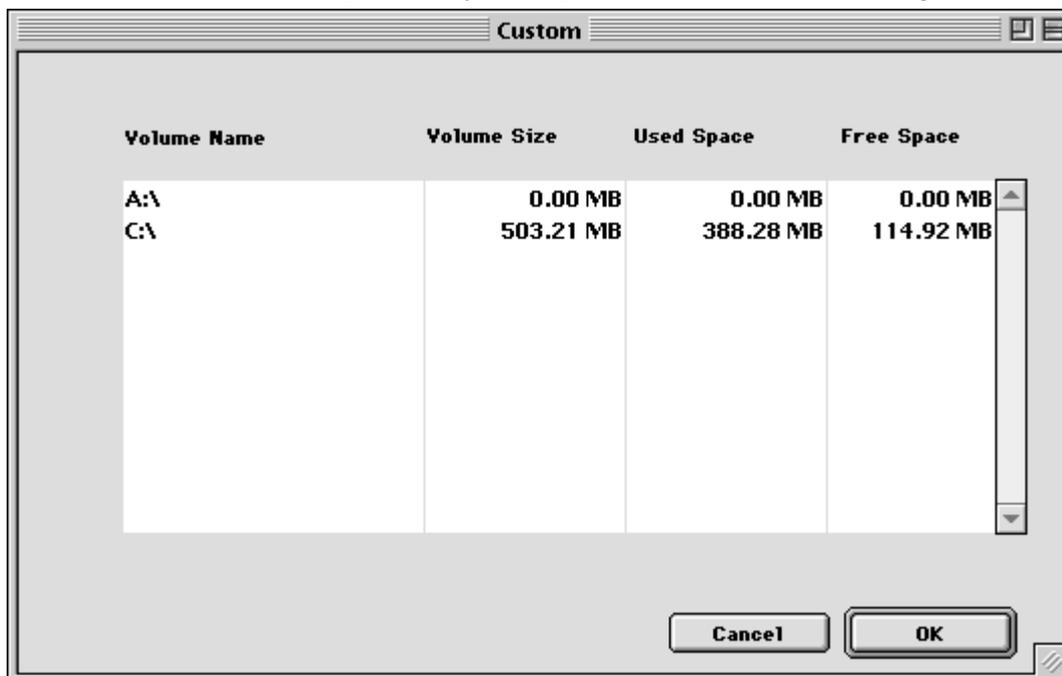
`ALERT("Error de petición "+String($vErrCode))`

`End if`

 ` Ya no es necesario el BLOB

`CLEAR VARIABLE(vxDat)`

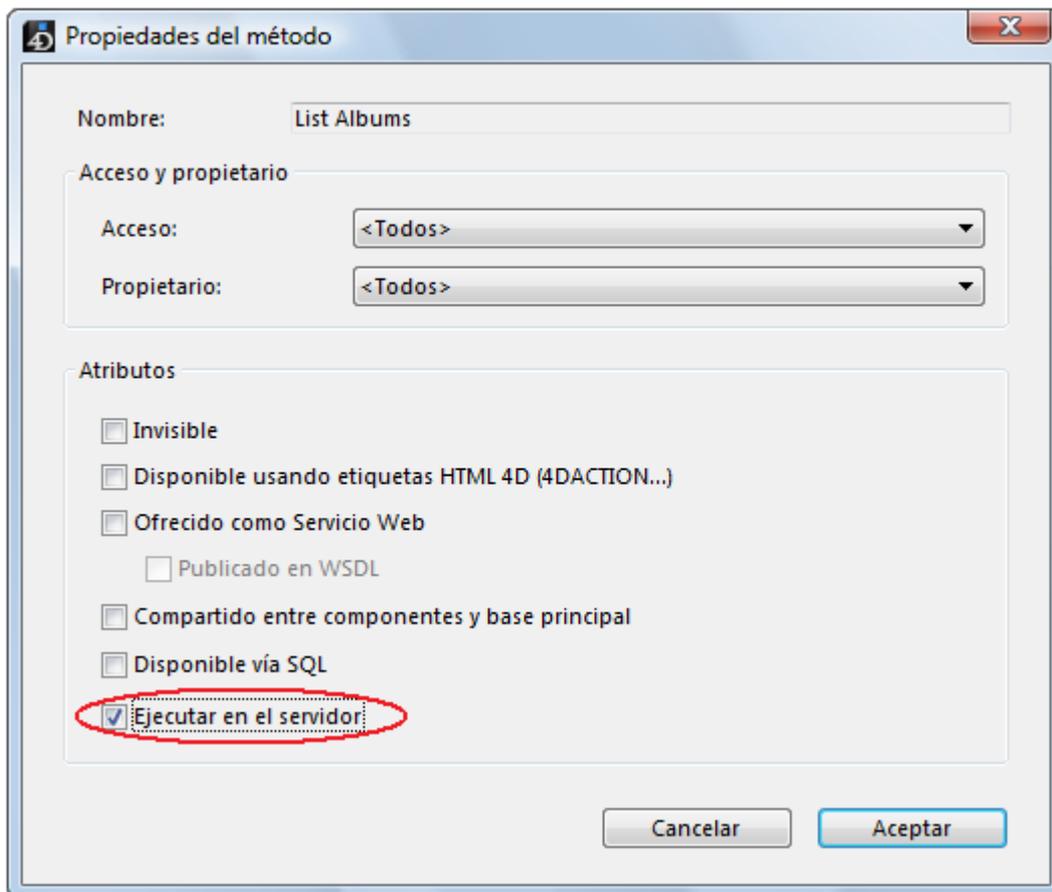
Este es el formulario `[SP Requests];"VOLUME LIST"` en ejecución:



Volume Name	Volume Size	Used Space	Free Space
A:\	0.00 MB	0.00 MB	0.00 MB
C:\	503.21 MB	388.28 MB	114.92 MB

Atributo Ejecutar en servidor

El atributo dle método de proyecto "Execute on Server" puede definirse en la siguiente caja de diálogo de modificación global de los atributos o en la caja de diálogo de las propiedades del método:



Cuando se selecciona esta opción, el método de proyecto siempre se ejecuta en el servidor, sin importar cómo se llama el método.

Nota: este atributo sólo se tiene en cuenta para una aplicación 4D ejecutada en cliente/servidor.

Contexto de ejecución

Cuando este atributo está seleccionado, el contexto de ejecución del método de proyecto es comparable con el de los triggers (ver la sección **4D Server y el lenguaje 4D**): el método en el servidor comparte el mismo contexto de base de datos para el bloqueo de registros y para las transacciones que el contexto correspondiente del lado del cliente, pero no el mismo contexto de lenguaje (variables proceso, conjuntos, selecciones actuales). Sin embargo, a diferencia de un trigger, el método ejecutado en el servidor no comparte el registro actual con el contexto del cliente.

Todos los parámetros del método (\$1, \$2, etc.) se envían en el servidor y el valor del parámetro \$0, si se utiliza, se devuelve al cliente.

A diferencia del comando **Execute on server**, esta opción no provoca la creación del proceso en el servidor. 4D Server utiliza el proceso "gemelo" del proceso cliente que solicitó la ejecución.

Además, esta opción simplifica el principio de delegación de la ejecución de un método en el servidor ya que la transferencia de parámetros se efectúa automáticamente en los dos sentidos, como para una llamada de método "normal". El comando **Execute on server**, funciona asincrónicamente y por lo tanto requiere más programación y uso de semáforos para la lectura de los resultados.

Comandos utilizables

Los métodos que tienen el atributo "Execute on Server" están sometidos a las mismas reglas que los procedimientos almacenados en materia de uso de los comandos del lenguaje 4D. La ejecución de ciertos comandos no está permitida en el servidor, la ejecución de otros no es recomendable. Para mayor información, consulte el párrafo "¿Qué no puede hacer un procedimiento almacenado (ejecutado en el servidor)?" en la sección **Procedimientos almacenados**.

Punteros

Si pasa un puntero a una variable (variable simple, array o elemento de array), el valor apuntado también se envía al servidor. Si el valor apuntado es modificado en el servidor por el método, el valor modificado se devuelve al cliente para actualizar la variable correspondiente del lado del cliente.

Los punteros en una tabla o campo se envían como referencia (número de tabla, número de campo). El valor del registro actual no se intercambia automáticamente.

Nota: esta opción funciona igual en modo interpretado y en modo compilado.

Ejemplo

Este es el código para el método de proyecto Myappli que tiene el atributo "Execute on Server":

```
C_POINTER($1) `Puntero a tabla
C_POINTER($2) `Puntero a campo
C_POINTER($3) `Puntero a array
C_TEXT($4) `Valor a buscar
C_LONGINT($0) `Resultado

`Buscar y enviar valores para cada registro
QUERY($1->;$2->=$4)
While(Not(End selection($1->)))
    APPEND TO ARRAY($3->;myFormula($1))
    NEXT RECORD($1->)
End while
UNLOAD RECORD($1->)
$0:=Records in selection($1->)
```

Del lado del cliente, la llamada al método se hace así:

ARRAY TEXT(myArray;0)

\$vlnum :=*MyAppli*(->[Table_1] ;->[Table_1]Field_1 ;->myArray;"to find")