

Guía del usuario de Power Form

Versión 2.0

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

Introducción.....	4
El diseño	4
La barra de menús.....	4
La barra de herramientas.....	5
El árbol Solución.....	7
El cuadro de propiedades	8
La ventana principal.....	9
La barra de estado.....	9
Conceptos básicos	10
¿Qué son las acciones?	10
Acción de script	15
Solución de problemas	23
Uso de las variables de formulario	24
Acción Actualizar objeto.....	24

Introducción

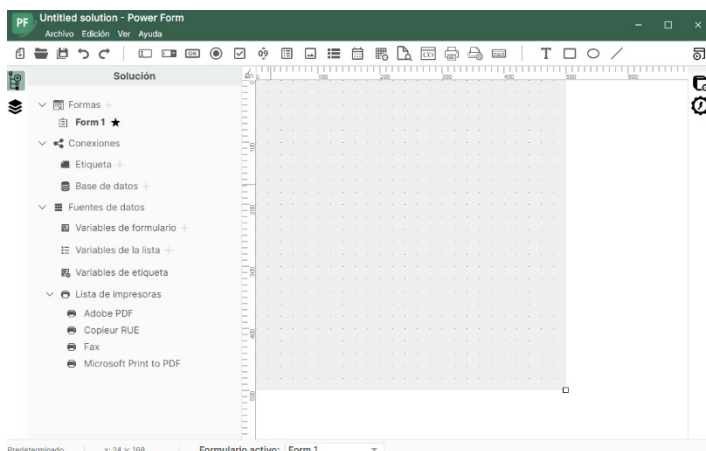
Power Form es una herramienta de diseño visual que permite crear rápidamente aplicaciones de entrada de datos y de impresión de etiquetas.

Incluye funciones como:

- Selección del tipo de etiqueta e impresora
- Vista previa de impresión
- Ajustes de página y de impresora
- Conexiones a bases de datos
- Aplicación con varias pantallas o formularios
- Valores predefinidos y control de datos

El diseño

La interfaz de Power Form es fácil de usar, lo que reduce el tiempo necesario para familiarizarse con el software. Incluye una barra de menús, una barra de herramientas, un árbol de soluciones, un cuadro de propiedades, una ventana principal y una barra de estado.



La barra de menús

La barra de menús contiene cuatro opciones: Archivo, Editar, Ver y Ayuda. Al hacer clic en cada una de ellas, se desplegará una nueva lista de opciones, lo que permitirá al usuario realizar diversas funciones. Para acceder más rápidamente a las opciones del menú, pulse Alt + la letra de la opción que desee:

- Alt+F abrirá el menú Archivo.
- Alt+E abrirá el menú Edición.
- Alt+V abrirá el menú Ver.
- Alt+H abrirá el menú Ayuda.



La barra de herramientas

La barra de herramientas suele aparecer justo debajo de la barra de menús. Está compuesta por iconos que representan los objetivos y funciones del software más utilizados para gestionar el formulario. Puede hacer clic en estos iconos para acceder rápidamente a estas funciones.

Los objetos se componen de 16 controles y 4 formas que se pueden añadir al formulario: Entrada de texto, Cuadro combinado, Botón, Botón de opción, Casilla de verificación, Numérico Arriba/Abajo, Cuadro de grupo, Cuadro de imagen, Cuadro de lista, Selector de fecha, Tabla de datos, Vista previa de etiqueta, Tabla de datos de variables de etiquetas, Selector de impresora, Ajustes de impresión, Texto, Rectángulo, Elipse y Línea.

Cada control tiene una función específica. Pueden vincularse a datos fijos o variables y asociarse a acciones. Para insertar un control en el formulario, solo tiene que hacer clic en él y arrastrarlo al formulario.

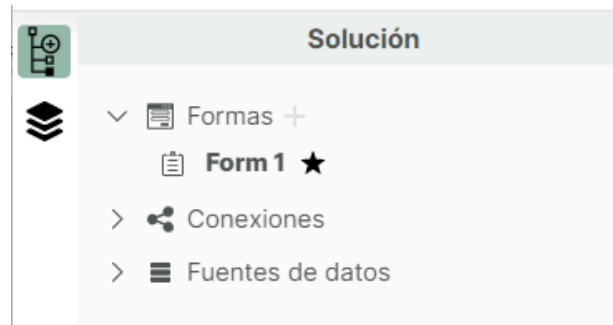
Control	Descripción
Entrada de texto	Campo de entrada de datos
Cuadro combinado	Muestra una lista desplegable de datos según lo definido en sus propiedades
Botón	Un control activo que se utiliza para realizar determinadas

	funciones o acciones
Botón de opción	Muestra una lista de datos en forma de grupo de opciones.
Casilla de verificación	Un control que se utiliza para activar o desactivar, o para seleccionar o deseleccionar, funciones predefinidas
Numérico arriba/abajo	Campo de entrada de datos solo para valores numéricos
Cuadro de grupo	Cuadro inactivo situado alrededor de un grupo de controles para facilitar la organización visual
Cuadro de imagen	Un cuadro en el que se puede importar una imagen
Cuadro de lista	Muestra una lista de datos según lo definido en sus propiedades
Selector de fecha	Campo de entrada de datos para el formato de fecha definido en sus propiedades
Tabla de datos	Muestra la tabla de la base de datos definida en las propiedades, junto con algunas funciones básicas, como filtros y campos de búsqueda
Vista previa de etiqueta	Un cuadro que mostrará una vista previa de la etiqueta asociada al formulario
Tabla de datos de variables de etiquetas	Muestra las variables de etiquetas "Cuando se imprime" como Nombre, Valor
Selector de impresora	Permite al usuario seleccionar la impresora activa y modificar los ajustes de la impresora seleccionada

Control	Descripción
Ajustes de impresión	Permite al usuario imprimir la etiqueta seleccionada y configurar la cantidad de etiquetas, los ejemplares de la etiqueta, los ejemplares de la página y los parámetros de intercalado
Teclado virtual	Un teclado en pantalla que se utiliza para introducir datos.
Texto	Texto inactivo que se utiliza para mostrar el nombre de un campo de entrada
Rectángulo	Muestra una figura con forma de rectángulo
Elipse	Muestra una figura con forma de elipse
Línea	Muestra una figura con forma de línea

El árbol Solución

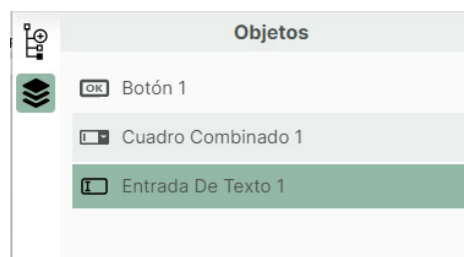
El árbol **Solución** es un conjunto de pestañas que se muestra en la parte izquierda de la aplicación. Se compone de secciones que se utilizan para aplicar configuraciones generales a la solución.



La pestaña **Solución** se compone de tres secciones diferentes que se pueden configurar y aplicar a todo el formulario: Formularios, Conexiones y Fuentes de datos.

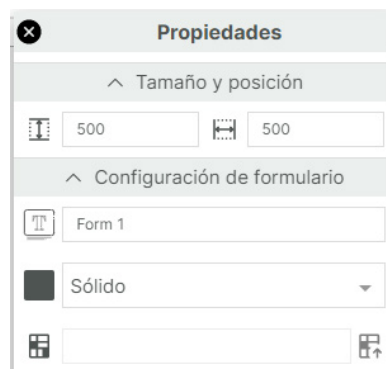
Sección	Descripción
Formas	Una lista de formularios que se utilizan en la solución para representar diferentes áreas de trabajo
Conexiones	Una lista de conexiones disponibles, que incluye la conexión a la API, la conexión a la etiqueta y la conexión a la base de datos.
Fuentes de datos	Contiene diferentes tipos de variables que se pueden utilizar en la solución. Existen cuatro tipos de fuentes de datos: variables de formulario, variables de lista, variables de etiqueta y lista de impresoras.

La pestaña **Objetos** muestra una lista de los objetos que se encuentran en el formulario activo.



El cuadro de propiedades

El **cuadro de propiedades** muestra todas las propiedades del control seleccionado. Aquí puede modificar las características de sus controles, como el nombre, el color, el tamaño o las acciones. Depende del contexto, por lo que la lista de propiedades variará en función del control seleccionado. Cuando se activa el cuadro, el menú contextual deja de estar disponible.

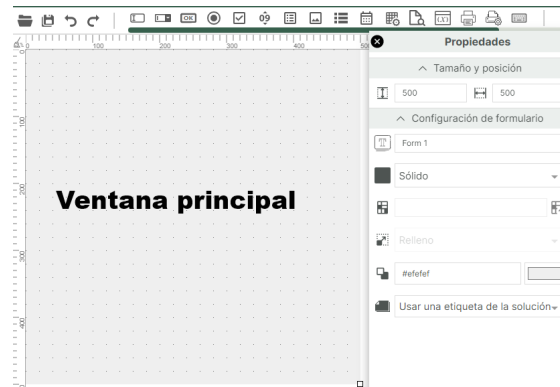


Algunos controles pueden incluir una sección Tipo de datos que permite a los usuarios asignar una variable o vincular una base de datos a ese control. Una vez asignado, el valor del control se vinculará al tipo de datos seleccionado y cualquier cambio que se realice, ya sea en el control o en el tipo de datos, se sincronizará.



La ventana principal

La **ventana principal**, que aparece en el centro, es su área de trabajo. Aquí podrá añadir controles y crear los formularios que utilizará para introducir los datos necesarios para la impresión de etiquetas. Una vez creado y configurado el formulario, puede ir a **Archivo > Ejecutar** para ejecutarlo y ver el resultado final.



La barra de estado

La **barra de estado** muestra información sobre la impresora activa, la posición del puntero en los ejes X e Y y el menú desplegable del selector de formularios activo.



Conceptos básicos

¿Qué son las acciones?

Las **acciones** son funciones que pueden asignarse a botones y abrir/cerrar formularios. En Power Form, se han incluido 14 acciones predefinidas en el menú "**Acciones**" para ahorrarle tiempo y evitar que tenga que programar. En algunos casos en los que es necesario configurar parámetros específicos, estos también se han predefinido en la configuración de la acción.

Las acciones predefinidas son:

Acción	Ventana emergente de confirmación
	Muestra el cuadro de diálogo de confirmación, donde el usuario puede ver un mensaje predefinido y unos botones para aceptar o cancelar el mensaje.
Parámetros	Título
	Permite al usuario establecer el título del encabezado.
	Mensaje
	Permite al usuario establecer el texto principal del mensaje.
	Botones
	Menú desplegable con varias opciones de botones que se mostrarán en la ventana emergente del mensaje.
	Icono
	Menú desplegable con varias opciones de iconos que se mostrarán en la ventana emergente del mensaje.

Acción	Actualizar valor de la variable
	Permite al usuario actualizar variables.
Parámetros	Nombre de la variable
	El usuario puede seleccionar el nombre de la variable que desea actualizar.

	Nuevo tipo de valor
	Menú desplegable con los tipos de valor que se pueden aplicar a la variable que se va a actualizar.
	Opción
	Permite al usuario añadir un nuevo valor al valor actual de la variable seleccionada.
	Nuevo valor
	Campo que permite al usuario seleccionar un nuevo valor para la variable seleccionada.

Acción	Mostrar formulario
	Permite al usuario abrir un nuevo formulario.
Parámetros	Formulario a mostrar
	Lista de formularios en la que el usuario puede seleccionar cuál desea visualizar.
	Comportamiento de visualización del formulario
	Abre el formulario seleccionado en una nueva ventana o en la misma.
	Opción
	Permite al usuario abrir el formulario que tenía abierto anteriormente, si existe.

Acción	Salir del formulario
	Permite al usuario cerrar la ventana del ejecutor de formularios.

Acción	Configurar impresora
	Permite al usuario imprimir desde la impresora seleccionada y anular cualquier otra impresora que pueda estar asociada al documento.
Parámetro	Configurar impresora
	Permite establecer una impresora como impresora predeterminada para su uso al hacer clic en el botón, o bien utilizar la impresora seleccionada de una lista, siempre que se utilice una variable de selección de impresora y esté definida en las propiedades del formulario.

Acción	Imprimir etiqueta
	Permite al usuario imprimir.
Parámetros	Número de etiquetas
	Envía a la impresora el número de etiquetas que se deben imprimir. Si el documento incluye un contador, este se incrementa a medida que aumenta el número de etiquetas.
	Ejemplares de la etiqueta
	Indica a la impresora cuántas copias idénticas de cada etiqueta se deben imprimir.
	Ejemplares de la página
	Indica a la impresora cuántas copias idénticas de cada página se deben imprimir.
	Etiquetas entre cortes
	Indica a la impresora cuántas etiquetas se deben imprimir antes de cada corte.

Acción	Leer datos del archivo
	Permite al usuario leer los datos de los archivos y asignar su valor a una variable o a un control.
Parámetros	Elegir archivo
	Permite al usuario seleccionar el archivo que se va a leer.
	Tamaño máximo
	Establece el tamaño máximo de las líneas que se leerán del archivo.
	Añadir tipo de valor
	Permite al usuario seleccionar dónde aplicar los datos del archivo; puede ser una variable o un control.
	Elegir variable/control
	Permite al usuario seleccionar qué control de datos se aplicará.

Acción	Abrir diálogo
	Permite al usuario abrir el diálogo Diseñador de etiquetas
Parámetro	Abrir diálogo
	Menú desplegable con la lista de los diálogos disponibles para abrir.

Acción	Abrir enlace web
	Permite al usuario abrir el enlace web seleccionado
Parámetro	Dirección de enlace
	Permite introducir el enlace que se va a abrir.

Acción	Ejecutar solicitud SQL
	Permite al usuario enviar una solicitud SQL a la base de datos conectada para su ejecución.
Parámetro	Base de datos
	Seleccione la base de datos en la que se ejecutará la solicitud SQL
Parámetro	Instrucción SQL
	Área de texto para el código SQL

Acción	Ejecutar solicitud de script
	Permite al usuario ejecutar código JavaScript nativo para comparar valores, actualizar variables, escribir datos en un archivo, etc.
Parámetro	Script
	Área de texto para el código
Parámetro	Categoría
	Menú desplegable con la lista de nombres de categorías para métodos, constantes, operadores, etc.
Parámetro	Fragmentos
	Lista de fragmentos en la categoría seleccionada
Parámetro	Agregar
	Botón para añadir el fragmento seleccionado al área de texto del código
Parámetro	Validar
	Botón para validar la sintaxis del código introducido en el área de texto.

Acción	Actualizar base de datos
	Permite al usuario forzar la actualización de la base de datos
Parámetro	Obtener datos de

	Un menú desplegable con la opción "Base de datos" para forzar la actualización de la base de datos y "A partir del resultado de la acción anterior" para actualizar la base de datos mediante el filtrado de acciones SQL.
--	--

Acción	Abrir enlace web
	Permite al usuario abrir el enlace web especificado.
Parámetro	Dirección de enlace
	Campo para introducir la dirección del enlace web que se abrirá.

Acción	Establecer foco
	Permite al usuario establecer el foco en el control seleccionado.
Parámetro	Elegir control
	Permite seleccionar el control que recibirá el foco.

Acción	Actualizar objeto
	Permite al usuario actualizar el control seleccionado.
Parámetro	Elegir control
	Permite seleccionar el control que se actualizará.
Parámetro	Lista de propiedades
	Muestra la lista de propiedades que se pueden actualizar en el control seleccionado

Acción de script

"Ejecutar acción de script" permite a los usuarios vincular código JavaScript a elementos de la interfaz de usuario (como botones) en una aplicación de Power Forms. Esto permite ampliar la lógica de la aplicación escribiendo scripts que respondan a las interacciones del usuario.

Instrucciones paso a paso:

1. Seleccione un elemento de la interfaz de usuario

1. Haga clic en el elemento de la interfaz de usuario al que quiera asociar un script (p. ej., un botón).
2. Consejo: Puede seleccionar botones, campos de entrada, cuadrículas o cualquier elemento que admita eventos.

2. Abra el panel Eventos

1. Haga clic en el icono del engranaje .
2. En el panel "Eventos", localice el evento que quiere gestionar (p. ej., "Clic").
3. Pulse el botón "..." situado junto al nombre del evento para configurarlo.

3. Añada una acción de script

1. En el cuadro de diálogo "Configurar acciones de eventos" que se abre:
 - a. Haga clic en el icono "Ejecutar acción de script" .
 - b. Aparecerá una nueva acción en la lista.

4. Escriba su script

En el panel "Propiedades", introduzca su lógica de JavaScript personalizada.

Ejemplo:

```
const a = 12;
```

```
const b = 32;
```

```
return b * a;
```

* Este fragmento de código JavaScript define dos variables constantes: `a`, a la que se le asigna el valor 12, y `b`, a la que se le asigna el valor 32. A continuación, devuelve el resultado de multiplicar `b` por `a`.

5. Valide y aplique

1. Haga clic en "Validar" para comprobar si su script contiene errores sintácticos.
2. Una vez completado el proceso, pulse "Aplicar" para guardar la acción.

Uso de fragmentos de código

La interfaz de scripting ofrece una lista de fragmentos de código predefinidos organizados por categorías. Estos fragmentos de código permiten insertar expresiones, constantes, estructuras y funciones de uso común sin necesidad de escribirlas desde cero.

Para utilizar un fragmento:

1. Seleccione una categoría en la lista desplegable.
2. Seleccione un fragmento en la segunda lista desplegable.
3. Haga clic en "Agregar" para insertarlo en el editor de scripts.

Todos los fragmentos disponibles:

Snippet	Description
Math	
Math.sqrt(number)	Returns the square root of a number
Math.abs(number)	Returns the absolute value
Math.cos(number)	Returns the cosine of a number (radians)
Math.sin(number)	Returns the sine of a number (radians)
Math.ceil(number)	Rounds up to the nearest integer
Math.floor(number)	Rounds down to the nearest integer
Math.round(number)	Rounds to the nearest integer
Math.random()	Returns a random number between 0 and 1
Math.pow(a, b)	Raises a to the power of b
Math.max(a, b)	Returns the maximum of two values

Snippet	Description
Math.min(a, b)	Returns the minimum of two values
Math.log(x)	Natural logarithm (base e) of x
Math.exp(x)	Returns e to the power of x
JSON	
JSON.stringify(object)	Converts an object to a JSON string
JSON.parse(jsonString)	Parses a JSON string into an object
Variables	
let x = value;	Declares a variable with block scope
const PI = 3.14;	Declares a constant
let result;	Declares an uninitialized variable
var total = 0;	Declares a function-scoped variable
Constants	
Number.MAX_VALUE	Maximum numeric value
Number.MIN_VALUE	Smallest positive numeric value
undefined	Primitive indicating no value
null	Intentional absence of any object value
NaN	Not-a-Number value
Math.PI	The ratio of circumference to diameter
Math.E	Euler's number
Math.LN2	Natural log of 2

Snippet	Description
Math.LN10	Natural log of 10
Math.LOG2E	Base 2 log of e
Math.LOG10E	Base 10 log of e
Math.SQRT1_2	Square root of 0.5
Math.SQRT2	Square root of 2
Logical	
$a > b$	Greater than comparison
$a < b$	Less than comparison
$a \geq b$	Greater than or equal to
$a \leq b$	Less than or equal to
$a \&\& b$	Logical AND
$a \parallel b$	Logical OR
$!a$	Logical NOT
$a != b$	Inequality comparison
$a !== b$	Strict inequality comparison
TRUE	Boolean true
FALSE	Boolean false
Operators	
$a + b$	Addition
$a - b$	Subtraction

Snippet	Description
<code>a * b</code>	Multiplication
<code>a / b</code>	Division
<code>a % b</code>	Modulo
<code>a ** b</code>	Exponentiation
<code>a += b</code>	Add and assign
<code>a -= b</code>	Subtract and assign
<code>a *= b</code>	Multiply and assign
<code>a /= b</code>	Divide and assign
<code>a == b</code>	Equality
<code>a === b</code>	Strict equality
<code>++a</code>	Pre-increment
<code>--a</code>	Pre-decrement
Statements	
<code>if (condition) { }</code>	Conditional execution
<code>else { }</code>	Executes if condition is false
<code>else if (condition) { }</code>	Checks another condition
<code>for (let i = 0; i < n; i++) { }</code>	For loop
<code>while (condition) { }</code>	While loop
<code>do { } while (condition);</code>	Do-while loop
<code>switch (expression) { case: break;}</code>	Switch-case conditional

Snippet	Description
try { } catch (e) { }	Try-catch block
return value;	Returns a value
break;	Breaks out of loop
continue;	Skips to next loop iteration
Array	
array.push(item)	Adds item to end of array
array.pop()	Removes last item from array
array.map(x => x)	Transforms each array element
array.filter(x => x > 0)	Filters elements based on condition
array.find(x => x === 0)	Finds first match
array.reduce(...)	Reduces array to a single value
array.includes(value)	Checks if value exists in array
array.slice(start, end)	Returns a shallow copy of array
array.indexOf(value)	Returns index of value in array
Object	
Object.keys(obj)	Returns keys of object
Object.values(obj)	Returns values of object
Object.entries(obj)	Returns key-value pairs
Object.assign({}, source)	Copies values into a new object
obj.hasOwnProperty(key)	Checks if object has property

Snippet	Description
String	
<code>str.toUpperCase()</code>	Converts to uppercase
<code>str.toLowerCase()</code>	Converts to lowercase
<code>str.includes("text")</code>	Checks for substring
<code>str.replace("", "")</code>	Replaces text
<code>str.split(",")</code>	Splits string into array
<code>str.trim()</code>	Removes whitespace
<code>str.slice(start, end)</code>	Extracts part of string
Number	
<code>Number(value)</code>	Converts to number
<code>num.toFixed(2)</code>	Formats number to fixed decimals
<code>num.toString()</code>	Converts number to string
Boolean	
<code>Boolean(value)</code>	Converts to boolean
<code>!value</code>	Negation
<code>!!value</code>	Double negation (truth check)
MapSet	
<code>const map = new Map();</code>	Creates new map
<code>map.set("key", value);</code>	Sets key-value pair
<code>map.get("key");</code>	Gets value by key

Snippet	Description
<code>map.has("key");</code>	Checks key existence
<code>map.delete("key");</code>	Deletes key
<code>const set = new Set();</code>	Creates new set
<code>set.add(value);</code>	Adds value to set
<code>set.has(value);</code>	Checks value existence
<code>set.delete(value);</code>	Removes value from set
DateTime	
<code>new Date("")</code>	Creates date from string
<code>dateVar.toISOString()</code>	Converts to ISO string
<code>dateVar.getTime()</code>	Gets time in ms
<code>Date.now()</code>	Gets current timestamp
Timers	
<code>setTimeout(() => {}, 1000);</code>	Delays function execution
<code>setInterval(() => {}, 1000);</code>	Repeats function execution
<code>clearTimeout(timeoutId);</code>	Clears timeout
<code>clearInterval(intervalId);</code>	Clears interval
FileSystem	
<code>fs.readFileSync("", "utf8")</code>	Reads file content
<code>fs.writeFileSync("", "")</code>	Writes to file
<code>fs.appendFileSync("", "")</code>	Appends to file

Snippet	Description
<code>const fd = fs.openSync("", "rw")</code>	Opens file
<code>fs.fsyncSync(fd);</code>	Flushes file content
<code>fs.closeSync(fd);</code>	Closes file
<code>fs.existsSync("")</code>	Checks file existence
<code>fs.readdirSync("")</code>	Reads directory content
Functions	
<code>function myFunction() { }</code>	Declares a function
<code>const add = () => null;</code>	Declares an arrow function
<code>myFunction();</code>	Calls a function

Solución de problemas

Nota: `console.log()` no está disponible en este entorno de scripting. Para detectar errores, revise los registros de errores disponibles en `%APPDATA%/TekLynx/web-forms/logs`

Problema	Solución
El script no se ejecuta	Compruebe si ha hecho clic en "Aplicar" y lo ha vinculado al evento correcto
La validación falla	Revise la sintaxis y asegúrese de que se han declarado todas las variables

Consejos

1. Utilice *"return"* para devolver el resultado al sistema de formularios.
2. Combine varias líneas de código: se admiten todas las funciones de JavaScript.

Uso de las variables de formulario

Puede actualizar directamente las variables del formulario dentro de su script asignándoles un nuevo valor mediante su nombre. Por ejemplo, si una variable del formulario se llama "candy_amount", puede establecer su valor de la siguiente manera:

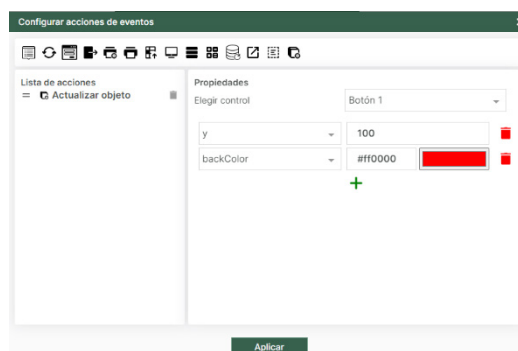
```
candy_amount = 12;
```

El motor de scripts reconocerá automáticamente esta asignación y aplicará el valor a la variable de formulario correspondiente en la fuente de datos una vez finalizada la ejecución del script.

También puede utilizar el resultado de una instrucción "return" de una acción de script en una acción posterior haciendo referencia a "Resultado de la acción anterior" en la configuración de la siguiente acción.

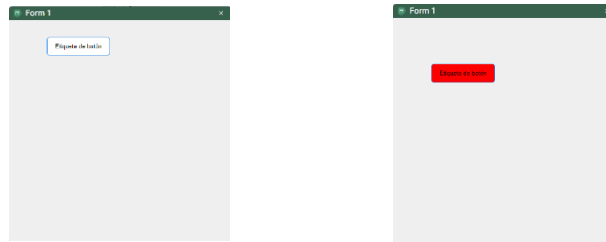
Acción Actualizar objeto

La acción Actualizar objeto permite al usuario cambiar las propiedades de un control seleccionado en tiempo de ejecución cuando se activa un evento. Tras añadir esta acción en el [cuadro de diálogo "Configurar acciones de eventos"](#), el usuario selecciona el control de destino en "Control seleccionado" y define una o varias propiedades que se deben actualizar en la sección "Lista de propiedades".



Esto permite modificar dinámicamente el comportamiento o la apariencia de un control, por ejemplo, cambiando la posición, el tamaño, el texto, el color, la visibilidad u otras propiedades

de valor único compatibles, en función de la interacción del usuario o del resultado de una acción previa.



Limitaciones: esta acción no se puede utilizar para establecer propiedades que requieran varias opciones o colecciones complejas, como los elementos de un cuadro de lista o un grupo de botones de opción. Tampoco se puede utilizar para configurar o cambiar la impresora activa.



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