TUTORIAL

Versión 2026

La información que contiene este manual no tiene carácter contractual y puede estar sujeta a cambios sin previo aviso.

La aplicación a la que se hace referencia en este manual se suministra bajo licencia de uso. Los términos de esta licencia estipulan las condiciones de uso, copia o reproducción autorizadas de la aplicación. Todos los demás casos están prohibidos.

Ninguna parte de este manual se puede copiar, reproducir ni transmitir de ninguna manera o por ningún medio sin el permiso expreso por escrito de TEKLYNX Corporation SAS para ningún propósito más que el del uso personal del comprador.

©2026 TEKLYNX Corporation SAS,

Reservados todos los derechos.

Tabla de contenido

Acerca de este manual ... 6

Convenciones tipográficas ... 6

Acerca de este producto ... 6

Conexión con bases de datos ... 7

Nociones básicas ... 7

Instalación de un origen de datos ODBC e importación de datos ... 8

Importación de datos ... 9

Creación de objetos variables ... 9

Active Query Builder ... 10

Introducción ... 10

Agregar un objeto a la consulta ... 11

Editar propiedades de objetos ... 12

Unir tablas ... 12

Ordenar campos de resultados ... 13

Definir criterios ... 14

Definir una consulta con parámetros ... 14

Tabla Resultado de la consulta ... 15

Database Manager ... 17

Interfaz de estructura de bases de datos ... 17

Cómo escoger una base de datos en la lista de conexiones ... 17

Cómo escoger una tabla de una base de datos ... 18

Cómo añadir una tabla a la base de datos activa ... 18

Cómo suprimir una tabla de la base de datos activa ... 18

Cómo mostrar u ocultar los datos de la tabla activa ... 19

Configurar un campo como campo clave ... 19

Cómo configurar el tipo de contenido de un campo ... 19

Cómo configurar el tamaño máximo de un campo 19
Cómo permitir que un campo quede vacío 19
Interfaz de edición de bases de datos 20
Cómo encontrar un registro que corresponda al contenido de un campo 20
Cómo seleccionar todos los registros idénticos 20
Cómo seleccionar un registro idéntico 21
Cómo crear un registro nuevo 21
Cómo modificar un registro 21
Cómo suprimir un registro 21
Interfaz de consulta de bases de datos 22
Cómo añadir una consulta 22
Cómo seleccionar o anular la selección de uno o más campos 22
Cómo modificar el orden de los campos seleccionados 22
Cómo crear un filtro con datos pre definidos 23
Cómo aplicar un operador lógico a varios filtros 23
Cómo suprimir un filtro 23
Cómo modificar un filtro escrito en lenguaje SQL 23
Interfaz de impresión 24
Cómo crear un modelo nuevo de etiqueta a imprimir 24
Cómo seleccionar un modelo de etiqueta ya existente 24
Cómo seleccionar la impresora a utilizar 24
Cómo configurar la impresora seleccionada 25
Cómo seleccionar manualmente los registros a imprimir 25
Cómo seleccionar automáticamente los registros a imprimir 25
Fórmula 26
Fuente de datos Fórmula 26
Funciones 26
Operadores 26
Operadores aritméticos 27
Operadores de comparación 27

Operador de concatenación ... 27
Operador lógico ... 27
Funciones de cálculo de caracteres de control ... 28
Funciones de conversión ... 30
Funciones de conversión ATA 2000 ... 36
Funciones relativas a fechas y horas ... 39
Funciones lógicas ... 46
Funciones matemáticas ... 48
Funciones relativas a texto ... 51
Cómo configurar las propiedades de una fuente de datos de tipo Fórmula ... 57
Ejercicio : Cálculo del "Modulo" adecuado ... 58
Método de cálculo de un carácter de control ... 58
Para adicionar el resultado del cálculo de pesos: ... 59
Para calcular el carácter de control: ... 59
Para calcular el mensaje a codificar: ... 59
Para crear el código de barras: ... 60
Instalación de la versión de red ... 61
Descripción funcional ... 61
Procedimiento de instalación ... 61
Preliminares ... 61
Presentación de Network Manager ... 62
Instalación de Network License Manager en el servidor ... 62
Configuración ... 63
Para configurar Administrador de usuarios ... 63
Inicio del administrador de licencias (License Service) ... 64
Para iniciar License Service ... 64
Instalación de la aplicación en las estacione ... 64
Para instalar la aplicación en una estación de trabajo ... 64

Acerca de este manual

Convenciones tipográficas

Para facilitar la lectura de esta guía, se han adoptado las siguientes convenciones de presentación:

- Los términos tomados de la interfaz (comandos, menús, etc.) aparecen en **negrita**.
- Las teclas se presentan en mayúsculas, como en el ejemplo siguiente: "Pulse la tecla SHIFT".
- Las listas numeradas indican cómo realizar una tarea, paso a paso,.
- La conjunción - o -, colocada al lado de un párrafo, indica una alternativa para ejecutar una misma acción.
- Cuando el comando de un menú tiene submenús, el nombre del menú, seguido del comando que se debe escoger, aparecen en negrita. Por ejemplo, "Escoja **Archivo** > **Abrir**..." significa que hay que señalar al menú **Archivo** y escoger el comando **Abrir**....

Acerca de este producto

Pudiera ocurrir que el producto que tiene en sus manos no disponga de determinadas funciones descritas en esta guía.

Consulte la ficha descriptiva que se incluye en el paquete del producto para conocer la lista exhaustiva de funciones y características de su versión.

Conexión con bases de datos

Nociones básicas

En este capítulo descubrirá la potencia de esta aplicación. Para ello vamos a vincular una etiqueta (el contenedor) con una base de datos (el contenido) mediante conexiones por ODBC (Open Data Base Connectivity) y OLE DB (Object Linking and Embedding Database).

La función de una base de datos es almacenar información. La información o datos está organizada en tablas de dos dimensiones. A esto se le llama relación. Cada fila o línea de la tabla es un registro. Cada registro contiene toda la información relativa a un mismo objeto, clasificada en los diferentes campos que delimita cada columna de la tabla.

Una base de datos puede contener varias tablas diferentes. Para relacionar las diferentes tablas de una misma base de datos se puede recurrir a junturas. Veremos más adelante, por medio de un ejemplo concreto, cómo crear junturas.

ODBC

Es una tecnología de conexión convertida en norma para conectarse con bases de datos. Gracias a ODBC, vincular una aplicación como ésta de etiquetado con determinado número de bases de datos es una operación muy sencilla.

OLE DB

Es una tecnología de conexión que permite conectarse con cualquier tipo de base de datos y acceder a los datos almacenados en los sistemas de mensajería.

Esta aplicación suministra varios controladores ODBC con los que conectarse a los tipos de bases de datos más habituales. Entre ellos encontramos:

- Microsoft Access Driver (*.mdb)
- Microsoft Excel Driver (*.xls)
- Microsoft FoxPro Driver (*.dbf)
- ...

Instalación de un origen de datos ODBC e importación de datos

Para acceder a los datos, el primer paso consiste en instalar y configurar el origen de datos adecuado. En este primer ejemplo se instalará un origen de datos para una base de datos de Access.

Para instalar el origen de datos:

1. Abra la etiqueta Recto_00.tpl, situada en la carpeta InstallDir\Samples\Forms\Tutorial.
   Se trata de la carátula de un disco compacto, en la que se indica el título del disco y los de los trozos grabados. Los datos que vamos a introducir en la etiqueta están almacenados en una base de datos llamada Volmusic.mdb.
2. Escoja **Fuente de datos > Base de datos > Crear/Modificar una consulta...**
   Se abre el cuadro de diálogo **Preparación de una consulta**.
3. Escoja ... a la derecha de la lista Seleccione la fuente de datos. Se abre el cuadro de diálogo Administrador de orígenes de datos ODBC

   Nota: ODBC es un programa totalmente independiente del aplicación de etiquetado. También se puede abrir el **Administrador de controladores ODBC** mediante el panel de control de Windows.
4. En la ficha DSN de usuario escoja **Añadir...** y, a continuación, seleccione el controlador Microsoft Access Driver (*.mdb). Por último, pulse **Terminar**.
   Se abre el cuadro de diálogo Instalación ODBC para Microsoft Access.
5. Introduzca un nombre en la zona **Nombre** del origen de datos como, por ejemplo, Classical music source y, si lo desea, introduzca una breve descripción.
   Ahora sabemos que un origen de datos está vinculado con una base de datos. El paso siguiente consiste en seleccionar la base de datos utilizada en nuestro ejemplo, es decir VOLMUSIC.MDB.

Para seleccionar la base de datos:

1. En el cuadro de diálogo Instalación ODBC para Microsoft Access, elija Seleccionar en la zona Base de datos.
2. Seleccione la base de datos, situada en \TUTORIAL\Volmusic.mdb y pulse **Terminar**.
3. Cierre el cuadro de diálogo Instalación ODBC para Microsoft Access escogiendo **OK** y salga del **Administrador de orígenes datos ODBC** escogiendo de nuevo **OK**.

Importación de datos

Ahora que ya está instalado el origen de datos, vamos a importar los datos almacenados en la base e introducirlos en la etiqueta.

Para importar los datos:

1. Si el cuadro de diálogo **Preparación de una consulta** no está abierto, escoja **Fusión Crear/Modificar una consulta...**
2. Seleccione Classical music source en la lista **Seleccione la fuente de datos** y, a continuación, la tabla volumes.
3. Escoja **Seleccionar todo** y pulse **Aceptar**. Esta acción permite crear automáticamente todas las variables de tipo Base de datos a partir de los campos de la base de datos. La lista de variables aparece subordinada a la rama **Base de datos** de la ficha **Fuentes de datos**.
4. Guarde el documento.

   La aplicación le pregunta si desea guardar la consulta.
5. Escoja **Sí**, seleccione InstallDir\Samples\Forms\Tutorial\ y escriba VOLMUSIC.CSQ como nombre de consulta.

Creación de objetos variables

1. Seleccione las variables que aparecen subordinada a la rama **Base de datos** de la ficha **Fuentes de datos**, a continuación, arrastrarlo y soltarlo en el área de trabajo.
2. Seleccionar Texto en el menú contextual.

Active Query Builder

Active Query Builder es un componente para generar consultas visuales que le permiten generar consultas SQL complejas por medio de una interfaz intuitiva para generar consultas visuales.

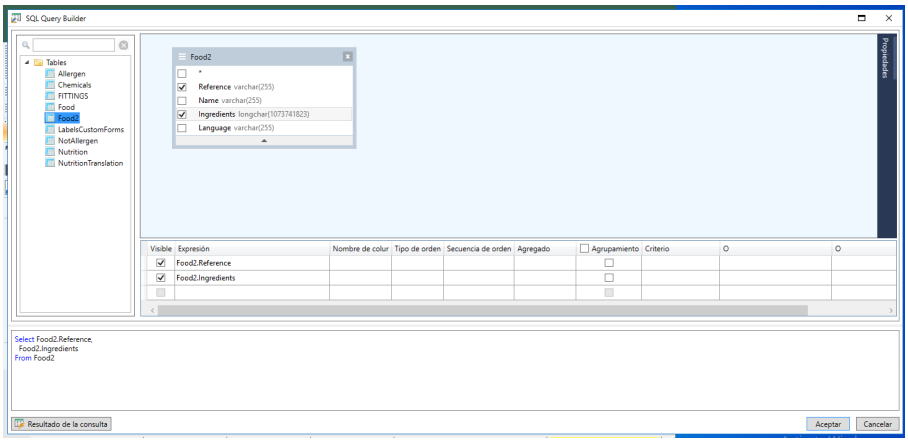
Para trabajar con Active Query Builder, necesita tener conocimientos básicos de conceptos SQL. Active Query Builder le ayudará a escribir correctamente el código SQL ocultando los detalles técnicos, pero sólo si entiende los principios de SQL será posible lograr los resultados deseados.

Introducción

Este es el aspecto de Active Query Builder cuando se inicia.

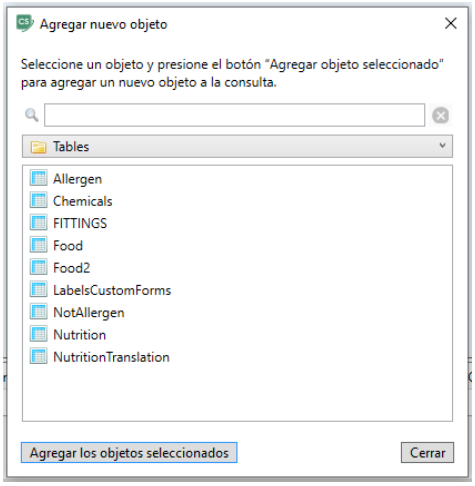
La ventana principal puede dividirse en las partes siguientes:

- El **Área para general consultas** es el área principal donde se muestra la representación visual de la consulta. Este área le permite definir objetos de base de datos de origen y tablas derivadas, definir vínculos entre ellos y configurar propiedades de tablas y vínculos.
- El **Panel de columnas** está ubicado debajo del área para generar consultas. Se utiliza para realizar todas las operaciones necesarias con columnas de salida para consultas y expresiones. Aquí puede definir alias de campo, ordenar y agrupar, y definir criterios.
- La **Zona de vista de tabla** está situada a la izquierda. Aquí puede examinar su consulta y localizar rápidamente cualquier parte de ella. Utilice el campo **Buscar** situado en la parte superior del panel para especificar su búsqueda.
- El control de páginas en la parte superior del área para generar consultas le permitirá cambiar entre la consulta principal y las subconsultas.
- El área pequeña con la letra "Q" en la esquina del área para generar consultas es el control de manejo de la unión de subconsultas. Aquí puede agregar uniones de subconsultas nuevas y realizar todas las operaciones necesarias con ellas.



Agregar un objeto a la consulta

Para agregar un objeto a la consulta haga clic con el botón derecho en el área para generar consultas y seleccione el elemento Agregar objeto en el menú desplegable.

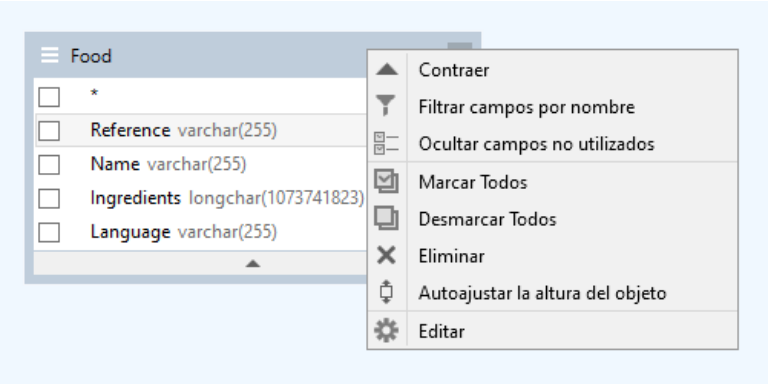


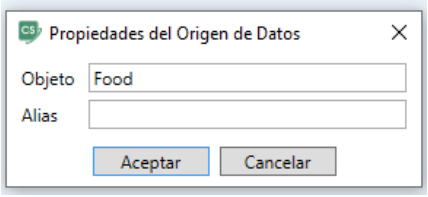
La ventana **Añadir Objeto Nuevo** le permite añadir múltiples objetos de una vez. Los elementos de tabla aparecen en el cuadro de lista. El filtro **Mostrar todos los objetos** mostrará todos los objetos disponibles para seleccionar. El **campo Buscar** en la parte superior de la ventana se utilizar para buscar un objeto que necesita. Puede seleccionar uno o varios objetos manteniendo pulsada la **tecla Ctrl** y luego pulsando el botón **Añadir objetos seleccionados** para añadir estos objetos a la consulta. Puede repetir esta operación varias veces. Cuando haya terminado de añadir objetos, haga clic en el botón **Cerrar** para ocultar esta ventana.

Para quitar un objeto de la consulta, haga clic en el botón **Cerrar** en el encabezado del objeto.

Editar propiedades de objetos

Puede cambiar las propiedades de cada objeto agregado a la consulta si hace clic con el botón derecho en el objeto y selecciona el elemento **Editar...** del menú desplegable o simplemente haciendo doble clic en el encabezado del objeto.

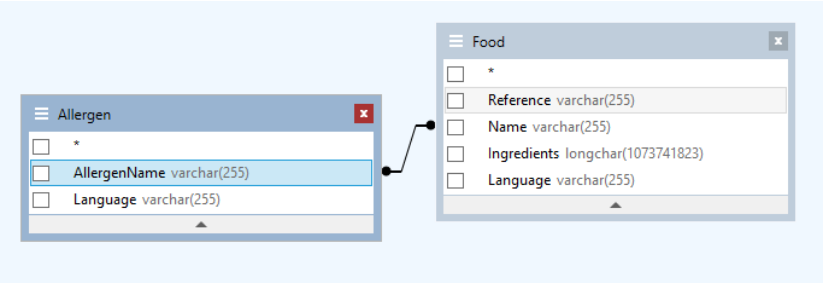




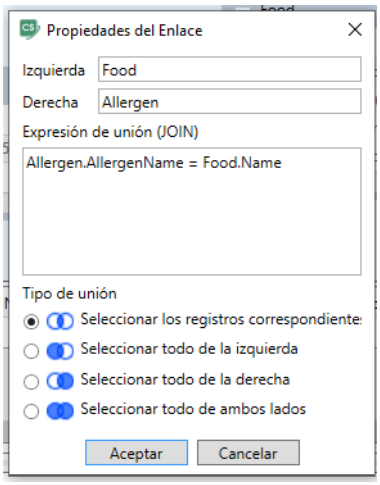
El diálogo **Propiedadesdel origen de datos** puede variar de un servidor a otro, pero al menos la propiedad **Alias** será la misma para todos los servidores de base de datos.

Unir tablas

Para crear un vínculo entre dos objetos (p.ej. unirlos) tiene que seleccionar el campo por el cual desea unir un objeto con el otro y arrastrarlo hasta el campo correspondiente del otro objeto. Después de finalizar el arrastre, aparecerá una línea conectando los campos vinculados.



El tipo de unión creado por defecto es UNIÓN INTERIOR, p.ej. sólo registros coincidentes de ambas tablas serán incluidos en el conjunto de datos resultantes. Para definir otros tipos de uniones tendrá que hacer clic con el botón derecho en el vínculo y seleccionar el elemento **Editar...** en el menú desplegable o simplemente hacer doble clic para abrir el diálogo **Propiedades del enlace**. Este diálogo le permite definir el tipo de unión y otras propiedades de vínculo.



Para eliminar un vínculo entre objetos, haga clic con el botón derecho del ratón en el vínculo y seleccione la opción **Quitar** en el menú desplegable, o seleccione la línea de vínculo y presione el botón **Eliminar**.

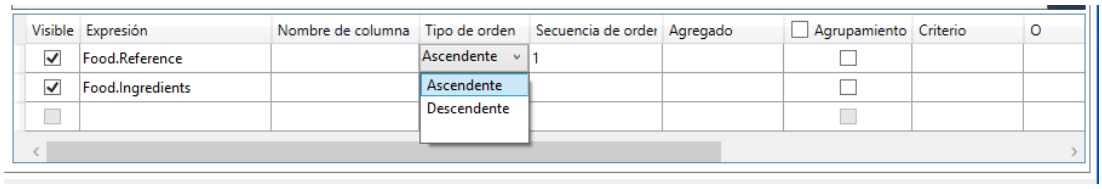
Ordenar campos de resultados

Para activar la ordenación de los campos de resultados de consultas tiene que utilizar las columnas **Tipo de orden** y **Orden** en el **Panel de columnas**.

La columna **Tipo de orden** le permite especificar el modo en el que los campos se ordenan - en orden **Ascendente** o **Descendente**.

La columna **Orden** le permite establecer el orden en el que los campos se ordenan, si se tiene que ordenar más de uno.

Para desactivar ordenar por algún campo tiene que borrar la columna **Tipo** para este campo.



Definir criterios

Para definir criterios para la expresión listada en el **Panel de columnas** tiene que utilizar la columna **Criterios**.

Allí tendrá que escribir el criterio omitiendo la expresión misma. Para obtener el criterio siguiente en su consulta

> **WHERE** (field >= 10) **AND** (field <= 20)
> tendrá que escribir
> >= 10 **AND** <= 20 en la columna **Criterios**.

También puede usar la **lista desplegable** en la columna **Criterios** para pegar expresiones y operadores habituales en el campo editar. Para la creación de criterios avanzados, use el botón **Editor de expresiones** en la columna **Criterios**. Aparece el cuadro de diálogo **Editor de expresiones**.

Puede especificar varios criterios para una única expresión utilizando las columnas **O...**. Estos criterios se concatenarán en la consulta utilizando el operador **O**.

Definir una consulta con parámetros

Query Builder le deja crear una consulta con parámetros donde el valor del parámentro se contiene en una variable.

Nota: Tiene que haber creado una variable de antemano.

1. Arrastre y suelte la tabla en la que se realizará su consulta.
2. Seleccione el(los) campo(s) al(a los) que se aplicará/n el criterio o los criterios.
   3. En la columna **Criterio** o en el campo de edición en formato SQL especifique la variable que se va a utilizar como el objeto de un criterio de búsqueda.

      Ejemplo: Para buscar el valor de la variable Var0 que creó de antemano:

- **En SQL:**

  SELECT [Table].*
  FROM [Table]
  WHERE [Table].Field = APPLICATION.DOCUMENT.Var0

- **Columna Criterio**
  = APPLICATION.DOCUMENT.Var0

4. Haga clic en el botón **Resultado de consulta** para ver el resultado de su consulta.

Tabla Resultado de la consulta

Para abrir la tabla **Resultado de la consulta**, escoja el botón del cuadro de diálogo **Preparación de una consulta,** al que se accede mediante la barra de herramientas de **Navegación de la base de datos de fusión**, o mediante el menú **Fuente de datos > Bases de datos**.

Esta tabla permite ver el resultado de la consulta o buscar una palabra determinada para imprimir las etiquetas correspondientes.

Nota: La etiqueta "<Datos binarios>" aparece en el cuadro de diálogo resultado de la consulta para campos de bases de datos que contienen el **tipo de datos BLOB** (p. ej.: archivos de imágenes).

La tabla **Resultado de la consulta** incluye:

- **Funciones de búsqueda**

  Un campo de búsqueda permite introducir el nombre del campo en cuyo contenido se efectuará la búsqueda.

  El valor de la búsqueda permite introducir la información que se busca.

  Búsqueda de datos en todo el campo o al principio del campo .

- **Funciones de desplazamiento entre los registros resultantes de la consulta**

  Primera ocurrencia

Ocurrencia precedente

Ocurrencia siguiente

Última ocurrencia

La tabla de resultados

Muestra las ocurrencias encontradas tras la búsqueda.

La búsqueda

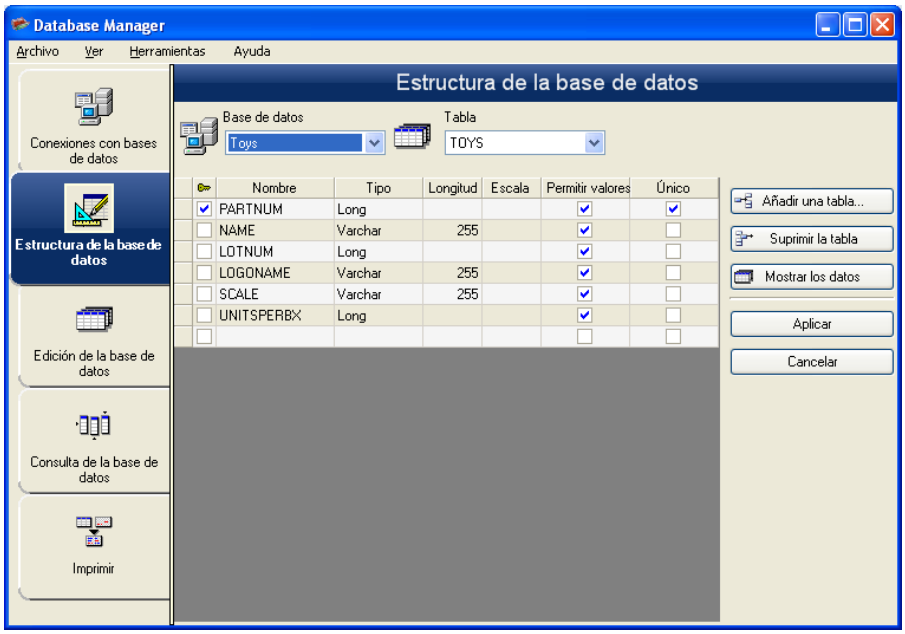
Vuelva a realizar la búsqueda y actualización de la red.

Repetir consulta

Repetir la solicitud y actualizar la cuadrícula.

Database Manager

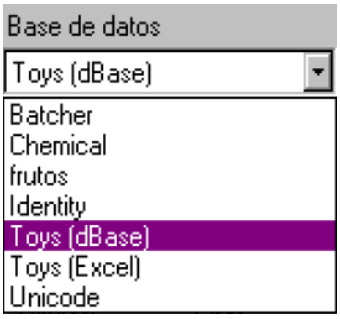
Interfaz de estructura de bases de datos



Así se presenta la ventana mediante la cual se administra la estructura del archivo de bases de datos añadiendo, modificando o suprimiendo tablas y campos.

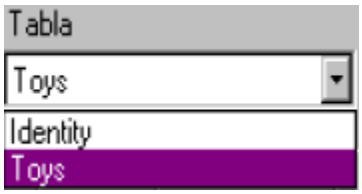
Cómo escoger una base de datos en la lista de conexiones

1. Haga clic en la lista desplegable **Base de datos**.
2. Seleccione el valor que desee



Cómo escoger una tabla de una base de datos

1. Despliegue la lista **Tabla**.
2. Seleccione el valor que desee.



Cómo añadir una tabla a la base de datos activa

1. Escoja **Añadir una tabla**.
2. Introduzca el nombre de la tabla nueva.
3. Escoja **Aceptar**.

También puede crear una tabla copiando la estructura de otra tabla presente en la base de datos seleccionada. Para ello:

1. Marque la casilla **Duplicar con**.
2. Despliegue la lista.
3. Seleccione el valor que desee.
4. Escoja **Aceptar**.

Cómo suprimir una tabla de la base de datos activa

1. Despliegue la lista **Tabla**.
2. Seleccione el valor que desee.
3. Haga clic en el botón **Suprimir la tabla**.

Cómo mostrar u ocultar los datos de la tabla activa

1. Escoja **Mostrar los datos**.

Configurar un campo como campo clave

1. Marque la casilla del campo deseado.

2. Escoja **Aplicar**.

Cómo configurar el tipo de contenido de un campo

1. Haga clic en la celda **Tipo** del campo deseado.
2. Haga clic en el icono de lista desplegable.
3. Seleccione el valor que desee.
4. Escoja **Aplicar**.

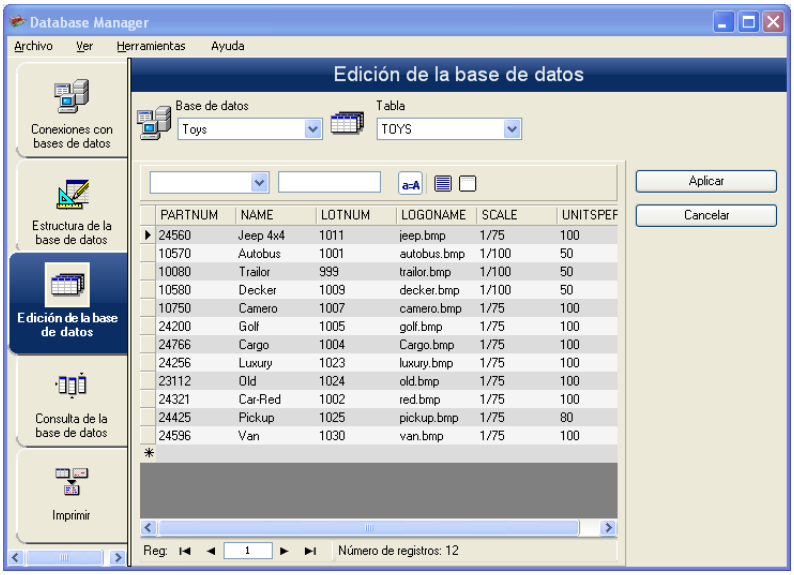
Cómo configurar el tamaño máximo de un campo

1. Haga clic en la celda **Longitud** del campo deseado.
2. Introduzca el valor que desee.
3. Escoja **Aplicar**.

Cómo permitir que un campo quede vacío

1. Haga clic en la celda **Permitir valores Null** del campo deseado.
2. Escoja **Aplicar**.

Interfaz de edición de bases de datos



Así se presenta la ventana mediante la cual se administra el contenido del archivo de bases de datos añadiendo, modificando o suprimiendo datos.

Las acciones posibles dependen del tipo de base de datos. Los registros de archivos en formato Excel, por ejemplo, no se puede modificar.

Cómo encontrar un registro que corresponda al contenido de un campo

1. Haga clic en el icono de lista desplegable.
2. Seleccione el valor que desee.
3. Haga clic en la zona de edición.
4. Introduzca el valor deseado en la zona de edición.

Cómo seleccionar todos los registros idénticos

El Database Manager debe estar abierto y mostrar la ventana de Edición de la base de datos con, al menos, un registro encontrado.

1. Haga clic en el icono de lista desplegable.
2. Seleccione el valor que desee.
3. Haga clic en la zona de edición.
4. Introduzca el valor deseado en la zona de edición.
5. Escoja **Seleccionar todo**.

Cómo seleccionar un registro idéntico

El Database Manager debe estar abierto y mostrar la ventana de Edición de la base de datos con, al menos, un registro encontrado y varios registros con idéntico contenido en el campo de búsqueda.

1. Para seleccionar el registro Primero, Anterior, Siguiente o Último haga clic respectivamente en los botones de la herramienta de búsqueda.

Cómo crear un registro nuevo

1. Haga clic en cualquier campo de la línea de edición, marcada con un asterisco.
2. Introduzca los valores que desee en los campos correspondientes.
3. Escoja **Aplicar**.

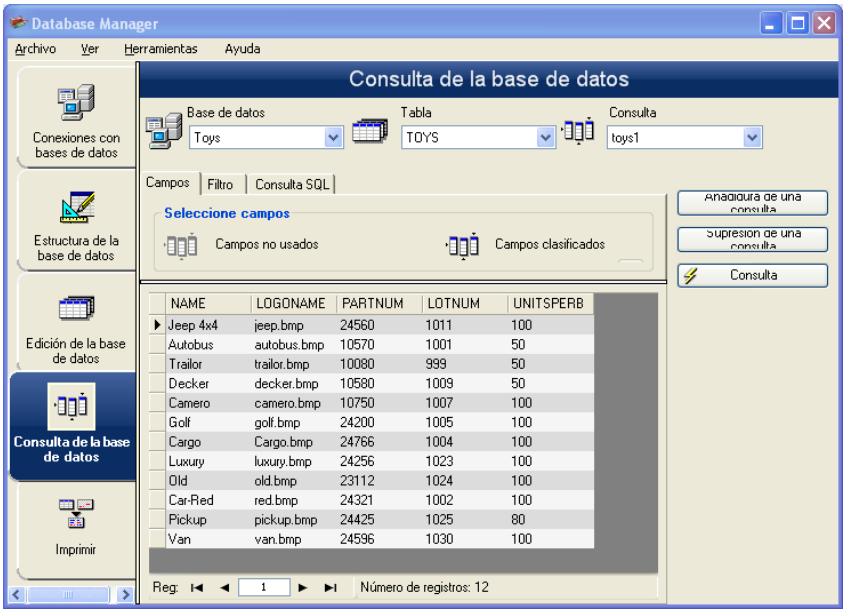
Cómo modificar un registro

1. Haga clic en el valor que desea modificar.
2. Introduzca el valor que desee.
3. Escoja **Aplicar**.

Cómo suprimir un registro

1. Haga clic en la casilla de selección del campo deseado.
2. Haga un clic con el botón derecho en la casilla de selección del campo deseado.
3. En el menú contextual, seleccione **Suprimir el registro seleccionado**.

Interfaz de consulta de bases de datos



Así se presenta la ventana que permite efectuar consultas, creando y aplicando diferentes filtros.

Cómo añadir una consulta

1. Escoja **Añadidura una consulta**.
2. Dé un nombre a la consulta.
3. Escoja **Aceptar**.

Cómo seleccionar o anular la selección de uno o más campos

1. Para seleccionar o anular la selección de uno o más campos, haga clic en .
2. Escoja **Consulta**.

Cómo modificar el orden de los campos seleccionados

1. Haga clic en el campo de la ventana **Campos ordenados** que desee.
2. Haga clic en el botón de flecha ascendente o descendente hasta llegar al valor deseado
3. Escoja **Aceptar**.

Cómo crear un filtro con datos pre definidos

El Database Manager debe estar abierto y mostrar la ficha "Filtro" de la ventana de Consultas de la base de datos.

1. Haga clic en el botón **Añadir línea** ( ).
2. Haga clic en el campo **Campo**.
3. Haga clic en el icono de lista desplegable.
4. Seleccione el valor que desee.
5. Haga clic en el campo **Operario**.
6. Haga clic en el icono de lista desplegable.
7. Seleccione el valor que desee.
8. Haga clic en el campo **Valor**.
9. Introduzca el valor que desee.
10. Escoja **Consulta**.

Cómo aplicar un operador lógico a varios filtros

Se deben haber creado varios filtros.

1. Haga clic en el campo **Lógico**.
2. Haga clic en el icono de lista desplegable.
3. Seleccione el valor que desee.
4. Haga clic en el botón **Consulta** para aplicar las modificaciones y ver el resultado.

Cómo suprimir un filtro

Se debe haber creado al menos un filtro.

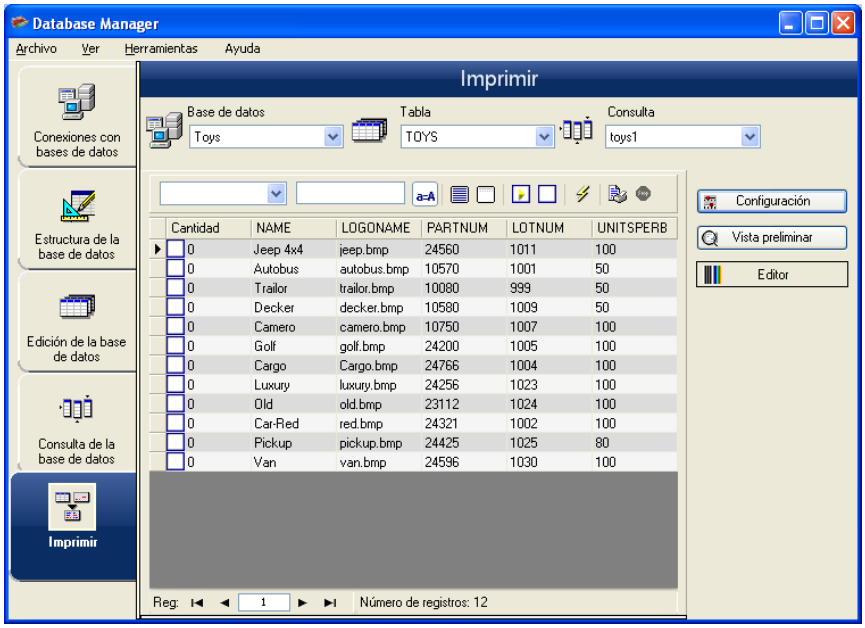
1. Haga clic en la casilla de selección del campo deseado.
2. Haga clic en el botón **Suprimir la línea**.

Cómo modificar un filtro escrito en lenguaje SQL

Se debe haber creado al menos un filtro.

1. Escoja la opción **Modificar la consulta efectuada en lenguaje SQL** para activar la consulta mediante el lenguaje SQL y poder modificarla manualmente.
2. Escoja **Consulta**.

Interfaz de impresión



Así se presenta la ventana que permite seleccionar el archivo a imprimir, atribuirle una impresora y configurar la impresión antes de ejecutarla.

Cómo crear un modelo nuevo de etiqueta a imprimir

1. Haga clic en el botón **Asistente de creación de etiquetas**.
2. Siga las instrucciones que le da el asistente.

Cómo seleccionar un modelo de etiqueta ya existente

1. Haga clic en el botón **Abrir un documento** ya existente.
2. Seleccione un archivo de tipo .lab.
3. Escoja **Aceptar**.

Cómo seleccionar la impresora a utilizar

1. Haga clic en el botón **Añadir o suprimir una impresora** ( ).
2. Seleccione la impresora que desee.
3. Escoja **Aceptar**.

Cómo configurar la impresora seleccionada

Se deben haber seleccionado un modelo de etiqueta y una impresora.

1. Haga clic en el botón de Configuración de la impresora ( ).
2. Efectúe la configuración que desee.
3. Escoja **Aceptar**.

Cómo seleccionar manualmente los registros a imprimir

1. Marque las casillas de los registros deseados.

Cómo seleccionar automáticamente los registros a imprimir

1. Haga clic en el icono de lista desplegable.
2. Seleccione el valor que desee.
3. Haga clic en la zona de edición.
4. Introduzca el valor que desee.
5. Escoja **Seleccionar todo**.
6. Haga clic en el botón **Comprobar**.

Fórmula

Fuente de datos Fórmula

La fuente de datos **Fórmula** contiene una lista de variables que se van creando a medida que se necesitan. Estas variables se van rellenando con los datos resultantes de una combinación de operadores, constantes, variables, variables de control, fórmulas y funciones. Los datos pueden ser de tipo numérico o alfanumérico.

Funciones

Nota: la configuración regional del sistema Windows puede afectar a los resultados de algunas fórmulas.

Una función es una fórmula predeterminada que efectúa cálculos utilizando valores especiales, llamados argumentos, presentados en un orden determinado, llamado sintaxis.

Las funciones se utilizan para obtener resultados de tipo numérico, textual o lógico tras efectuar cálculos u operaciones.

Para construir fórmulas se emplean tipos de funciones:

- Funciones de cálculo de caracteres de control (check digits)
- Funciones de conversión
    - Funciones de conversión ATA 2000
- Funciones relativas a fechas y horas
- Funciones lógicas
- Funciones matemáticas
- Funciones relativas a cadenas de texto

Operadores

La aplicación acepta operadores aritméticos, comparativos, de concatenación y lógicos.

Operadores aritméticos

Operador	Se utiliza para
*	Multiplicar dos números
+	Sumar números
-	Restar un número de otro o indicar el valor negativo de un operando
/	Dividir un número por otro
^	Elevar un número a la potencia del exponente
%	Modulo

Nota: algunas fórmulas no devuelven un error cuando normalmente se esperaría que lo hicieran. Algunos ejemplos son: 1/0, 0/0, 0,0/0,0, 1/000 y 0^(-2). Esto es intencional para evitar que el proceso de impresión se bloquee. El usuario debe definir el comportamiento deseado añadiendo controles adicionales si es necesario.

Operadores de comparación

Operador	Significado
<	Inferior a
<=	Inferior o igual a
>	Superior a
>=	Superior o igual a
=	Igual a
	Diferente de

Operador de concatenación

Se utiliza para enlazar dos cadenas.

Operador	Significado
&	Concatenación de dos cadenas

Operador lógico

(consulte también las funciones lógicas)

Operador	Significado
!	No lógico

Funciones de cálculo de caracteres de control

Nota: la configuración regional del sistema Windows puede afectar a los resultados de algunas fórmulas.

addmodulo10 («cadena»): devuelve el valor entero, con el carácter de control Modulo 10 añadido al final.

addmodulo10_212(«cadena»): devuelve un carácter de control de tipo modulo 10_212 (una variante del carácter de control estándar de tipo modulo 10) e incluye la cadena de texto activa, añadiendo al final la suma de control.

addmodulo43(«cadena»): devuelve un carácter de control de tipo modulo 43 e incluye la cadena de texto activa, añadiendo al final el carácter de control.

bimodulo11 («cadena»): devuelve dos caracteres de control de tipo modulo 11 (código 11).

canadacustomscd (*cadena*): devuelve el carácter de control del Canada Customs Standard.

Check103(«string»): Esta función calcula Mod103.

check128 («cadena»): devuelve el carácter de control de los códigos 128 y EAN 128.

checkPZN(«string»): Calcula el carácter de control para el código de barras PZN. El código de barras PZN se basa en el código 39 y se utiliza para los productos farmacéuticos en Alemania.

CheckSum2Mod47*(«string»)*: Esta función calcula Mod47.

CheckSumMod10*(«string»)*: Esta función calcula Mod10.

CheckSumMod10_Codabar*(«string»)*: Esta función calcula la suma de comprobación específica de Mod10 para Codabar.

CheckSumMod10_MSI*(«string»)*: Esta función calcula la suma de comprobación de Mod10 para el código de barras MSI.

CheckSumMod11_3Suisse*(«string»)*: Esta función calcula la suma de comprobación de Mod11 para 3Suisses.

CheckSumMod34*(«string»)*: Esta función calcula Mod34.

checkupce (*cadena*): devuelve el carácter del control de UPC-E (6 caracteres obligatorios).

CRC16(«string»): Devuelve el valor CRC-16.

Ejemplo:
CRC16("123456789") = 47933
"0x" & hex(CRC16("123456789"), 4) = 0xBB3D

modulo10 («valor»): devuelve un carácter de control de tipo modulo 10 para los códigos 2/5 Interleaved, EAN13, EAN 8, EAN 128 K-Mart y VPCA.

modulo10_212(«cadena») devuelve una variante del carácter de control de tipo modulo 10.

modulo10IBM(«cadena»): devuelve un carácter de control de tipo IBM modulo 10 (una variante del carácter de control estándar de tipo modulo 10).

modulo10UPS(«cadena»): devuelve un carácter de control de tipo UPS modulo 10 (una variante del carácter de control estándar de tipo modulo 10).

modulo11 («cadena»): devuelve un carácter de control de tipo modulo 11 (código 11).

modulo11IBM(«cadena»): devuelve un carácter de control de tipo modulo 11 IBM (una variante del carácter de control estándar de tipo modulo 11).

modulo16(«cadena») : devuelve un carácter de control de tipo modulo 16.

modulo24 («cadena»): devuelve un carácter de control de tipo modulo 24 (código PSA).

modulo32 («valor»): devuelve un carácter de control de tipo modulo 32 (industria farmacéutica italiana).

modulo43 («cadena»): devuelve un carácter de control de tipo modulo 43 (código 39).

modulo47 («cadena»): devuelve dos caracteres de control de tipo modulo 47 (código 93).

Plessey («cadena»): devuelve dos caracteres de control (código Plessey).

pricecd (*cadena*): devuelve 4 caracteres de control UPC Random Price.

pricecd5 (*cadena*): devuelve 5 caracteres de control UPC Random Price.

StringToExt39*(«string»)*: Esta función prepara la cadena que se va a usar en el código de barras Extended 39.

UPSCheckDigit (*cadena*): Devuelve los caracteres de comprobación de UPS. La entrada para el método es una cadena. Este método deja el resto del número de seguimiento a su elección.

Toma una secuencia de 15 caracteres y calcula el dígito de comprobación mediante esta secuencia.

1. Los primeros dos caracteres deben ser "1Z".
2. Los siguientes 6 caracteres los completaremos con nuestro número de cuenta UPS "XXXXXX".
3. Los siguientes 2 caracteres indican el tipo de servicio:

   "01" para envíos aéreos al siguiente día.
   "02" para envíos aéreos al segundo día.
   "03" para envíos terrestres.

4. Los siguientes 5 caracteres es nuestro número de factura (nuestras facturas tienen 6 dígitos; omitimos el primer dígito, p. ej., la factura 123456 produciría los caracteres 23456).
5. Los siguientes 2 dígitos es el número de paquete con un cero delante. P. ej., paquete 1 es "01", 2 es "02".
6. El carácter último y final es el dígito de comprobación.

Nota: La secuencia descrita anteriormente es de 17 caracteres, de los cuales solo se necesitan 15 para calcular el dígito de comprobación. Para ello, omita la parte "1Z" y utilice solo los últimos 15 caracteres del método.

Funciones de conversión

Nota: la configuración regional del sistema Windows puede afectar a los resultados de algunas fórmulas.

AI253: Función específica para preparar la cadena para el identificador 253 de la aplicación.

AI8003: Función específica para preparar la cadena para el identificador 8003 de la aplicación.

ASCII («cadena»): devuelve el código ASCII del primer carácter del argumento cadena.

Ejemplo:
ASCII("A") = 65

char («integer»)

- Para los valores de 128 a 255: devuelve el carácter correspondiente al código ASCII de *"integer".*
- Para todos los valores negativos y los valores entre 0 y 127 y más de 255 devuelve el carácter Unicode correspondiente al "integer".

Ejemplo:
char (65) = "A"

CodabarData(«data», «start», «stop»): adjusta datos para CodabarData.

Code128CData(«cadena»): adjusta datos para Code 128 C.

Currencytoeuro («valor»): convierte la divisa empleada en euros.

DatabarData(«data», «min», «max», «hasComposite»): adjusta datos para DatabarData.

DBCSToUnicode(«string»,«codepage»): La fórmula afecta a 2 parámetros - se convierten los datos y se utiliza la página de códigos. El parámetro «codepage» puede ser uno de los siguientes idiomas (en inglès): Thai, Japanese, ChineseGBK, Korean, ChineseBig5, European, Eastern, Cyrillic, Greek, Turkish, Hebrew, Arabic, Baltic, Vietnamese, UTF-8, ACP) o un valor numérico que corresponde a la página de códigos de la lengua.

Ejemplo:
DBCSToUnicode(unicodetoDBCS("傍傍傍傍", "UTF-8"), "UTF-8") =傍傍傍傍

Observación: Esta función es el contrario de la función UnicodetoDBCS.

dollar («valor»): modifica un campo para presentar un precio.

Ejemplo:
Si un campo llamado PRICE tiene como valor 199, entonces…
dollar(PRICE) presentará $199,00.

Ean128Data («cadena_1»,«cadena_2»): Adjust "string_1" data for EAN 128 barcode according to template "string_2"

Eurotocurrency («valor»): convierte los seguros en la divisa empleada.

Observación: el número de decimales se debe configurar en la ficha **Salida** de la variable **Fórmula**, activando la opción **Mostrar los decimales**. El tipo de conversión se configuran mediante la ficha **Otras** del cuadro de diálogo **Opciones**.

FileToData(«fileName», «errorData», «maxSize»): devuelve los datos leídos del archivo.

fixed («valor», «n°_decimales», «sin_sep») devuelve una cadena de caracteres compuesta por el número de decimales indicado en *n°_decimales*, con o sin separador de millares.

Ejemplos:
fixed (1234.5678, 3, TRUE) = "1234.568"
fixed (1234.5678, 3, FALSE) = "1,234.567"

Observación: el separador de millares se debe configurar en la ficha **Salida** de la variable **Fórmula**, activando la opción **Mostrar los decimales**.

FormatDate(«date», «dateFormat», «localeID»): convierte «date» en una cadena formateada según «dateFormat».

Valor de retorno	cadena	Un valor fecha en formato cadena, formateado según un formato.
date	cadena	Valor de fecha base en formato cadena.
dateFormat	cadena / Número	Formato de valor de fecha base. - Los números codifican formatos predefinidos. (P. ej. 1- dd/mm, 2- dd mmmm etc.). Los valores posibles están disponibles aquí. - 0 ó valores negativos significan el empleo de formato de fecha del sistema actua.l - El valor de cadena codifica el formato directamente. (P. ej. "aaaa/mm/dd")
localeID	Número	Opcional (por defecto 0, el idioma del sistema está utilizado) Este parámetro especifica el idioma utilizado para mostrar el valor de la fecha.

FormatMoney(«amount», «use separation characters(T/F)», «price characters», «force leading zaro (T/F)», «location for price character», «# of decimal places»): Esta función le permite añadir el símbolo de la divisa, puntos, forzar el cero inicial, usar caracteres de separación

Ejemplo:

FormatMoney("123456.234", "T","$","F",0,2) = $123,456.23

FormatMoney("1234","F","$","T",8,2) = $1234.00

FormatMoney("0.234","F","$","F",0,2) = $.23

FormatMoney("0.234","F","$","T",0,2) = $0.23

Nota: la ubicación del carácter del precio es responsable de la posición del carácter del precio, los espacios se agregan entre el carácter del precio y la cantidad. Este valor se ignora si es menor que la extensión de la cadena de resultado.

FormatNumber(number): permite formatear un campo numérico donde el símbolo almohadilla (#) significa que solo se muestra si hay un valor y el cero (0) significa que se muestra siempre.

Ejemplos:

FormatNumber(123.45, "US$ #,###,###.00") = US$ 123.45

FormatNumber(123.45, "US$ 0,000,000.00") = US$ 0,000,123.45

FormatNumber(.45, "#,##0.00") = 0.45

FormatNumber(.45, "#,###.00") = 45

FormatNumber(7188302335, "(###) ###-####") = (718) 830-2335

FormatNumber(123.45, "00.00") = 23.45

FormatNumber(123.567, "###,##0.00") = 123.57

GetEnv(«name»): Devuelve el valor «name» (cadena) de la variable del entorno.

Ejemplo:

GetEnv("OS") = " Windows_NT".

Observación: Esta función funciona con las variables del entorno definidas por el usuario

GS1AIData(«AIName», «AIData», «isLastAI»): adjusta datos para GS1 AI.

GS1HRData(«AIName», «AIData», «calcCheckDigit»): Adjusts data for GS1 Human readable.

GS1Norm («string»): Adjusts data for GS1 Norm.

int (valor) devuelve el número entero más grande cuyo valor es inferior o igual al argumento *valor*.

Ejemplos:

int (-5.863) = -6

int (5.863) = 5

LmChar («integer») : Devuelve el carácter correspondiente al argumento de número entero en la tabla ANSI. Esta tabla no depende del sistema de configuración regional.
El valor se ubica entre 0 y 255.

MaxiCodeData(«Mode»,«PostalCode»,«CountryCode»,«ClassOfService»,«TrackingNumber»,«UpsShipperNumber»,«JulianDate»,«ShipmentID»,«PackageNumber»,«TotalPackages»,«PackageWeight»,«AdressValidation»,«ShipToAdress»,«ShipToCity»,«ShipToState»): Crea datos maxicode de salida **«string».**

text («valor», «formato») devuelve el argumento *valor* presentado conforme al argumento *formato*, que corresponde al formato ajustado para la variable Formulario.

Ejemplos:

text(012.345, "######,(###)") = 12,(345)

text(1012.345, "######,(###)") = 1012,(345)

trunc («valor») devuelve la parte entera del argumento *valor*.

Ejemplo:

trunc (123.45) = 123

unicodetoDBCS(«Datos», «Página de códigos»): La fórmula afecta a 2 parámetros - se convierten los datos y se utiliza la página de códigos. El parámetro de la página de códigos puede tener uno de los siguientes valores:

Thai

Japanese

Chinese GBK

Korean

Chinese Big5

European eastern

Greek

Turkish

Hebrew

Arabic

Baltic

Vietnamese

value («cadena») devuelve el valor numérico de una cadena de caracteres.

Ejemplo:

value("123") = 123

value("16:48:00")-value("12:00:00") = "16:48:00"-"12:00:00" = 0,00, el número seriado equivalente a 4 horas y 48 minutos.

Observación: no suele ser necesario utilizar la función **value** en una fórmula, ya que la aplicación convierte automáticamente el texto en números cuando es necesario.

ValueEx(«string»): Convierte «string» en un valor numérico.

VoiceCode(«string», «string», «string»): Un VoiceCode es un número de 4 dígitos computado utilizando el GTIN, Lot y Date opcional de un PTI.

Return value	**string**	**Calcula un valor de VoiceCode.**
GTIN	string	Número GTIN de 14 dígitos. Solo se permiten dígitos. Se devolverá error en caso de un valor que no sea un dígito. Se aceptan 14 dígitos en la parte izquierda en caso de que la longitud de los datos sea mayor de 14. '0' se agregan a la izquierda en caso de que la longitud de los datos sea menor de 14 dígitos.
LotNumber	string	Datos de hasta 20 símbolos alfanuméricos. Se aceptan 20 símbolos en la parte izquierda en caso de que la longitud de los datos sea mayor de 20 dígitos.
Date	string	Este es un parámetro opcional. Datos de 6 dígitos que representan la fecha en formato YYMMDD. La fórmula devuelve error en caso de longitud incorrecta, valor que no sea un dígito o fecha incorrecta.

Esta computación se realiza como sigue:

1. Computar PlainText.

   a. PlainText es el GTIN de 14 dígitos anexado por el Lot Code y el Date (donde esté presente).

b. No incluya los prefijos de identificador de aplicación ni paréntesis.

c. No existen espacios entre los campos GTIN, Lot y Date.

d. Date, si está presente, está representado por YYMMDD sin cadenas ni '/' caracteres.

2. Computar el hash ANSI CRC-16 de los bytes de PlainText ASCII mediante el hash ANSI CRC-16 estándar con el polinomial de X16 + X15 + X2 + 1
3. Computar el VoiceCode del hash tomando los 4 últimos dígitos significantes en forma decimal (hash mod 10000).

Ejemplo:

GTIN = (01) 10850510002011
Lot = (10) 46587443HG234
PlainText = 1085051000201146587443HG234
CRC-16 Hash = 26359
VoiceCode = 6359
Dígitos largos = 59
Dígitos cortos = 63

Ejemplos:

VoiceCode("10850510002011", "46587443HG234") = 6359
VoiceCode("65457886676767", "2", "100126") = 5836

Funciones de conversión ATA 2000

Nota: la configuración regional del sistema Windows puede afectar a los resultados de algunas fórmulas.

Bin2Hex(«value», «pad»)

Convierte un valor decimal en un valor de salida hexadecimal. El parámetro "pad" se utiliza para definir la longitud de toda la salida añadiendo los caracteres 0 a la izquierda del resultado.

Ejemplo:
Bin2Hex("0010100111101001101",0) devuelve 14F4D
Bin2Hex("0010100111101001101",0) devuelve 00014F4D

CRC16_CCITT(«string»)

Calcula CRC16 CCITT suma de comprobación

Se utiliza en la normativa ATA2000 para el cálculo de TOC para CRC.

Ejemplo:

CRC16_CCITT ("A") devuelve B915

CRC16_CCITT ("AB") devuelve 4B74

CRC16_CCITT ("ABC") devuelve F508

CRC32_CCITT(«string»)

Calcula CRC32 CCITT suma de comprobación

Se utiliza en la normativa ATA2000 para el cálculo de TOC para CRC.

Ejemplo:

CRC32_CCITT ("A") devuelve D3D99E8B

CRC32_CCITT ("AB") devuelve 30694C07

CRC32_CCITT ("123456789") devuelve CBF43926

Dec2Bin(«value», «pad»)

Convierte un valor decimal en un valor de salida binario. El parámetro "pad" se utiliza para definir la longitud de toda la salida añadiendo caracteres 0 a la izquierda del resultado.

Ejemplo:

Dec2Bin("5", 0) devuelve 101

Dec2Bin("5", 4) devuelve 0101

Dec2Hex(«value», «pad»)

Convierte un valor decimal en un valor de salida hexadecimal. El parámetro "pad" se utiliza para definir la longitud de toda la salida añadiendo los caracteres 0 a la izquierda del resultado.

Ejemplo:

Dec2Hex(510, 0) devuelve 1FE

Dec2Hex(510, 4) devuelve 01FE

String2Hex(«value», «pad»)

Convierte un valor cadena en un valor de salida hexadecimal. El parámetro "pad" se utiliza para definir la alineación en Bytes para la salida añadiendo caracteres 0 a la derecha del resultado.

En RFTAG, es muy frecuente que los datos estén encriptados en hexadecimal con una alineación de palabra (2 Bytes).

Ejemplo:
String2Hex("A", 0) devuelve 41 (**A** es código **65** AINSI en decimal, que es **41** en hexadecimal)
String2Hex("ABC", 2) devuelve 414243
String2Hex("ABC", 4) devuelve 41424300

Este es un ejemplo de un registro de nacimiento RFID ATA2000:

String2Hex("MFR S0671*SEQ M37GXB92*PNO PQ7VZ4*PDT CONTROLLER*ICC 456789*UIC 2*DMF 20160701*UNT KG*HAZ UN0003*ESD 1*LOT 123456*CNT FR*PMLPML123456789", 2) devuelve
4D46522053303637312A534551204D333747584239322A504E4F20505137565A342A504454204
34F4E54524F4C4C45522A49

String6BitsEncoding(«value», «pad»)

Convierte un valor cadena en un valor de salida encriptado ASCII 6-Bit.
El parámetro "pad" se utiliza para definir la alineación en Bytes para la salida añadiendo caracteres 0 a la derecha del resultado.
En ATA2000, los datos se pueden encriptar en 6 bits (en lugar de 8 bits) con la finalidad de comprimirlos.

Ejemplo:
String6BitsEncoding("11", 0) devuelve C710

VDAString6BitsEncoding(«cadena»)

Convierte una cadena de valor en un valor de salida de codificación ASCII de 6 bits utilizando los requisitos de la Asociación Alemana de la Industria del Automóvil (VDA).
En ATA2000 los datos pueden codificarse en 6 bits (en lugar de 8 bits) con fines de compresión.
La tabla utilizada para la conversión está disponible **aquí**. Para más detalles, consulte las especificaciones VDA5500 y VDA4994.

Ejemplo:
VDAString6BitsEncoding("11") devuelve C71860820820

Funciones relativas a fechas y horas

La variable fecha aquí tratada se refiere a la fecha y hora del sistema y no a la variable fecha ajustada por el programa.

Nota: la configuración regional del sistema Windows puede afectar a los resultados de algunas fórmulas.

BestBefore(«date» , «dateFormat», «offset» , «offsetUnit» , «changeMonth» , «outputFormat» , «localeID»)

Le permite calcular la fecha de consumo preferente basada en el valor de entrada del teclado (y no en la fecha actual).

Valor de retorno	cadena	Un valor de fecha en formato cadena, calculado por una fecha base y un desplazamiento.
date	cadena	Valor de fecha base en formato cadena.
dateFormat	cadena / Número	Formato de valor de fecha base. - Los números codifican formatos predefinidos. (P. ej. 1- dd/mm, 2- dd mmmm etc.). Los valores posibles están disponibles aquí. - 0 ó valores negativos significan el empleo de formato de fecha del sistema actua.l - El valor de cadena codifica el formato directamente. (P. ej. "aaaa/mm/dd")
offset	Número	Número de unidades de fecha para añadir a la fecha base.
offsetUnit	cadena / Número	Significado del parámetro offset. - **1** ó **"d"** ó **"D"** significa días. - **2** ó **"m"** ó **"M"** significa meses. - **3** ó **"y"** ó **"Y"** significa años.
changeMonth	cadena	Opcional (predeterminado: True) - **0** significa **False.** - **1** significa **True.** Si la fecha calculada no existe en ese mes (p. ej. 30 de febrero). **True** remite al primer día del mes siguiente (p. ej. 1.er de marzo). **False** remite al último día del mes actual (p. ej. 28 de febrero).

outputFormat	cadena / Número	Opcional (predeterminado: como en *dateFormat*) Este parámetro especifica el formato de salida de la fecha calculada. Los valores posibles son los mismos que en el caso de dateFormat.
localeID	Número	Opcional (por defecto 0, el idioma del sistema está utilizado) Este parámetro especifica el idioma utilizado para mostrar el valor de la fecha.

Ejemplo:

BestBefore («*date*» , «dateFormat» , «offset» , «offsetUnit» , «changeMonth» , «outputFormat»)

BestBefore("14/06/2012" ,"dd/mm/yyyy","18","m", 1, "dd/mm/yyyy") = 14/12/2013.

Ejemplo: Formula con una variable fecha.

Cree una variable fecha 'date0' con la fecha actual (dd/mm/yy formato). La fecha actual para el ejemplo es 26/06/2012.

BestBefore(date0 ,"dd/mm/yy","18","m", 1, "dd/mm/yyyy") = 14/12/2013.

CFIA_Month(«fecha»):

Devuelve el nombre del mes utilizado por la Agencia Canadiense de Inspección Alimentaria (CFIA): JA (ENE), FE (FEB), MR (MAR), AL (ABR), MA (MAY), JN (JUN), JL (JUL), AU (AGO), SE (SET), OC (OCT), NO (NOV), DE (DIC).

Ejemplos:

CFIA_Month ("03/01/2021") devuelve MR (MAR)

CFIA_Month (today()) devuelve el nombre del mes en función de la fecha de hoy, si hoy es 01/01/2021, el valor devuelto es JA (ENE)

CFIA_Month ("06/01") devuelve JN (JUN)

Nota: el formato del argumento "fecha" depende de la configuración del sistema.

DateOffset(«date» , «offset», «offsetUnit» , «changeMonth»)

Return value	date	Añada un intervalo de fechas a la fecha base especificada.
date	date	Valor de fecha base.
offset	number	Número de unidades de fecha para añadirse a la fecha base.

offsetUnit	string / number	Significado del parámetro offset. - **1** ó **"d"** ó **"D"** significa días. - **2** ó **"m"** ó **"M"** significa meses. - **3** ó **"y"** ó **"Y"** significa años.
changeMonth	number	Opcional (predeterminado: True) - **0** significa **False.** - **1** significa **True.** Si la fecha calculada no existe en ese mes (p. ej. 30 de febrero) **True** remite al primer día del mes siguiente (p. ej. 1.er de marzo). **False** remite al último día del mes actual (p. ej. 28 de febrero).

Ejemplo:

DateOffset("26/06/2012",1,"d",1) = 27/06/2012

-O-

DateOffset(today(),1,"D", 0)

Ejemplo: Formula con una variable fecha.

Cree una variable fecha 'date0' con la fecha actual (dd/mm/yy formato). La fecha actual para el ejemplo es 26/06/2012.

DateOffset(Date0,1,"D", 0) = 27/06/2012

DateValue(«formattedDate» , «format»)

Return value	date	Vuelve al valor de fecha del parámetro *formattedDate*.
formattedDate	string	Una fecha de formato texto para ser convertida en valor de fecha.
format	string	Opcional (predeterminado: el formato de fecha del sistema) Si se ha especificado, la cadena de formato directo. P. ej. "dd/mm/aa"

Ejemplo:

DateValue(22062012,"ddmmyyyy") = 22/06/2012.

day («fecha») devuelve el día del mes del argumento *fecha*.

FiscalDate(«fiscalStartDate», «outputFormat»)

Le permite calcular la fecha de inicio del año fiscal (yyyy.mm.dd).

Ejemplos (si la fecha actual es 2009.11.24):

FiscalDate("2009.01.01", 1) = 09

FiscalDate("2009.01.01", "yyyy") = 2009

FiscalDate("2009.01.01", 3) = 47

FiscalDate("2009.01.01", "day") = 328

hour («fecha») devuelve la hora del argumento *fecha*.

minute («fecha») devuelve los minutos del argumento *fecha*.

month («fecha») devuelve el mes del argumento *fecha*.

now () devuelve la fecha y hora corrientes.

second («fecha»): Proporciona el segundo del argumento de fecha.

shiftcode(«items»)

El Shift code se utiliza como una fuente de datos para un campo en la etiqueta que tiene que cambiar, basándose en la hora actual. Un cambio representa un espacio de tiempo que está definido y tiene un nombre.

valor de retorno	cadena	Devuelve el valor de shift code actual.
item (elemento)	cadena	Elementos de shift code en el siguiente formato: inicioHora:inicioMin-finHora:FinMin-valor|…| inicioHora:inicioMin-finHora:FinMin-valor

Por ejemplo, los cambios se podrían definir como aparece a continuación:

- 7:00 a 15:00 = Día
- 15:00 a 23:00 = Tarde
- 23:00 a 7:00 = Noche

Notas:

1. El reloj de 24 horas se utiliza , para definir valores de tiempo. 0:00 es medianoche; 12:00 es mediodía.

2. Si hay alguna coincidencia de hora en la entrada, entonces el primer valor aceptable se devuelve.

3. Si no hay ningún cambio para la hora actual, entonces se devuelve una cadena vacía.

Ejemplo (si la hora actual es 16:00):

shiftcode("7:00-15:00-Día|15:00-23:00-Tarde|23:00-7:00-Noche") ="Tarde"

SpecificDateFormat("[date format]", "+/[offset data][date interval]")

La función FormatoDeFechaEspecífico le permite personalizar una impresión de fecha, desplazando la fecha en un intervalo determinado, y usar esta fecha personalizada en una expresión de fórmula. La función FormatoDeFechaEspecífico desplaza la fecha actual basándose en el reloj del sistema, usando el formato de fecha predeterminado o un formato diferente de su elección.

Cuando se utiliza en una expresión, la función FormatoDeFechaEspecífico debe escribirse usando los siguientes parámetros:
SpecificDateFormat("[date format]", "+/[offset data][date interval]")

Ejemplo : SpecificDateFormat("mmmm","+2M")

- [date format] Especifica el formato de fecha que debe usarse. Este parámetro debe incluirse entre paréntesis. Vea la ***tabla*** de formatos de fecha válidos.

Configuración de fecha	Salida	Descripción
D y d	1	fecha del mes (sin cero inicial)
DD	01	fecha del mes con 2 caracteres (cero inicial)
dd	1	2 char date of month (lead space)
DDD	224	fecha del mes con 2 caracteres (espacio inicial)
ddd	224	días desde el comienzo del año (cero inicial)
DDDDD y ddddd	33482	days from 1/1/1900 (5 to 9 chars available)
W y w	1	días desde el comienzo del año (espacio inicial)
WW	34	días desde 1/1/1900 (de 5 a 9 caracteres disponibles)
ww	34	semanas desde el comienzo del año (espacio inicial)
WWW y www	783	semanas desde 1/1/1900 (3 últimos dígitos)
WWWW y wwww	4783	semanas desde 1/1/1900 (de 4 a 9 caracteres disponibles)

WWWWWWWWWW	000004783	semanas desde 1/1/1900 (cero inicial)
wwwwwwwww	4783	semanas desde 1/1/1900 (espacio inicial)
M y m	9	dígitos del mes (sin cero inicial)
MM	09	dígito del mes de 2 caracteres (cero inicial)
mm	9	dígito del mes de 2 caracteres (espacio inicial)
MMM	009	dígito del mes de 3 caracteres (cero inicial)
mmm	9	dígito del mes de 3 caracteres (espacio inicial)
MMMM y mmmm	1101	meses desde 1/1/1900
MMMMM	01101	meses de 5 caracteres desde 1900 (cero inicial)
mmmmm	1101	meses de 5 caracteres desde 1900 (espacio inicial) (de 5 a 9 caracteres disponibles)
MMMMMMMMM	000001101	meses de 9 caracteres desde 1900 (cero inicial)
mmmmmmmmm	1101	meses de 9 caracteres desde 1900 (espacio inicial)
YY y yy	91	año de 2 dígitos
Y y y	1991	año completo
YYY y yyy	991	últimos 3 dígitos del año (de 3 a 9 caracteres disponibles)
YYYYYYYYY	000001991	año completo (cero inicial)
yyyyyyyyy	1991	año completo (espacio inicial)

- +/[offset data] Este valor especifica la cantidad en la que se desplazará la fecha. El intervalo debe ir precedido por un signo de más (+) o de menos (), dependiendo de si desea agregar o quitar de la fecha actual.
- [date interval] El intervalo de fecha debe ser d para días, s para semanas, m para meses o a para años.

La siguiente tabla muestra ejemplos de la función FormatoDeFechaEspecífico usado en expresiones con diferentes formatos. (Nota: Para estos ejemplos, suponemos que la fecha actual es 29 de julio de 2013)

SpecificDateFormat("mm dd yy" , "+1y") = 7 29 14 (1 año después de la fecha actual)
SpecificDateFormat("ww/m/yyyy" , "30w) = 1/12/2012 (30 semanas antes de la fecha actual)
SpecificDateFormat("ddddd / wwww /mmmm" , "") = 41484/5928/1363 (días, semanas y meses desde 1900)

TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"time interval +/offset time")

La función DesplazamientoDeHora le permite personalizar una impresión de hora, desplazando la hora en un intervalo determinado, y usar esta hora personalizada en una expresión de fórmula. La función DesplazamientoDeHora desplaza la hora actual basándose en el reloj del sistema, usando el formato de hora predeterminado.

Cuando se utiliza en una expresión, la función DesplazamientoDeHora debe escribirse usando los siguientes parámetros:

TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"intervalo de tiempo +/ hora de desplazamiento")

Ejemplo:TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"h+5")

- Puede usar solo el formato predeterminado de "dd/mm/aaaa hh:nn:ss". Para introducir un formato diferente, tiene que ajustar el DesplazamientoDeHora con la función ValorDeFecha y ajustarlo después en la función FechaDeFormato, donde puede especificar otro formato de hora.
- "+/offset time" Este valor especifica la cantidad en la que se desplazará la hora. El desplazamiento debe ir precedido por un signo de más (+) o de menos (), dependiendo de si desea agregar o quitar de la hora actual.
- "time interval" El intervalo de tiempo debe ser s para segundos, m para minutos o h para horas.

La siguiente tabla muestra ejemplos de la función DesplazamientoDeHora usado en expresiones con diferentes formatos. (Nota: Para estos ejemplos, suponemos que la hora actual es 4:12:10 PM del 30 de julio de 2013)

TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"h+5") = 07/30/2013 21:12:10 (5 horas después de la hora actual)
FormatDate(DateValue(TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"), "n+10"), "mm/dd/yyyy hh:nn:ss"), "hhnn") = 16-22 (10 minutos después de la hora actual)

today () devuelve la fecha corriente.

Week («fecha») devuelve el número de la semana del argumento *fecha*.

weekday («fecha») devuelve el día de la semana del argumento *fecha*.

Observación: se considera que el primer día de la semana es el domingo.

Ejemplo:
para el "martes, 20 de noviembre de 1999", se obtiene la fecha corriente weekday(now()) = 3

WeekISO8601(«Date», «DateFormat»)

Permite crear una fórmula que admita la normativa internacional ISO8601, que regula el intercambio de datos de fecha y hora.

Return value	date	Proporciona la semana de la fecha especificada.
Date	string	Una fecha con formato de texto, que se convierte en un valor de fecha.
DateFormat	string	Formato del valor de la fecha base. - Formatos predefinidos de codificación de números. (p. el. 1 dd/mm/aa, 2 dd/mm/aaaa, y así sucesivamente) - El 0 o los valores negativos significan que se está utilizando el formato de fechas del sistema actual - El valor de String codifica el formato directamente. (p. ej. "aaaa/mm/dd")

Ejemplo:

WeekISO8601(" 03/01/2010 ",0) = 53
WeekISO8601(" 02/01/2011 "," dd/mm/yyyy ") = 52
WeekISO8601(" 01/01/2011 ", " dd/mm/yyyy ") = 52

year («fecha») devuelve el año del argumento fecha.

E**jemplo**:
minute (now ()) devuelve los minutos de la hora corriente.
year (today ()) devuelve el año de la fecha corriente.

Funciones lógicas

Las funciones lógicas permiten comprobar si una o varias condiciones son verdaderas o falsas.

Nota: la configuración regional del sistema Windows puede afectar a los resultados de algunas fórmulas.

Observación: TRUE equivale a 1 y FALSE a 0

and («expr_1», «expr_2») da como resultado TRUE cuando los dos argumentos son verdaderos y FALSE cuando al menos uno de los dos argumentos es falso. Los argumentos deben poder interpretarse como valores lógicos.

Por ejemplo:
and(exact("cadena","cadena"),exact("cadena","cadena")) = 0
and(exact("cadena","cadena"),exact("cadena","cadena")) = 1

exact («cadena_1», «cadena_2») devuelve TRUE cuando ambas cadenas son idénticas y FALSE de lo contrario. Esta función toma en cuenta las mayúsculas y las minúsculas.

Por ejemplo:
exact("aplicación","aplicación") = 1
exact("apicación","aplicación") = 0

if («expr», «val_si_verdadero», «val_si_falso») devuelve el argumento *val_si_verdadero* cuando *Expr* es verdadera y *Val_si_falso* cuando *Expr* es falsa.

Por ejemplo:
if(exact("c adena", "cadena"), "verdadero", "falso")) = falso
if(exact("cadena", "cadena"), "verdadero", "falso")) = verdadero

not («lógico») devuelve el valor opuesto al argumento *lógico*.

Por ejemplo: not(exact("cadena", "cadena")) = 1
not(exact("cadena", "cadena")) = 0
not(False) = 1 o not(O) = 1
not(True) = 0 o not(1) =0
not(1+1=2) = 0

or («expr_1», «expr_2») devuelve TRUE cuando uno de los dos argumentos es verdadero y FALSE cuando los dos argumentos son falsos. Los argumentos deben poder interpretarse como valores lógicos.

Por ejemplo:
or(exact("c adena", "cadena"),exact("c adena", "cadena")) = 0
or(exact("cadena", "cadena"),exact("c adena", "cadena")) = 1
or(true,true) = 1 o or(1,1) = 1
or(true,false) = 1 o or(1,0) = 1
or(false,false)= 0 o or(0,0) = 0

Funciones matemáticas

Nota: la configuración regional del sistema Windows puede afectar a los resultados de algunas fórmulas.

Abs(data): Esta función proporciona el valor (positivo) absoluto de los datos. Permite utilizar letras después de números.

Ejemplos

Abs(-5) = 5

Abs(5) = 5

base10tobaseX(«cadena_1»,«cadena_2») convierte *cadena_2* de base 10 en base *cadena_1.*

Ejemplos

Si el campo llamado Base 16 contiene la cadena "0123456789ABCDEF"

BASE10TOBASEX(Base16, 12) da como resultado C

BASE10TOBASEX(Base16, 10) da como resultado A

BASE10TOBASEX("012345","9") da como resultado 13

baseXtobase10(«cadena_1»,«cadena_2») convierte *cadena_2* de base *cadena_1* en base 10.

Ejemplos

Si el campo llamado Base 16 contiene la cadena "0123456789ABCDEF"

BASEXTOBASE10(Base16, "E") da como resultado 14

BASEXTOBASE10(Base16,10) da como resultado A

BASEXTOBASE10("012345","9") da como resultado 13

Ceil(data): Esta función redondea los datos hasta el siguiente número entero. Permite utilizar letras después de números.

Ejemplos:

Ceil(3.234) = 4

Ceil(7.328) = 8

Decimals(data1, data2): Esta función utiliza data2 decimales en data1. Permite utilizar letras después de números.

Ejemplos:

Decimals(4, 2) = 4.00

Decimals(3.524, 1) = 3.5

eval_add(«cadena»,«cadena»): devuelve la suma de parámetros.

Ejemplo

eval_add(5,5)=10

eval_div(«cadena»,«cadena»): devuelve la división de parámetros.

Ejemplo

eval_div(20,2)=10

eval_mult(«cadena»,«cadena»): devuelve la multiplicación de parámetros.

Ejemplo

eval_mult(5,2)=10

eval_sub(«cadena»,«cadena»): devuelve la sustracción de parámetros.

Ejemplo

eval_sub(20,10)=10

Floor(data): Esta función redondea los datos hasta el siguiente número entero. Permite utilizar letras después de números.

Ejemplos:

Floor(3.234)= 3

Floor(7.328)= 7

hex(«val_1»,«val_2») convierte el número decimal *val_1* en número con formato hexadecimal y una longitud total de *val_.*

Ejemplo

*h*ex(2, 8) = 00000002

int («valor») devuelve el número entero más grande cuyo valor es inferior o igual al argumento *valor*.

Ejemplos:

int (-5.863) = -6

int (5.863) = 5

max(data1, data2): Esta función muestra el valor más alto. Permite utilizar letras después de números.

Ejemplo:

Max(5, 12) = 12

min(data1, data2): Esta función muestra el valor más bajo. Permite utilizar letras después de números.

Ejemplo:

Min(5, 12) = 5

mod («val_1», «val_2») devuelve el resto resultante de la división del argumento *val_1* por el argumento *val_2*. El resultado obtenido tiene el mismo signo (positivo o negativo) que el divisor.

Ejemplos:

mod (7,2) = 1

mod (-7,2) = 1

mod (7,-2) = -1

mod (-7,-2) = -1

quotient («val_1», «val_2») devuelve el entero resultante de la división del argumento *val_1* por el argumento *val_2*.

Ejemplo

quotient(10, 2) = 5

round («val_1», «val_2») devuelve el argumento *val_1* redondeado al número de cifras indicado por el argumento *val_2*.

- Si *val_2* es más grande que 0, *val_1* se redondea al número de decimales indicado.
- Si *val_2* da como resultado 0, *val_1* queda redondeado al número entero más próximo.
- Si *val_2* es inferior a 0, se redondean los enteros de *val_1*, sin tomar en cuenta los decimales.

Ejemplos:

round (4.25,1) = 4.3

round (1.449, 1) = 1.4

round (42.6,-1) = 40

Funciones relativas a texto

Nota: la configuración regional del sistema Windows puede afectar a los resultados de algunas fórmulas.

Una cadena de caracteres o cadena de texto es similar a una tabla en la que cada celda contiene un carácter de texto. Se caracteriza por su longitud (número total de caracteres de la cadena, incluidos los espacios). La **posición** de un carácter dentro de una cadena corresponde al lugar que ocupa en la tabla: el primer carácter ocupa la posición 1. Posición 3 corresponde al tercer carácter de la cadena.

cyclebasex *( )* permite efectuar una numeración con cualquier sistema de numeración de una base de datos. El sistema de numeración debe haber sido indicado dentro de la expresión vinculada, así como el valor inicial, el valor de cada incremento y el número de ejemplares de cada número. Todos estos valores pueden estar vinculados con otros campos de la etiqueta, pero los nombres de los campos no se deben escribir entre comillas.

Ejemplo:

Si un campo llamado Base16 contiene la cadena de caracteres 0123456789ABCDEF, entonces:

cyclebasex(base16, « 8 », 1 ,1) = 8,9,A,B,C

cyclebasex(base16, « F », -1,1) = F,E,D,C,B,A 9,8,7

cyclebasex(base16, « B0 », 1,1) = B0, B1, B2

cyclebasex(«012345 », « 4 »,1,2) = 4,4,5,5,10,10,11,11...

cyclechar *()* crea una serie de caracteres personalizada sobre un ciclo completo.

Ejemplo:

cyclechar(« A », « C ») = A B C A B C A B C

cyclechar(« A », « C », 1,2) = A A B B C C A A B B

cyclenumber *( )* permite ajustar una secuencia de números propia, opuesta a la forma habitual de incrementar los números o las letras (0,1,2… ó A,B,C).

Ejemplo:

cyclenumber(1,3) producirá la secuencia de etiquetas siguiente: 1 2 3 1 2 3 1 2 3...

cyclenumber(1,3,1,2) producirá la secuencia de etiqueta siguiente: 1 1 2 2 3 3 1 1 2 2 3 3 1 1...

cyclestring ( ) permite configurar un grupo de palabras o caracteres como ciclo completo (como un campo de incrementación). La cadena entera debe estar delimitada por comillas (" ") y cada palabra o grupo de caracteres separado de los demás por un punto y coma (;)).

Ejemplo:
cyclestring(« Mon ; Tue ; Wed ; Thu ; Fri ; Sat ; Sun ») = Mon Tue Wed Thu Fri Sat Sun
El ejemplo siguiente se aplica a la impresión de etiquetas identificadas con las letras del alfabeto, O e I excluidas.
cyclestring(« A;B;C;D;E;F;G;H;J;K;L;M;N;P;Q;R;ST;U;V;W;X;Y;Z »)

exact («cadena_1», «cadena_2») devuelve TRUE cuando ambas cadenas son idénticas y FALSE de lo contrario.

Ejemplo:
exact("aplicación","aplicación") = 1
exact("apicación","aplicación") = 0

Extract(«string», «sep», «pos») devuelve la subcadena desde la cadena de caracteres «string» a la posición especificada «pos» que contiene los datos separados por la cadena «sep».

Ejemplo:
Extract("1;2;3;4", ";", 3) = 3

find («cadena», «clave», «inicio») devuelve la posición de la primera ocurrencia del argumento *clave* dentro del argumento *cadena*. Dentro del argumento *cadena*, la búsqueda empieza a partir de la posición indicada por el argumento *inicio* (*inicio* >= 1). La función devuelve cero como resultado si no se encuentra ninguna ocurrencia del argumento *clave*. Se toman en cuenta las mayúsculas y las minúsculas.

Ejemplo:
find("Peter McPeepert","P",1) = 1
find("Peter McPeepert","p",1) = 12

left («cadena», «número») devuelve la cadena de caracteres extraída del argumento *cadena*. Esta cadena empieza en la posición 1 del argumento *cadena* y su longitud corresponde a la indicada en el argumento *número*.

Ejemplo:
left("Peter McPeepert",1) = P
left("Peter McPeepert ",5) = Peter

len («cadena») devuelve la longitud del argumento *cadena*. Los espacios se cuentan como caracteres.

Ejemplo:
len("París, New York") = 15
len("") = 0
len(" ") = 1

lower («cadena») convierte todas las letras mayúsculas de una cadena de texto en letras minúsculas.

Ejemplo:
lower("París, New York") = parís, new york

LTrim(«string»): elimina cualquiera de los espacios iniciales o finales de los datos de la izquierda.

Ejemplo:
LTrim(" No."): No

mid («cadena», «inicio», «número») devuelve la cadena de caracteres extraída del argumento *cadena*. Esta cadena empieza en la posición indicada por el valor del argumento *inicio* (*inicio* >=1) y su longitud corresponde a la indicada en el argumento *número*.

Ejemplo:
mid("Paris, New York",8,8) = New York

Output («variable»): devuelve el valor de la variable en el formato adecuado.

Ejemplo:
partiendo de un Contador0 con el prefijo "número", la fórmula **contador0&output(contador0)** da como resultado:
1 **número:** 001

Observación: el formato de la variable se configuran mediante la ficha **Salida**.

pad («cadena», «longitud», «carácter») añade caracteres en la parte izquierda del campo para que la entrada quede configurada con una longitud determinada. Se puede escoger cualquier carácter para que sirva de carácter de relleno.

Ejemplo:
Si un campo llamado GREETING tiene como valor HELLO, entonces:
pad(GREETING,8) = 000HELLO

pad(5,3) = 005

pad(Nine,6,a) = aaNine

replace («cadena», «inicio», «número», «nueva_cadena») devuelve el argumento *cadena* transformado. El número de caracteres indicado por el argumento *número* a partir de la posición indicada por el argumento *inicio* se sustituye por el argumento *nueva_cadena*.

Ejemplo:

replace("Paris, New York",8,8,"Singapour") = Paris, Singapour

ReplaceString(«cadena», «vieja_cadena», «nueva_cadena») reemplaza todas las ocurrencias de un «old_string» especificado en la cadena de caracteres «string» por otro «new_string» especificado.

Ejemplo:

ReplaceString("abc12def12", "12", "") = abcdef

rept («cadena», «número») devuelve una cadena de caracteres en la que el argumento *cadena* se repite *número* de veces.

Ejemplo:

rept("Ah París ! ",2) = Ah París ! Ah París !

right («cadena», «número») devuelve la cadena de caracteres formada por los últimos caracteres del argumento *cadena*, con la longitud que indica el argumento *número*.

Ejemplo:

right("Vale de pedido",6) = pedido

RTrim («string»): elimina cualquiera de los espacios iniciales o finales de los datos de la derecha.

Ejemplo:

RTrim("Part ") :Part

search («cadena», «clave», «inicio») devuelve la posición de la primera ocurrencia del argumento *clave* dentro del argumento *cadena*. La búsqueda empieza a partir de la posición que indica el argumento de *inicio* (*inicio* >= 1). La función devuelve cero como resultado si no se encuentra ninguna ocurrencia del argumento *clave*.

Ejemplo:

search("Vale de pedido","pedido,1) = 9

search("Vale de pedido","p",1) = 9

Sort(«lista», «separador», «dirección»): Devuelve una lista ordenada de elementos. El orden se realiza por comparación de cadenas. La lista es un elemento en cadena que contiene varios elementos con un separador como una coma o un punto y coma. La dirección es un número opcional que define la dirección de búsqueda, 0 = ascendente, 1 = descendiente. Si no se especifica dirección, será ascendente por defecto.

Ejemplos:

Sort(“3;5;6;1”, “;”) = “1;3;5;6”

Sort("one;two;three", ";", 1) =“two;three;one”

StrAfter(«data»,«start after», «length»): devuelve una cadena con la misma longitud de caracteres después de un origen especificado posterior al carácter.

Ejemplos:

StrAfter("1234-5678", '-', 3)= Selecciona los 3 caracteres después del guión (567).

StrAfter("1234-5678", '-')= Selecciona todos los caracteres después del guión (5678).

StrBefore(«data»,«start before», «length»): devuelve una cadena con la misma longitud de caracteres antes de un origen especificado previo al carácter.

Ejemplos:

StrBefore("1234-5678", '-', 2)= Selecciona los 2 caracteres inmediatamente antes del guión (34).

StrBefore("1234-5678", '-')= Selecciona todos los caracteres antes del guión (1234).

SuppressBlankRows («string»): Devuelve una cadena donde se han omitido las líneas vacías. Permite crear un objeto, colocar dentro los campos que desee y eliminar los que estén en blanco.

Ejemplo:

SuppressBlankRows({Var0} & char(10) & {Var1} & char(10) & {Var2}) (Var0, Var1 and Var2 son variables, char(10) es símbolos "\n" que significan línea nueva)

Variables/Valores	Fórmula	muestra

Var0 = "Doris" Var1 = "Bull Run Ranch" Var2 = "Aurora"	SuppressBlankRows({Var0} & char(10) & {Var1} & char(10) & {Var2})	Doris Bull Run Ranch Aurora
Var0 = "Craig" Var1 = "" Var2 = "Chicago"	SuppressBlankRows({Var0} & char(10) & {Var1} & char(10) & {Var2})	Craig Chicago
Var0 = "Craig" Var1 = "" Var2 = "Chicago"	{Var0} & char(10) & {Var1} & char(10) & {Var2}	Craig Chicago

SuppressEmpty(«lista», «separador»): elimina las entradas vacías en la lista de objetos. La lista es un elemento en cadena que contiene varios elementos con un separador como una coma o un punto y coma.

Ejemplos:
SuppressEmpty("3;;;1", ";") = "3;1"

trim(«cadena») devuelve el argumento *cadena* transformado. Se suprimen todos los espacios en blanco situados al principio o al final de la cadena. El número de espacios en blanco comprendidos entre dos palabras se reduce a 1.

Ejemplo:
trim(" Vale de pedido ") = Vale de pedido

trimall(«cadena») devuelve el argumento *cadena* transformado. Se suprimen todos los espacios.

Ejemplo:
trimall("Paris / New York / Rome") = Paris/NewYork/Rome

Unique(«lista», «separador», «comparador»): elimina las entradas duplicadas en la lista de objetos. La lista es un elemento en cadena que contiene varios elementos con un separador como una coma o un punto y coma. El comparador indica si la comparación distingue entre mayúsculas y minúsculas o no, 0 = distingue entre mayúsculas y minúsculas, 1 = ignora mayúsculas y minúsculas. Si no se especifica, el comparador distingue entre mayúsculas y minúsculas por defecto.

Ejemplos:
Unique(“3;5;3;1”, “;”) = “3;5;1”
Unique("one;ONE;two", ";", 1) = “one;two”

upper («cadena») devuelve el argumento *cadena* en mayúsculas.

Ejemplo:
upper("Vale de pedido") = VALE DE PEDIDO

ztrim («cadena») suprime todos los ceros a la izquierda del primer número (diferente de cero). Esta función se aplica a los valores numéricos únicamente.

Ejemplo:
Si un campo llamado WEIGHT tiene como valor 000200, entonces...
ztrim(weight) = 200

Cómo configurar las propiedades de una fuente de datos de tipo Fórmula

Comando: **Fuente de datos > Fórmula > Propiedades...**

1. Introduzca la fórmula en la zona de edición.
   - o -
   Seleccione con el ratón los elementos que desee y escoja **Insertar**.

2. Pulse **Aceptar**.

Sugerencia: también puede insertar el elemento haciendo doble clic en su nombre.

Cuidado: si, para construir una fórmula, utiliza una variable cuyo nombre incluye uno de los caracteres siguientes **&+-*/<>=^%,!\"**, debe introducir el signo {} al comienzo y al final del nombre.

Observaciones:
Vista previa dinámica que respresenta el resultado del cálculo de la fórmula actual, incluido el formato definido en la página Salida. En caso de error, la vista previa aparece en rojo. Si el valor resultante aparece cortado, debe modificar el ajuste de longitud máxima en la ficha **Salida**.

Ejercicio : Cálculo del ''Modulo'' adecuado

Esta aplicación incluye funciones de cálculo del modulo para integrar un carácter de control en los mensajes codificados por el código de barras.

No obstante, cuando se debe calcular un carácter de control poco usual, las funciones de cálculo que integran aplicación pueden no ajustarse a sus necesidades.

En ese caso, debe crear la función usted mismo. En este apartado veremos cómo efectuar cualquier tipo de cálculo de carácter de control mediante fórmulas.

El ejemplo aquí presentado ilustra el cálculo de un carácter de control destinado a un código de barras 2/5 interleaved.

Método de cálculo de un carácter de control

El método consiste en multiplicar el primer carácter del mensaje por 1, el segundo por 2, el tercero por 1, etc.

- Abra la etiqueta CHECK_01.LAB, situada en la carpeta TUTORIAL.

El cálculo del carácter de control se efectuará a partir del dato Lot Number.

Para calcular el peso

1. Seleccione la rama **Fórmula** de la ficha **Fuentes de datos** y añada una **Fórmula**.
2. Introduzca la fórmula siguiente:

   mid(LOT NUMBER,1,1)&
   mid(LOT NUMBER,2,1)*2 &
   mid(LOT NUMBER,3,1)&
   mid(LOT NUMBER,4,1)*2 &
   mid(LOT NUMBER,5,1)

   El resultado obtenido es 2120103, ya que:

   2*1 = 2
   6*2 = 12
   0*1 = 0

3*1 = 3

La concatenación de los resultados forma 2120103.

3. Llame a la fórmula WEIGHTED.

Para adicionar el resultado del cálculo de pesos:

El paso siguiente consiste en adicionar los números resultantes de la fórmula precedente.

1. Cree una nueva Fórmula y llámela SUM.
2. En la zona de edición, introduzca la expresión siguiente:

   mid(weighted,1,1)+mid(weighted,2,1)+
   mid(weighted,3,1)+mid(weighted,4,1)+
   mid(weighted,5,1)+mid(weighted,6,1)+ mid(weighted,7,1)

   El resultado obtenido es 9, ya que: 2+1+2+0+1+0+3= 9.

Para calcular el carácter de control:

Vamos a utilizar el resultado precedente para calcular el valor del carácter de control.

1. Cree una nueva Fórmula y llámela CHECK DIGIT.
2. En la zona de edición, introduzca la expresión siguiente:

   if((sum % 10)>0,10-sum % 10,0)

   El resultado obtenido es 1, ya que: sum % 10 = 9 (% = modulo).

   como 9 es mayor que 0, efectuamos la sustracción: 10- 9 = 1.

Observación: Si el resultado hubiera sido igual a 0, el valor del carácter de control hubiera sido 0.

Para calcular el mensaje a codificar:

La creación del código de barras exige que se indique el mensaje que se desea codificar, en este caso el valor de la variable ”r;LOT NUMBER” seguida del valor del carácter de control (CHECK DIGIT).

1. Cree una nueva Fórmula y llámela DATA.
2. En la zona de edición, introduzca la expresión siguiente: LOT NUMBER & CHECK DIGIT.
   El mensaje a codificar resultante es: 260531

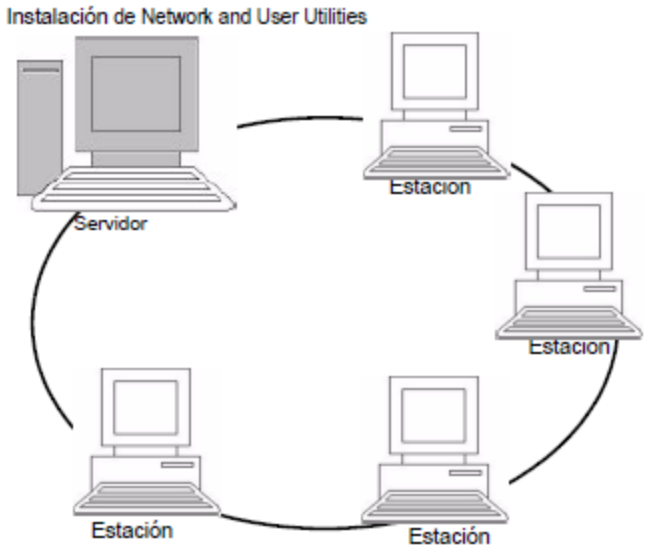
Para crear el código de barras:

1. Escoja la herramienta de creación de Códigos de barras, situada en la barra de herramientas de creación, seleccione luego la fórmula DATA y arrástrela para colocarla en la etiqueta.
2. Seleccione el código de barras y escoja **Propiedades** del objeto en su menú contextual.
3. Configure el cuadro del diálogo.
4. Ya está acabada la etiqueta. Encontrará la etiqueta de ejemplo en TUTORIAL\CHECK.TPL.

Instalación de la versión de red

Descripción funcional

Para utilizar la aplicación de etiquetado en versión de red o multiusuario debe empezar por instalar **Network License Manager** en el servidor o estación que se utilizará como servidor antes de instalar la aplicación de etiquetado en cada una de las estaciones.



Procedimiento de instalación

Preliminares

Antes de empezar la instalación, el administrador de red debe haber decidido cómo definir la estructura de la red para el grupo de usuarios, es decir:

- Definir el servidor de licencias en el que se instalarán **Network License Manager** y la mochila.
- Definir las estaciones de trabajo o las estaciones de trabajo clientes que se utilizarán la aplicación de etiquetado.

Presentación de Network Manager

Network License Manager permite utilizar la configuración de red de su aplicación de etiquetado. **Network License Manager** incluye:

- **Network License Manager** (License Service)
- **Network Settings Wizard**: un asistente que ayuda a configurar la red.
- **Administrador de usuarios**: queda instalado al mismo tiempo que **Network License Manager** y permite configurar los derechos de utilización en red de la aplicación de etiquetado.

Instalación de Network License Manager en el servidor

Antes de instalar el software de etiquetado en todas las estaciones de trabajo que van a utilizarlo, debe instalar la utilidad **Network License Manager** en el servidor para configurar la red.

Para instalar Network License Manager en el servidor

1. Introduzca el DVD de instalación en la unidad correspondiente. Aparece la ventana de instalación.
   Si el DVD no se ejecuta automáticamente: Vaya a Windows Explorer y expandir la letra de la unidad de DVD. Haga doble clic en index.hta (por ejemplo, D:\index.hta).
2. Seleccione **Network License Manager**, que incluye **License Service** y **Administrador de usuarios**. Después haga clic en el botón **Instalar**.
3. Siga las instrucciones de la pantalla.
4. Si desea definir los ajustes de su configuración de red, inicie **Network Settings Wizard** en el servidor. Si no modifica la configuración, cada estación de trabajo tendrá una configuración específica por defecto.

Configuración

Todas las herramientas necesarias para configurar la versión de red están disponibles en **Network toolbar**, a la que se puede acceder desde la barra de tareas de Windows (Systray).

Desde la barra de tareas de Windows, haga doble en el icono para que la **Network toolbar** sea visible.

Network Settings Wizard le ayuda a definir los ajustes de su versión de red.

1. Para iniciar **Network Settings Wizard**, haga clic en el icono .
2. Selección un modo de configuración: Genérico, por usuario o por estación.

    - **generico**: la configuración será la misma para todos los usuarios y estaciones de trabajo. (user.ini)
    - **por usuario**: cada usuario podrá acceder a su configuración a partir de cualquier estación de trabajo. (user name.ini)
    - **por estación**: cada estación de trabajo tiene su propia configuración (station.ini).

3. Especifique dónde desea guardar la configuración. Si desea compartir la configuración entre varias estaciones de trabajo, especifique una ruta de red accesible a todas ellas. (Ejemplo: TKDongle).
4. Especifique dónde desea guardar los datos compartidos (fuentes de datos, listas, archivo histórico de impresión, ...). Asegúrese de que todos los usuarios tiene acceso correcto a estas carpetas.

Para configurar Administrador de usuarios

Si desea definir privilegios de acceso de red para todos los usuarios de la aplicación de etiquetado, debe realizarlo en el momento de la instalación (consulte la ayuda de **Administrador de usuarios**). Haga clic en el icono **Administrador de usuarios** en la barra de herramientas Red.

Inicio del administrador de licencias (License Service)

Antes de instalar la aplicación de etiquetado en todas las estaciones de trabajo, debe asegurarse de haber iniciado el administrador de licencias.

License Service está instalado como un servicio. Este servicio, denominado **TkxWebLicenseServer**, se habilita automáticamente cuando se inicia el servidor.

Para iniciar License Service

Haga click en el icono disponible en la barra de herramientas **Network License Manager**

- o -

Haga doble clic en el archivo TkxWebLicenseServerController.exe.

- o -

Haga clic con el botón derecho del ratón en el icono de la barra **Network License Manager** desde la barra de tareas de Windows y seleccione **Controlador del License Service**

Instalación de la aplicación en las estacione

La aplicación de etiquetado debe estar instalada en todas las estaciones de trabajo en las que se utilice.

Para instalar la aplicación en una estación de trabajo

1. Introduzca el DVD de instalación en la unidad correspondiente. Aparece la ventana de instalación.
   Si el DVD no se ejecuta automáticamente: Vaya a Windows Explorer y expande la letra de la unidad de DVD. Haga doble clic en index.hta (por ejemplo, D:\index.hta).
2. Seleccione el producto que desea instalar, haga clic en el botón **Instalar** y siga las instrucciones que irán apareciendo en la pantalla.
3. Inicie la aplicación de etiquetado. Se muestra el administrador de licencias. Haga clic en **Probar** para iniciar el software.. En el menú **Herramientas**, escoja **Administración de la red.**
   - o -
   Desde el menú Inicio de Windows, seleccione el acceso directo **Administración de redes** en el grupo de software de etiquetado.
4. Active la opción **Usar la licencia de red**.

5. Seleccione el **Tipo de licencia Network**.

   - **Licencia de carpeta compartida** usa la función compartir archivos de Windows para la comunicación entre el software y el License Manager.

   - **Licencia web** usa la comunicación http/https entre el software y el License Manager.

6. Especifica el **puerto de servidor** si ha seleccionado el tipo **Licencia web**.
7. Haga clic en **Modificar...** para seleccionar en qué servidor están instalados el administrador de licencias y la mochila.
   - o -
   Haga clic en **Examinar...** para buscar automáticamente el servidor en el que está instalado el administrador de licencias.

   Si ya existe una configuración de red aparecerá un mensaje preguntando si se desea utilizar la configuración de red actual que se muestra.

8. Si desea modificar o configurar los ajustes de red, haga clic en el botón **Network Settings Wizard**.
9. Haga clic en **OK**.
10. Vuelva a iniciar el programa.

France
+33 (0) 562 601 080

Germany
+49 (0) 2103 2526 0

Singapore
+65 6908 0960

United States
+1 (414) 837 4800

Copyright 2026 TEKLYNX Corporation SAS. All rights reserved. LABEL MATRIX, LABELVIEW, CODESOFT, LABEL ARCHIVE, SENTINEL, PRINT MODULE, BACKTRACK, TEKLYNX CENTRAL, TEKLYNX, and Barcode Better are trademarks or registered trademarks of TEKLYNX Corporation SAS or its affiliated companies. All other brands and product names are trademarks and/or copyrights of their respective owners.

www.teklynx.com