

MANUAL DE USUARIO



CronSQL

■ Índice

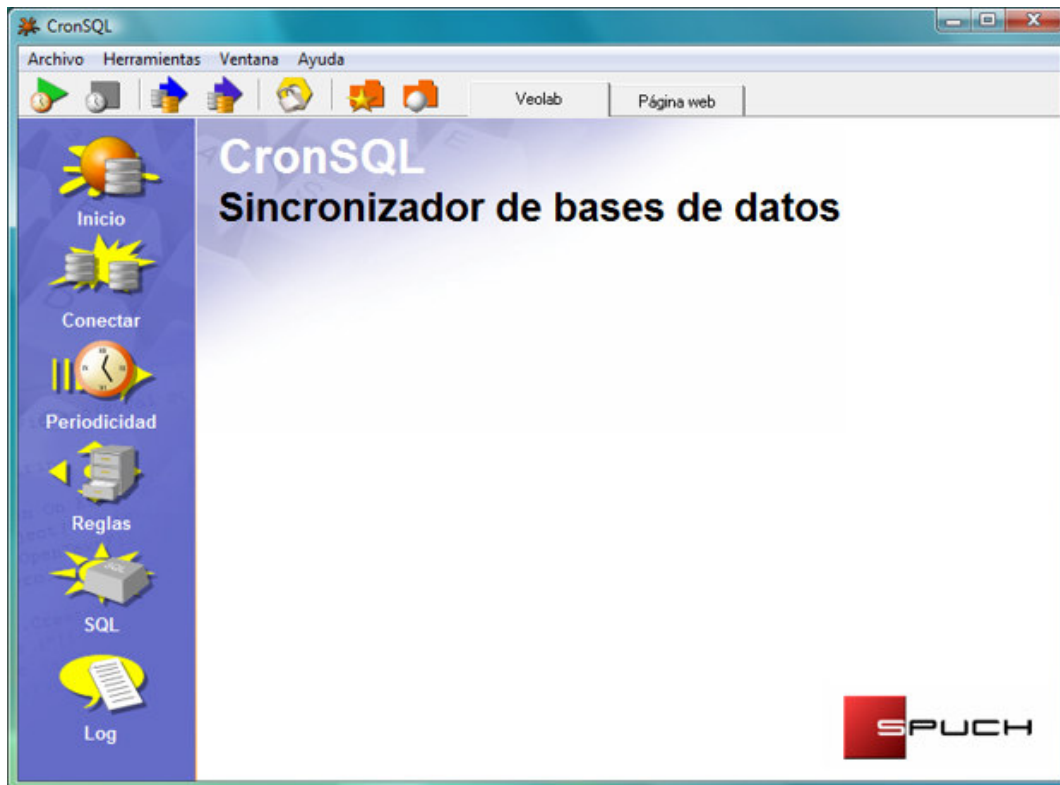
• Índice	2
• Introducción	2
• Pantalla inicial.....	3
• Conectar las bases de datos	4
• Periodicidad de sincronización	6
• Reglas de sincronización.....	7
• Ejecutar consultas SQL	8
• Registro de sucesos (Log).....	9

■ Introducción

CronSQL es una aplicación para la sincronización de bases de datos, que permite mantener varios sistemas de gestión de información sincronizados total o parcialmente.

Aunque CronSQL dispone de una interfaz visual muy sencilla, se requieren conocimientos básicos en consultas SQL, así como de los modelos de datos a sincronizar, para poder configurar correctamente la sincronización.

Este software está orientado principalmente a programadores, desarrolladores, administradores de bases de datos y administradores de sitios web. También está indicado para pymes con software heterogéneo y que precisan centralizar de forma rápida y sencilla el alta de datos sin recurrir a programación a medida o modificaciones del software instalado.



■ Pantalla inicial

En este apartado es posible nombrar la sincronización seleccionada en la pestaña.

El cajetín para habilitar o deshabilitar la sincronización permite mantener una sincronización parcialmente configurada sin que se produzcan errores por intentos de sincronización del servicio, cuando ésta es deshabilitada.

Las demás opciones que constan en esta sección se refieren a toda la aplicación y no sólo a la sincronización de la pestaña actual.

Si se permite bloquear el acceso a las opciones de configuración a través de contraseña, se habilitarán los campos para la definición de la contraseña; el botón de la barra de tareas que permite iniciar el bloqueo; y el cajetín que indica si se desea bloquear automáticamente al iniciar la aplicación. Este último asegura el bloqueo total a las opciones de configuración por usuarios no autorizados, aún cuando el servidor sea reiniciado y esté activado el inicio automático con Windows.

En cualquier momento se puede, desde la barra de tareas o desde las opciones de menú, crear o borrar sincronizaciones. Cuando se borra una sincronización, todos los datos del registro de sucesos, reglas de sincronización, cadenas de conexión, etc. vinculados con la sincronización, se pierden. No hay un límite en el número de sincronizaciones que se pueden definir, y cada una se configura de forma independiente, incluso la periodicidad de sincronización.



CronSQL es una aplicación diseñada para ejecutarse en un ordenador (generalmente un servidor) con algún tipo de conexión a las bases de datos (incluso a través de Internet), de modo que una vez configurado, el funcionamiento es totalmente automático, permitiendo la posibilidad de iniciar con Windows automáticamente, arrancar el servicio de sincronización al iniciar, y mostrarse minimizado en la barra de tareas como un servicio más del sistema.

■ Conectar las bases de datos

En cada sincronización hay una base de datos de origen y otra de destino. Las actualizaciones siempre van desde el origen hasta el destino. Esto es, en cada sincronización se detectan las diferencias del origen y se llevan al destino según las reglas definidas en la sincronización.

En este apartado se definen las cadenas de conexión de origen y destino para la sincronización seleccionada. Se pueden introducir directamente tecleando la cadena de conexión en el recuadro de texto, o bien, se pueden obtener con el asistente mostrado al pulsar el botón *Examinar*.

CronSQL acepta, en principio, cualquier tipo de proveedor OLE DB. Como caso excepcional, también acepta conexiones a bases de datos a través de Internet mediante consultas XML, dado que es bastante frecuente encontrar servidores de alojamiento de páginas web con los accesos externos a las bases de datos capados o bloqueados (generalmente por seguridad). Esto hace imposible una conexión OLE DB a través de Internet, de modo que se ha creado una interfaz en PHP que puede ser utilizada con CronSQL para establecer una conexión con la base de datos de un sitio web. Para ello es necesario subir un archivo nombrado, por ejemplo, como **interfazsql.php** con el siguiente contenido, capaz de obtener consultas en XML:

```
<?php
error_reporting(0);
echo '<?xml version="1.0" encoding="ISO-8859-1"?>','\n';
echo '<CONSULTA>';
if ($enlace = mysql_connect($_GET['host'], $_GET['user'], $_GET['pass'])) {
    if (mysql_select_db($_GET['database'])) {
        if (isset($_GET['consulta'])) {
            $consulta = str_replace("\", \"\", $_GET['consulta']);
            if ($resultado = mysql_query($consulta, $enlace)) {
                if (strtoupper(substr($consulta,0,6)) == "SELECT") {
                    // Construye la estructura
                    echo '<DESCRIPCION>';
                    $num = mysql_num_fields($resultado);
                    for ($i=0; $i < $num; $i++) {
                        $field = mysql_fetch_field($resultado,$i);
                        echo '<CAMPO>';
                        echo '<NOMBRE>',$field->name,'</NOMBRE>';
                        echo '<TIPO>',$field->type,'</TIPO>';
                        echo '<LONGITUD>',$field->max_length,'</LONGITUD>';
                        echo '</CAMPO>';
                    }
                    echo '</DESCRIPCION>';
                    // Construye la parte de datos
                    while ($linea = mysql_fetch_row($resultado)) {
                        echo '<FILAS>';
                        for ($i=0; $i<$num; $i++) {
                            $field = mysql_fetch_field($resultado,$i);
                            echo '<',$field->name,'>',$linea[$i],'</',$field->name,'>';
                        }
                        echo '</FILAS>';
                    }
                } else {
                    echo '<RESULTADO>';
                    echo '<CODIGO>0</CODIGO>';
                    echo '<MENSAJE>La consulta se ha ejecutado correctamente.</MENSAJE>';
                    echo '</RESULTADO>';
                }
            } else {
                echo '<RESULTADO>';
                echo '<CODIGO>-1</CODIGO>';
                echo '<MENSAJE>',mysql_error(),'</MENSAJE>';
            }
        }
    }
}
```

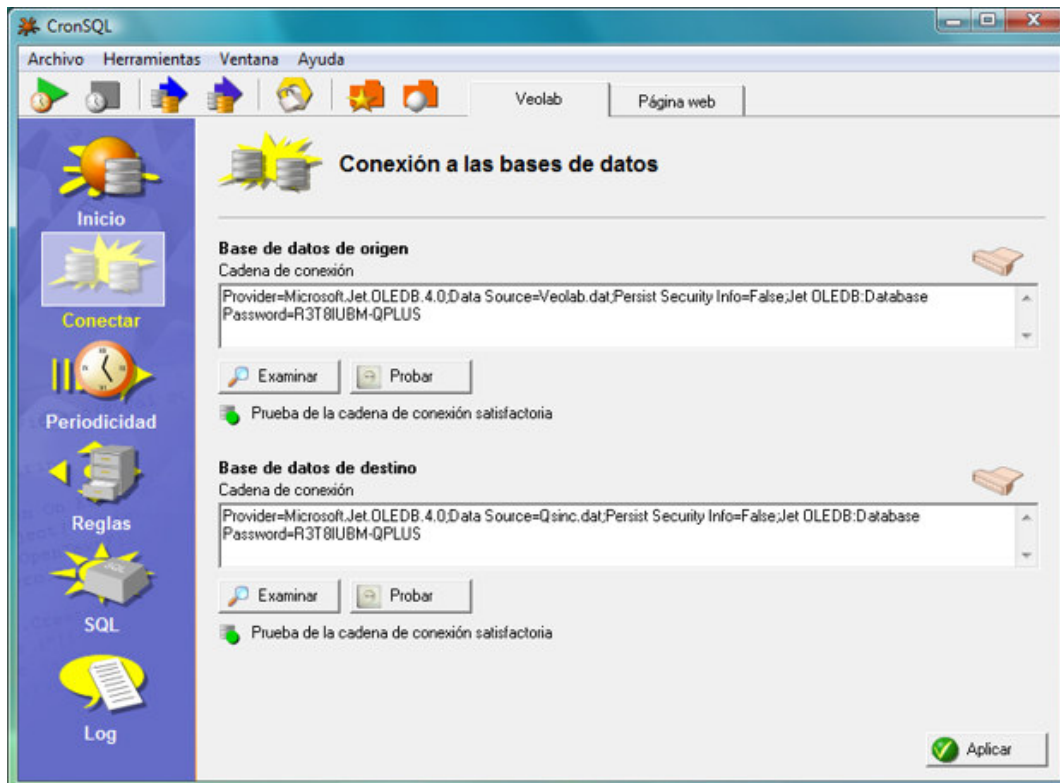
```

        echo '</RESULTADO>';
    }
} else {
    echo '<RESULTADO>';
    echo '<CODIGO>0</CODIGO>';
    echo '<MENSAJE>No ha introducido ninguna consulta.</MENSAJE>';
    echo '</RESULTADO>';
}
} else {
    echo '<RESULTADO>';
    echo '<CODIGO>-2</CODIGO>';
    echo '<MENSAJE>',mysql_error(), '</MENSAJE>';
    echo '</RESULTADO>';
}
} else {
    echo '<RESULTADO>';
    echo '<CODIGO>-3</CODIGO>';
    echo '<MENSAJE>',mysql_error(), '</MENSAJE>';
    echo '</RESULTADO>';
}
}
echo '</CONSULTA>';
?>

```

Para utilizar lo anterior, en la cadena de conexión se deberá escribir directamente la URL del archivo subido que soportará las consultas, en el caso anterior sería la siguiente URL (sustituyendo las palabras en negrita por las que corresponda):

<http://www.sitioweb.com/interfaz.php?host=servidor&database=basededatos&user=usuario&pass=contraseña>



En cualquier momento se puede probar la cadena de conexión introducida. Ambas cadenas de conexión son probadas automáticamente al iniciar la aplicación.

Como en el resto de los apartados, cualquier modificación activa el botón *Aplicar*, y no es efectiva en el sistema hasta que éste es pulsado para aplicar los cambios.

Si se quieren realizar varias modificaciones en distintas tablas entre dos bases de datos, lo habitual es que se definan diferentes reglas de sincronización dentro de la misma sincronización, salvo que se desee una periodicidad diferente para las distintas reglas, entonces se definirían otras sincronizaciones para el mismo origen y destino.

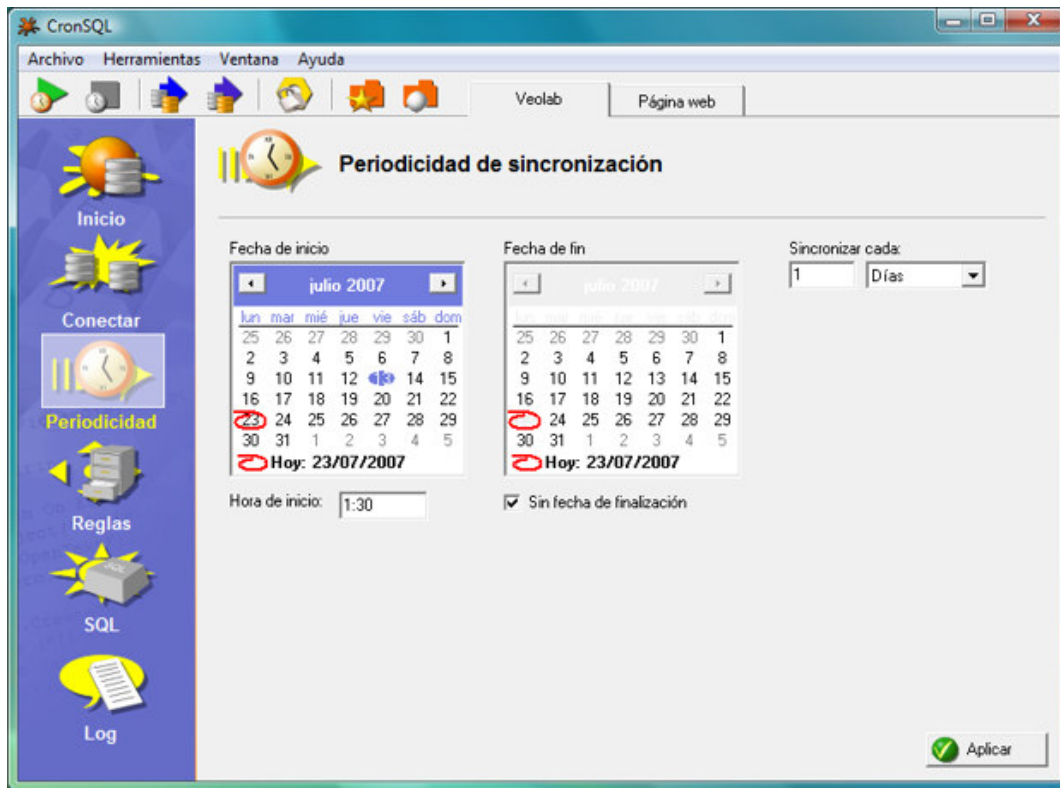
Se pueden establecer sincronizaciones recíprocas entre dos bases de datos creando dos sincronizaciones e intercambiando las conexiones de origen y destino. Este caso no tendría sentido si se sincronizara la misma tabla en ambas direcciones sin restricciones de filas, o si se habilita el borrado en el destino para filas no encontradas en el origen.

■ Periodicidad de sincronización

En este apartado se establece la periodicidad de sincronización cuando el servicio está iniciado. Se puede fijar una fecha y hora de inicio, una fecha de fin, y la frecuencia con que se realizará la sincronización.

Si se desea desactivar una sincronización para que no sea tenida en cuenta al iniciar el servicio de sincronización, basta con deshabilitarla en las opciones de inicio, o bien, definir un rango de fechas obsoleto. Tampoco se produciría sincronización si se introduce un cero en la frecuencia.

La periodicidad mínima de sincronización que se puede establecer es de 1 minuto. Dependiendo de las funciones del servidor, de la probabilidad de modificación de los datos y de la necesidad de inmediatez de sincronización, se elegirá la frecuencia más adecuada. Cuanto más largo sea el periodo de sincronización menos recursos consumirá ésta.



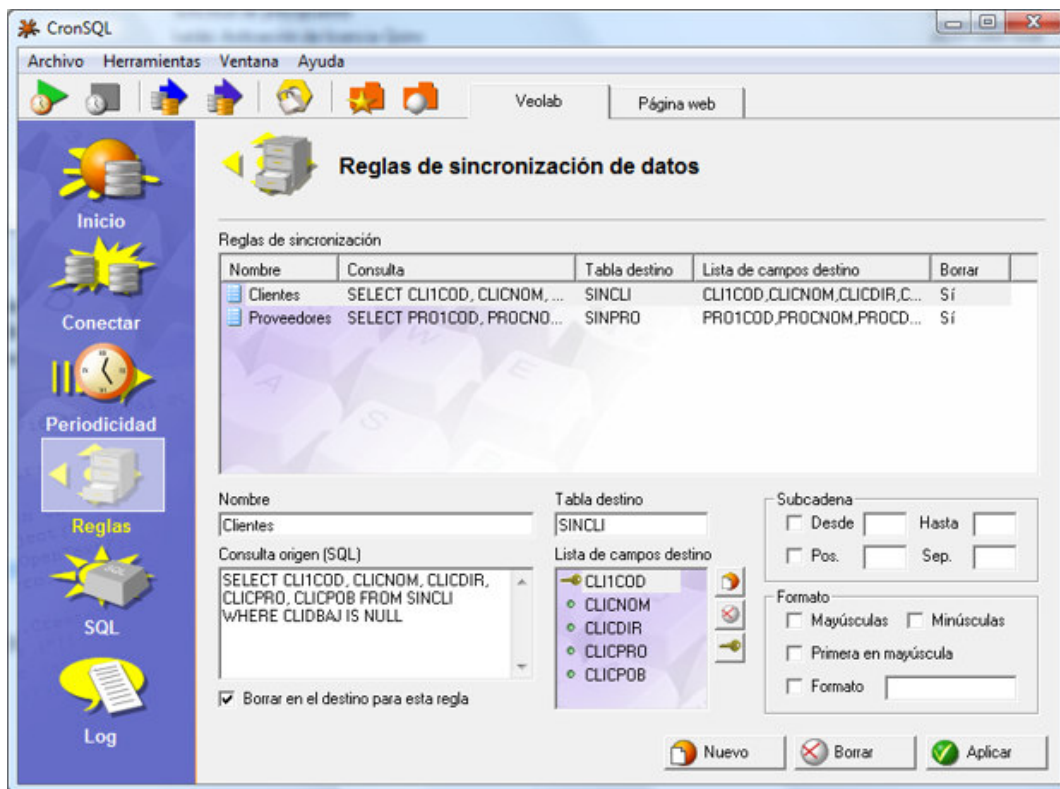
Es interesante considerar una hora de inicio en la que la base de datos esté desocupada y no vaya a ralentizar otras tareas, por ejemplo, cualquier hora fuera del horario de oficina. En base a la fecha y hora de inicio se puede determinar la fecha y hora del resto de sincronizaciones, en el caso de que sea diaria, el cálculo es trivial.

■ Reglas de sincronización

Las reglas de sincronización son las que realmente definen las actualizaciones que se deben llevar a cabo desde el origen al destino. Se pueden crear tantas reglas como sea necesario para una sincronización. Generalmente cada regla corresponderá a una tabla en el destino, aunque no necesariamente en el origen, por ejemplo, si se desea descomponer una tabla del origen en varias tablas en el destino, se utilizará una regla para cada tabla en el destino con la misma tabla en el origen, aunque variando las condiciones de la consulta SQL.

Cada regla se compone de un nombre que la identifica, una consulta SQL (sólo se acepta el comando SELECT), la tabla destino y una lista de los campos de la tabla destino que corresponderían a los enunciados en la consulta SQL.

El hecho de permitir introducir consultas con el comando SELECT amplía la potencia de sincronización, ya que esto permite tomar datos de agrupaciones, vistas, tablas parciales, uniones, etc.



El cajetín *Borrar en el destino para esta regla* se activa por defecto. Cuando está activado, las líneas de la tabla destino que no tengan correspondencia en la consulta origen serán borradas. Este cajetín se suele desactivar cuando se hacen actualizaciones parciales de una tabla, por ejemplo para sincronizar una tabla de clientes que cumplen ciertas condiciones, en base a una consulta en el origen, sin borrar los demás clientes que no figuran en dicha consulta.

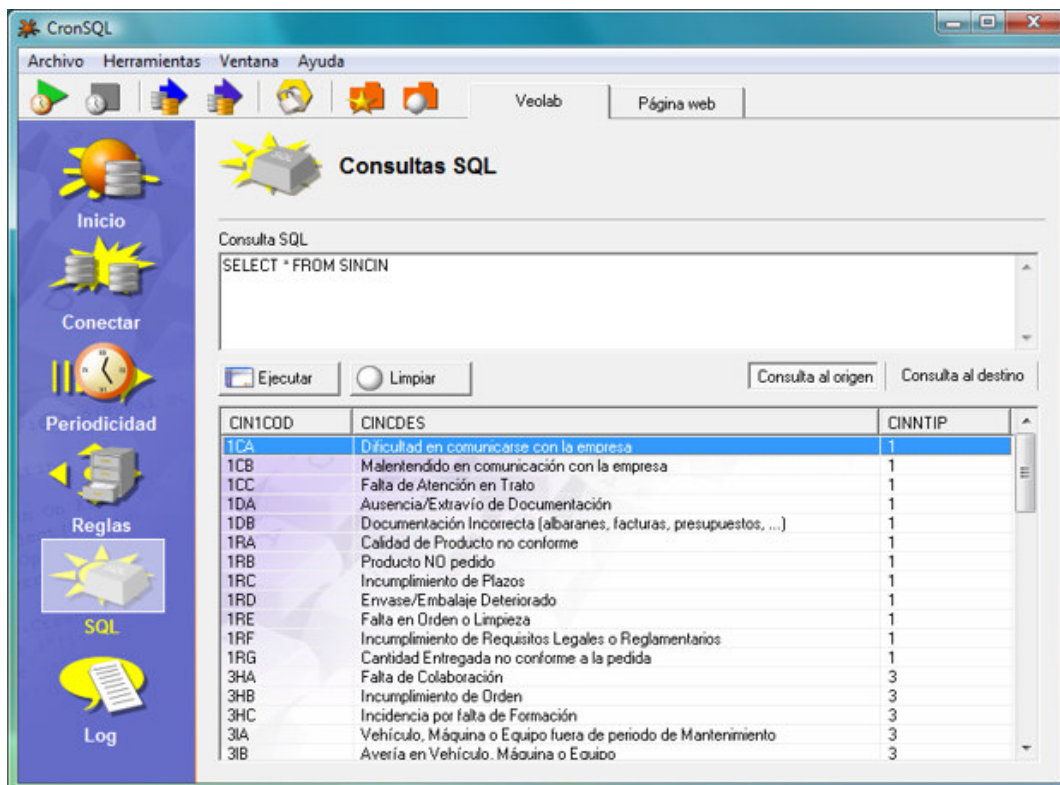
La lista de campos de la tabla de destino debe coincidir en número y en orden con los campos definidos en el SELECT de origen. Además se debe identificar el campo que funcionará como clave, éste no necesariamente debe coincidir con la clave real definida en el modelo de datos, aunque suele ser lo más habitual. Puede haber más de un campo clave. Los campos marcados como clave identifican la fila a actualizar y realizarán dicha operación en base a los mismos.

Puede ser muy frecuente el hecho de querer sincronizar datos con cierto formato, pasando el texto a minúsculas o mayúsculas, tomando sólo una parte de un campo, etc. Para estos casos, teniendo seleccionado el campo en la lista de campos destino, se puede configurar con la sección *Subcadena*, la posibilidad de tomar el contenido del origen entre dos posiciones (*Desde / Hasta*) o bien utilizando separadores (*Pos. / Sep.*). Este segundo caso es especialmente interesante para reutilizar campos de textos sobrantes en la aplicación base (que se toma como origen de los datos), formando al usuario de modo que utilice un carácter de separación para introducir los distintos valores.

La sección *Formato* permite aplicar los formatos indicados a los valores tomados del origen a la hora de aplicarlos al destino. Se puede convertir cadenas de texto a minúsculas, mayúsculas, sólo la primera en mayúsculas, y formato libre, basado en cualquier expresión de formato contemplada por Visual Basic, tanto para cadenas, números como fechas. (Ejemplos: ##,##0.00, dd/mm/yy, Hh:Nn, &&&&, etc.

■ Ejecutar consultas SQL

Para realizar consultas, actualizaciones o para lanzar cualquier tipo de comando SQL en la base de datos de origen o de destino de cualquier sincronización definida, existe este apartado que facilita las pruebas necesarias antes de configurar una sincronización.



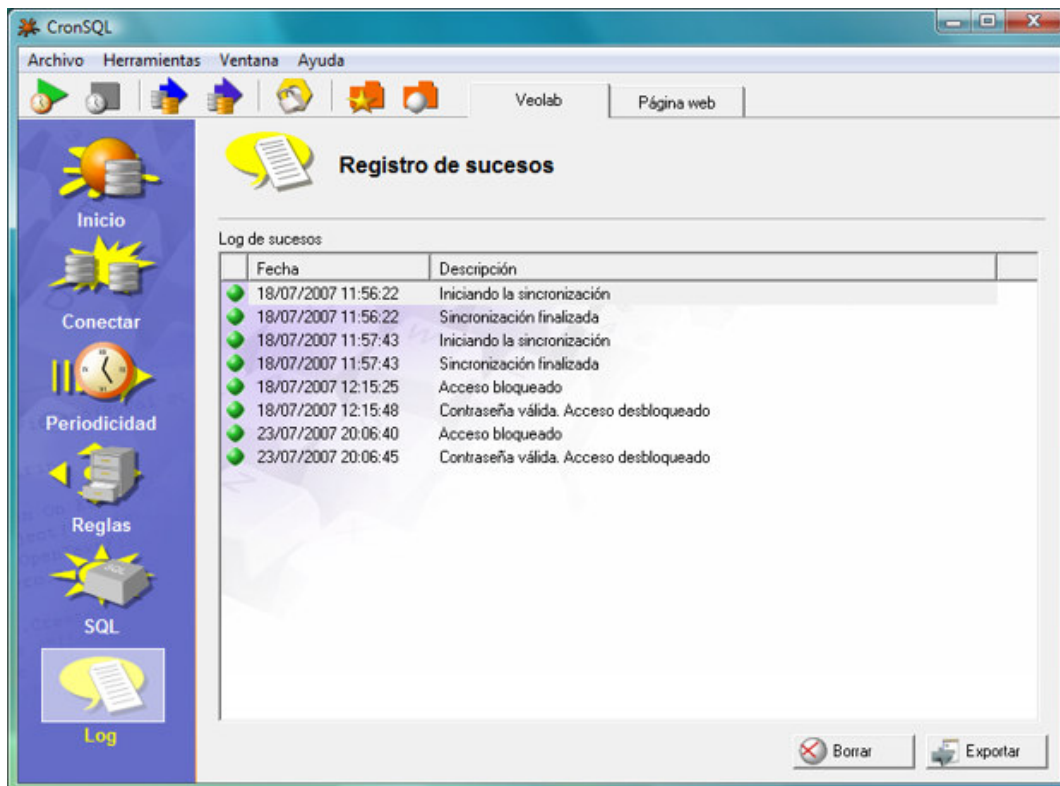
■ Registro de sucesos (Log)

Cualquier suceso y/o error producido durante la activación del servicio de sincronización queda registrado en CronSQL. Existe un registro de sucesos diferente para cada sincronización, aunque los eventos comunes quedarán registrados en todos los registros de sucesos.

El registro de sucesos debe ser consultado con la frecuencia que se considere, una vez se ha activado el servicio de sincronización, para determinar si se producen errores en la sincronización.

Cuando una sincronización falla, por el motivo que sea, el servicio lo reintenta hasta cuatro veces, apuntando el hecho en el registro de sucesos correspondiente a la sincronización que falló. El fallo en una sincronización no interrumpe el servicio para las demás sincronizaciones.

Si se desea obtener dicho registro de Log en archivo de texto de plano, se puede exportar usando el botón correspondiente.



CronSQL

Archivo Herramientas Ventana Ayuda

Veolab Página web

Registro de sucesos

Log de sucesos

Fecha	Descripción
18/07/2007 11:56:22	Iniciando la sincronización
18/07/2007 11:56:22	Sincronización finalizada
18/07/2007 11:57:43	Iniciando la sincronización
18/07/2007 11:57:43	Sincronización finalizada
18/07/2007 12:15:25	Acceso bloqueado
18/07/2007 12:15:48	Contraseña válida. Acceso desbloqueado
23/07/2007 20:06:40	Acceso bloqueado
23/07/2007 20:06:45	Contraseña válida. Acceso desbloqueado

Borrar Exportar