



Ministerio de Cultura y Educación  
de la Nación  
Universidad Nacional de Cuyo  
Facultad de Ingeniería



# ***INSTITUTO DE AUTOMÁTICA Y ELECTRÓNICA INDUSTRIAL***

## ***SCADA WINLOG LITE GUÍA DE UTILIZACIÓN***

**SEPTIEMBRE DE 2024**

## Índice

|          |  |    |
|----------|--|----|
| 1.       | DESCRIPCIÓN .....                                    | 3  |
| 2.       | SIMULADOR MODBUS TCP/IP .....                        | 4  |
| 3.       | WINLOG LITE.....                                     | 5  |
| 3.1.     | CREACIÓN DE UN NUEVO PROYECTO .....                  | 5  |
| 3.2.     | CONFIGURACIÓN DE DISPOSITIVOS DE ENTRADA/SALIDA..... | 7  |
| 3.3.     | CARGA DE TAGS .....                                  | 10 |
| 3.3.1.   | CARGA DE TAGS ANALÓGICOS.....                        | 11 |
| 3.3.2.   | CARGA DE TAGS DIGITALES.....                         | 15 |
| 3.4.     | CÓDIGO DE PROGRAMA.....                              | 17 |
| 3.5.     | TEMPLATES.....                                       | 19 |
| 3.5.1.   | LEDS.....  | 21 |
| 3.5.1.1. | AGREGAR UN TAG QUE NO SE ENCUENTRA EN LA LISTA.....  | 22 |
| 3.5.2.   | LLAVES.....  | 25 |
| 3.5.3.   | VÚMETROS .....                                       | 26 |
| 3.5.4.   | CUADRO DE TEXTO (EDIT).....                          | 27 |
| 4.       | EJECUCIÓN DEL PROGRAMA .....                         | 29 |
| 5.       | IMPORTAR UN PROYECTO .....                           | 32 |
| 6.       | USUARIOS Y GRUPOS .....                              | 34 |
| 7.       | JAVA WEB TEMPLATES.....                              | 36 |
| 8.       | SMART APPS .....                                     | 42 |
| 8.1.     | CREACIÓN DE UNA SMARTAPP.....                        | 43 |
| 8.2.     | ELEMENTOS DE SOLO LECTURA.....                       | 46 |
| 8.2.1.   | LABELS:.....   | 46 |
| 8.2.2.   | N-STATES LABEL:.....                                 | 46 |
| 8.2.3.   | GAUGE:.....  | 47 |
| 8.3.     | ELEMENTOS DE LECTURA/ESCRITURA.....                  | 48 |
| 8.3.1.   | EDIT.....  | 48 |
| 8.3.2.   | COMBO BOX .....                                      | 49 |
| 8.3.3.   | UPDOWN .....   | 50 |
| 8.3.4.   | N-STATES .....                                       | 51 |
| 8.3.5.   | TOGGLE SWITCH.....                                   | 52 |
| 8.3.6.   | SLIDER.....  | 52 |
| 8.3.7.   | BUTTON.....  | 53 |
| 8.4.     | EJECUCIÓN DE LA SMART APP.....                       | 54 |
| 9.       | GENERACIÓN DE REPORTES. ....                         | 56 |

## WinLog Lite

### 1. DESCRIPCIÓN

Winlog es el software de control supervisado y adquisición de datos (SCADA) que utilizaremos en los trabajos integradores de Industrial. Existen muchos otros, pero este destaca por la poca demanda de recursos en la PC y su disponibilidad libre, adecuado para los trabajos integradores.

Existen actualmente dos versiones: Winlog Lite 3.0 y Winlog EVO 4.0, con interfaz de trabajo similar pero con algunas funcionalidades adicionales en EVO, pensadas para la Industria 4.0.

Se recomienda leer la comparativa de funcionalidades y licencias.

[https://www.sielcosistemi.com/sp/products/winlog\\_scada\\_hmi/winlog-licenses.html](https://www.sielcosistemi.com/sp/products/winlog_scada_hmi/winlog-licenses.html)

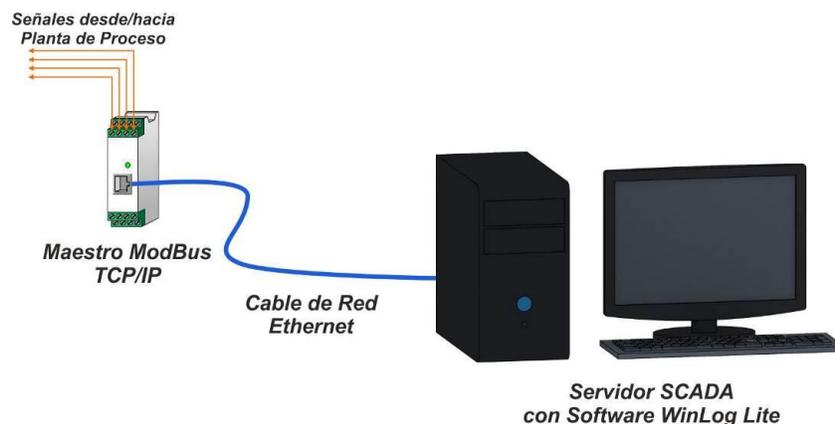
La versión de prueba (Winlog Lite 3.0 o Winlog EVO 4.0) se puede descargar desde el sitio

<https://www.sielcosistemi.com/en/download/public/download.html>

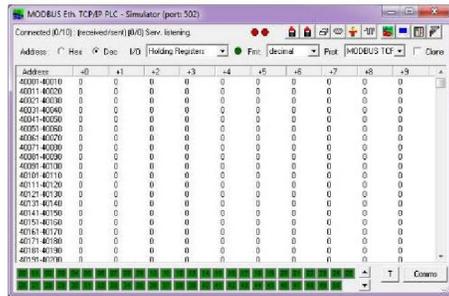
Requiriendo solamente completar datos mínimos.

Nota: La guía está orientada a Winlog Lite, pero la mayoría de la información es idéntica para la versión Winlog EVO.

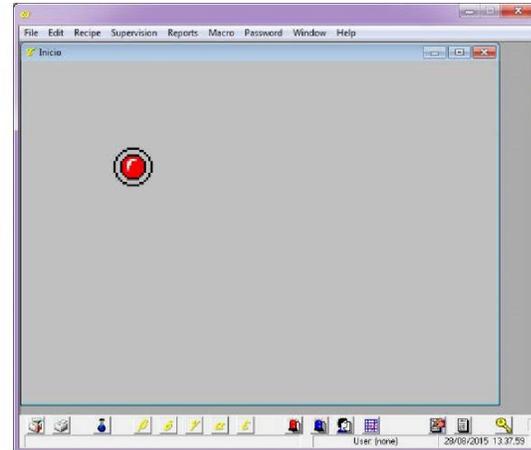
Un esquema de conexión típico para un Maestro ModBus TCP es:



En caso de no disponer de un dispositivo externo es posible utilizar un simulador de ModBus TCP para simular la entrada/salida de datos desde la planta de proceso, e interconectar el mismo con WinLog en forma local (en la misma PC).



Simulador MODBUS TCP



Software WinLog Lite

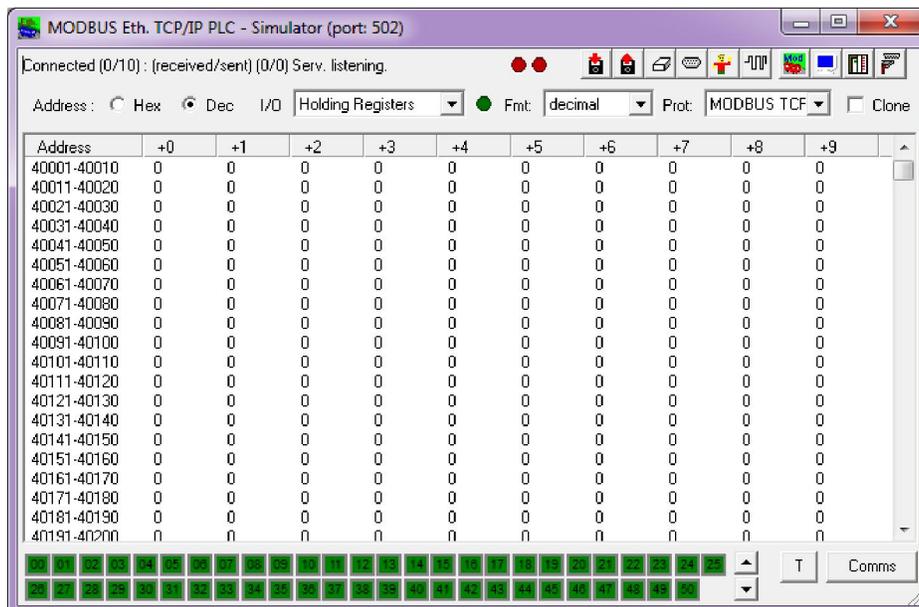
De esta manera se pueden simular entradas/salidas analógicas o digitales que serán procesadas por Winlog. En el ejemplo que se desarrollará a continuación se utilizará este último método.

## 2. SIMULADOR MODBUS TCP/IP

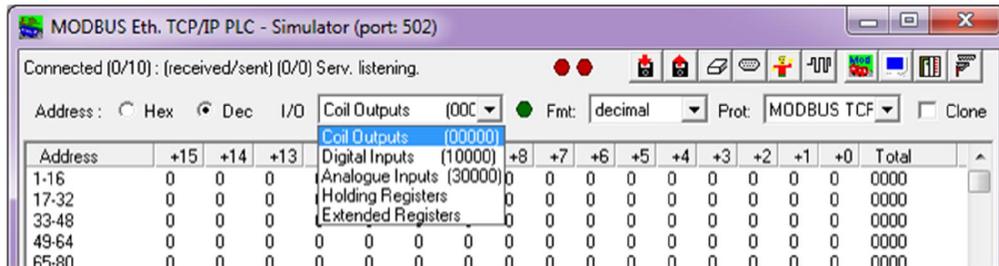
Para este ejemplo se utilizará el Simulador RSsim (Simulador de un PLC Modbus TCP/IP) que se puede descargar del siguiente [enlace](#):

Este simulador permite simular entradas y salidas analógicas y digitales que pueden ser transmitidas mediante Modbus TCP/IP.

Al ejecutar el software se va a ver la siguiente ventana:



El menú I/O permite seleccionar el tipo de entrada/salida que se desea activar.



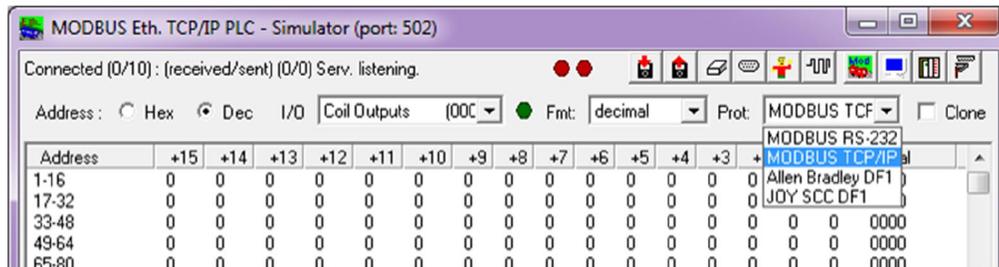
**Coil Outputs:** Actúan como variables Digitales de entrada/salida

**Digital Inputs:** Entradas digitales PLC

**Analogue Inputs:** Entradas analógicas del PLC

**Holding Registers:** variables Analógicas de entrada/salida

En Port se elige **MODBUS TPC/IP**



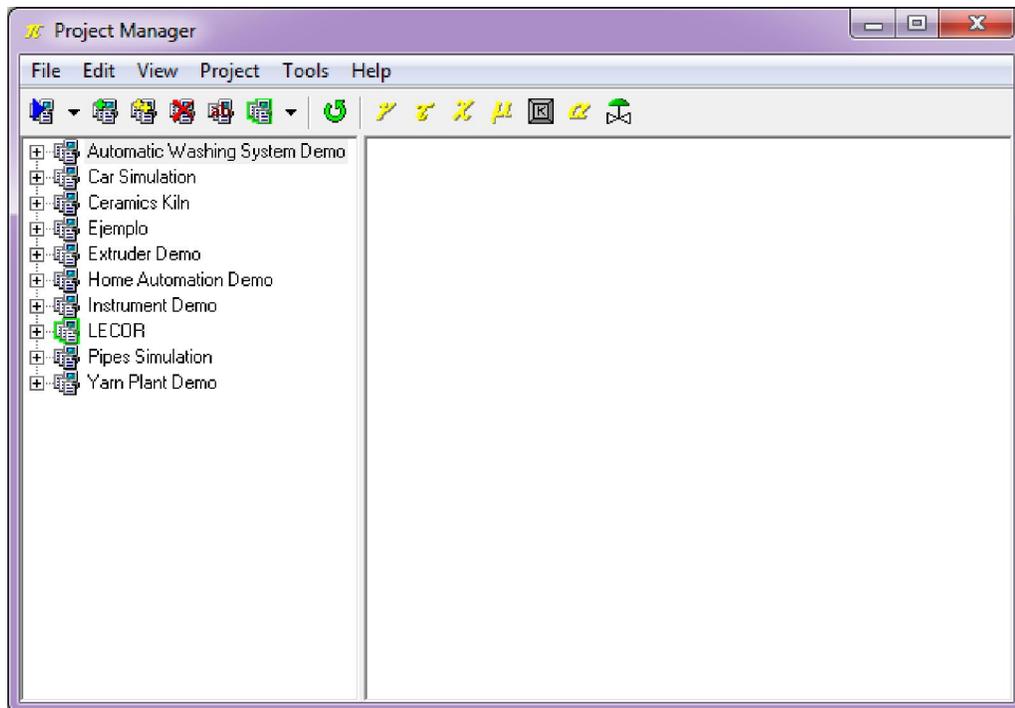
### 3. WINLOG LITE/EVO

#### 3.1. CREACIÓN DE UN NUEVO PROYECTO

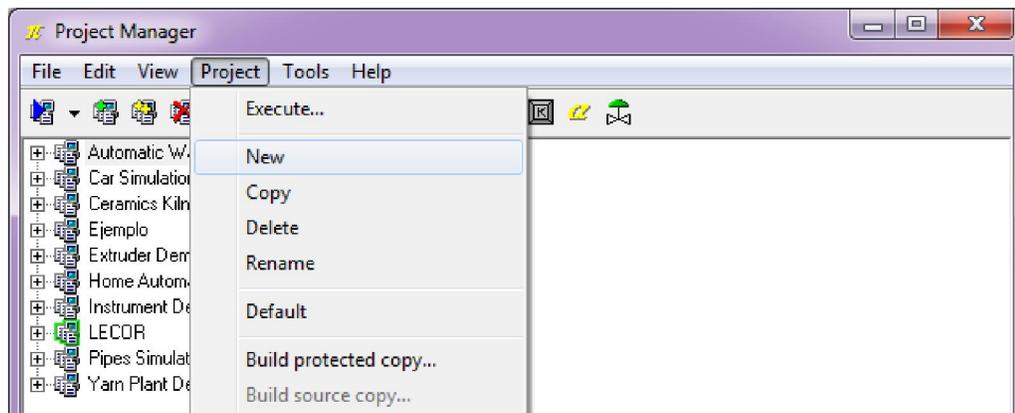
Luego de instalar Winlog ejecutar Project Manager.



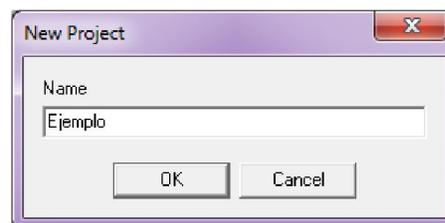
En la ventana que se abre se puede ver en el panel izquierdo todos los proyectos existentes.



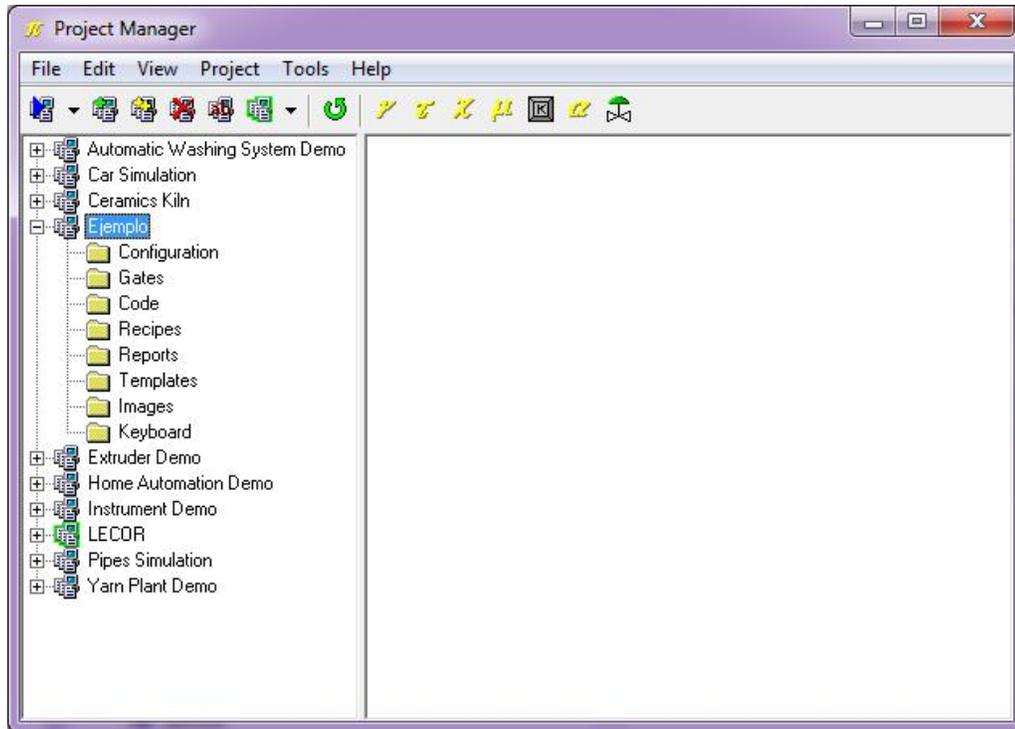
Haciendo clic en el botón  (o en Project → New) se puede crear un nuevo proyecto.



WinLog va a solicitar un nombre para el proyecto, en este caso “Ejemplo”



Con lo cual veremos aparecer el nuevo proyecto Ejemplo con todos sus subítems.



### 3.2. CONFIGURACIÓN DE DISPOSITIVOS DE ENTRADA/SALIDA

En configuración se encuentran las siguientes opciones:

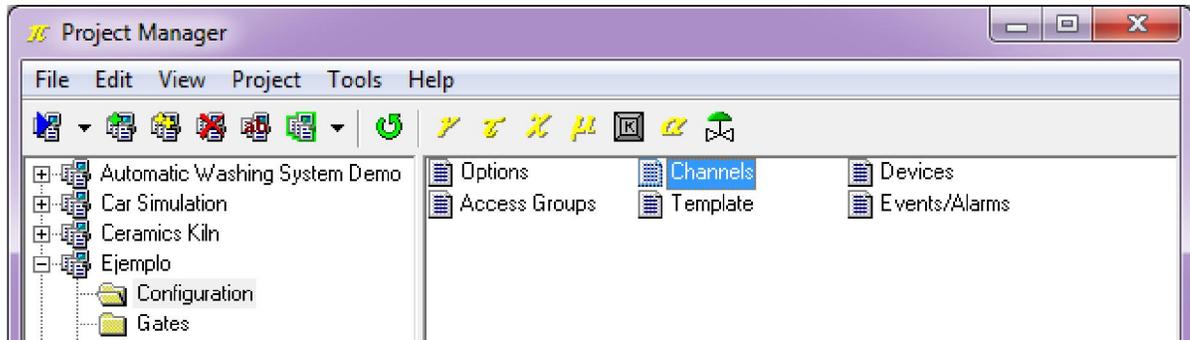
- *Options*
- *Channels*
- *Devices*
- *Access Groups*
- *Template*
- *Events/Alarms*

De estas opciones las que nos interesan son: ***Channels*** y ***Devices***.

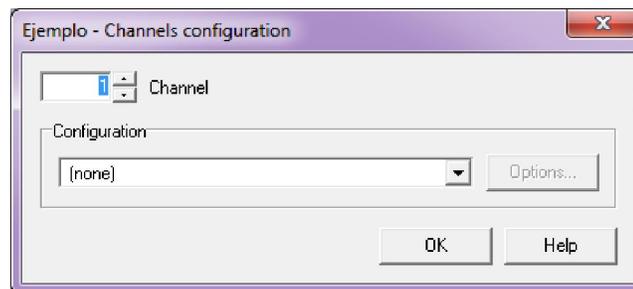
Channels nos permite agregar un canal de comunicación hacia el dispositivo que posee las señales de entrada/salida que deseamos leer.

La forma en que una PC se comunica con el dispositivo pueden ser variadas: a través de un protocolo propietario por ejemplo Profibus, o un protocolo estándar como Modbus, este último puede utilizar como vínculo físico un bus serie RS485 o una red de datos Ethernet.

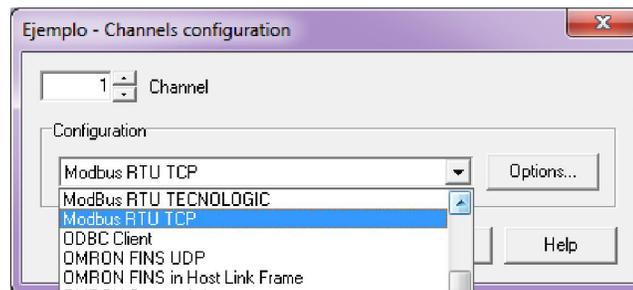
Como mencionamos anteriormente, para este ejemplo, vamos a considerar un vínculo Ethernet, utilizando un simulador Modbus TCP/IP.



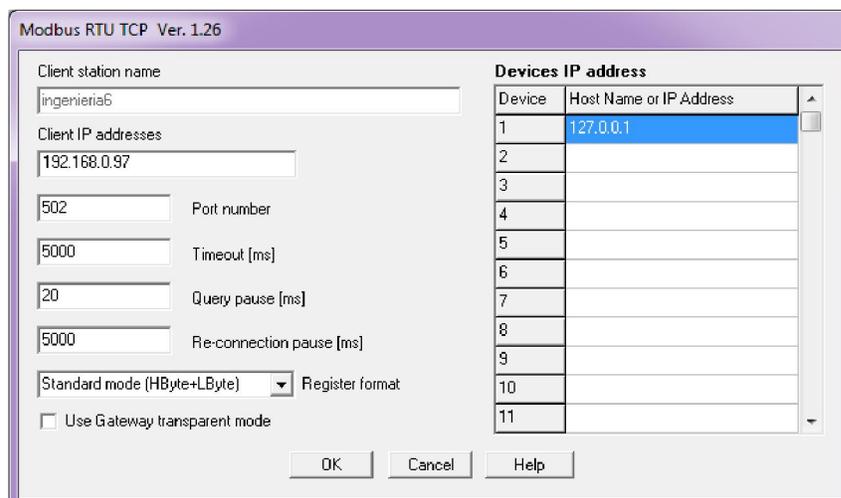
Al hacer doble clic en **Channels** nos encontramos con la siguiente ventana de configuración:



Elegimos como **Canal 1** y como protocolo de datos **Modbus RTU TCP**



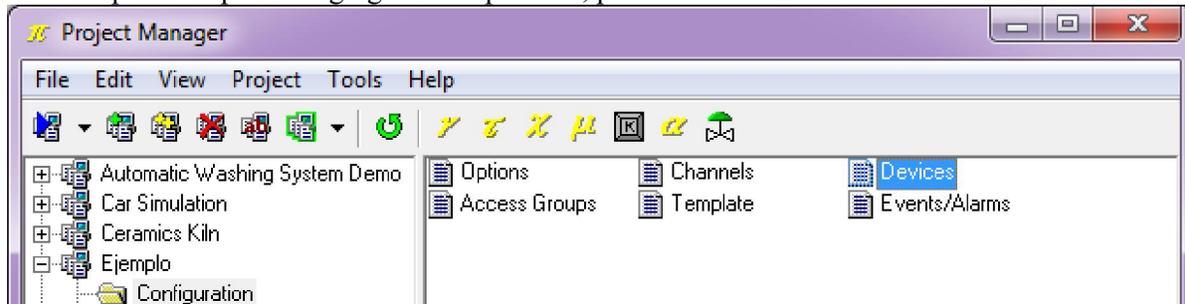
Haciendo clic en el botón **Options** se accede a la ventana de configuración Modbus:



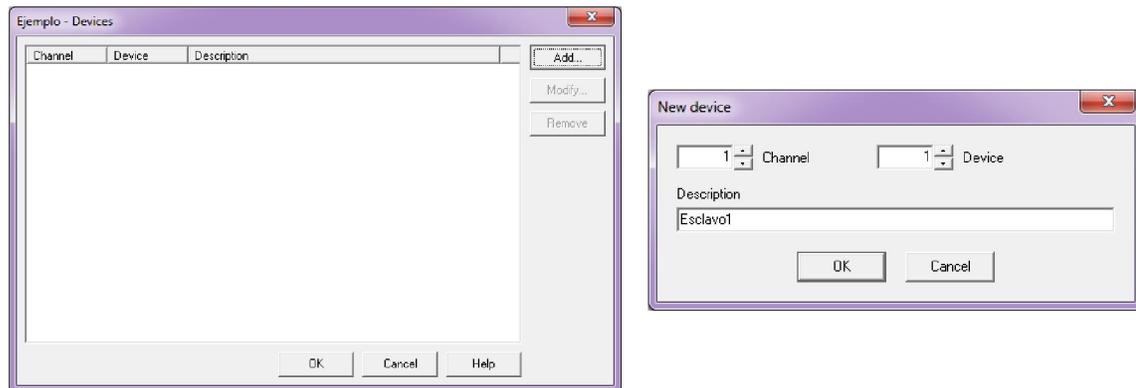
En esta ventana solamente es necesario configurar la dirección IP del dispositivo 1, en este caso por tratarse de un simulador que se ejecuta en la misma PC se utilizará la dirección de red del localhost<sup>1</sup> (127.0.0.1)

El resto de los parámetros se dejan como están y se hace clic en OK.

El próximo paso es agregar un dispositivo, para esto se hace doble clic en **Devices**.



En la ventana que aparece se hace clic en **Add** y se agrega el nuevo Dispositivo. En este ejemplo los valores elegidos son: **Channel: 1, Device: 1 y Descripción: Esclavo1**

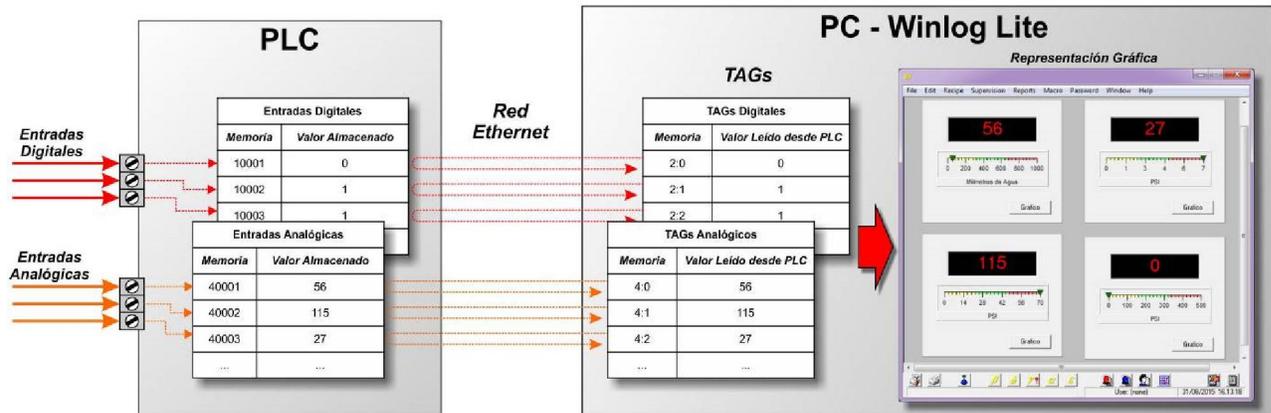


Se hace clic en OK y con esto se finaliza la configuración.

<sup>1</sup> LocalHost hace referencia a la misma PC del usuario. Una interrogación realizada a la dirección 127.0.0.1 le indica a la PC que la consulta es para ella misma, independientemente de si está o no conectada a una red.

### 3.3. CARGA DE TAGS

Los TAGs son variables que tienen su valor asociado con sectores de memoria del dispositivo recolector de los datos del proceso (por ejemplo PLC).



Estos datos son procesados por el software de SCADA para luego ser presentados al operador. Como se vió en el punto 2 en la memoria de un PLC (o en nuestro caso del simulador) la memoria se divide en 4 sectores:

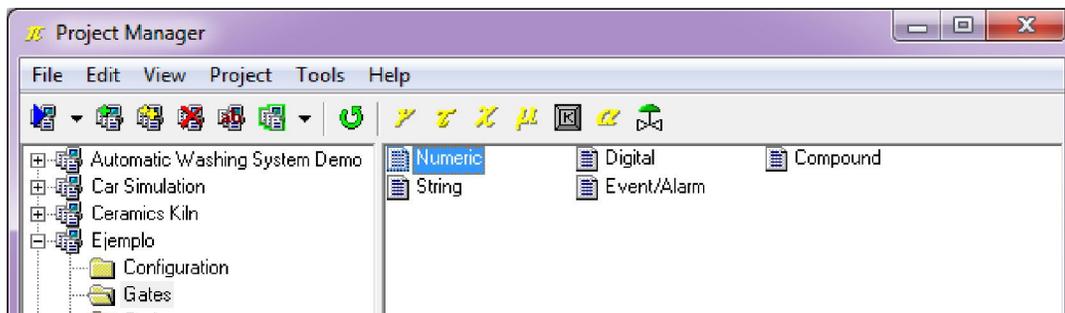
**Entradas digitales:** Es el sector de memoria donde se almacenan los valores leídos en las entradas digitales del PLC, son valores de solo lectura.

**Coil Outputs:** son memorias internas del PLC, donde se pueden almacenar datos digitales, al ser memorias estas pueden ser de lectura y/o escritura. Se utilizan para almacenar valores durante la ejecución de un programa.

**Entradas Analógicas:** En este sector se almacenan los valores de las variables analógicas recibidas por el PLC, son valores de solo lectura.

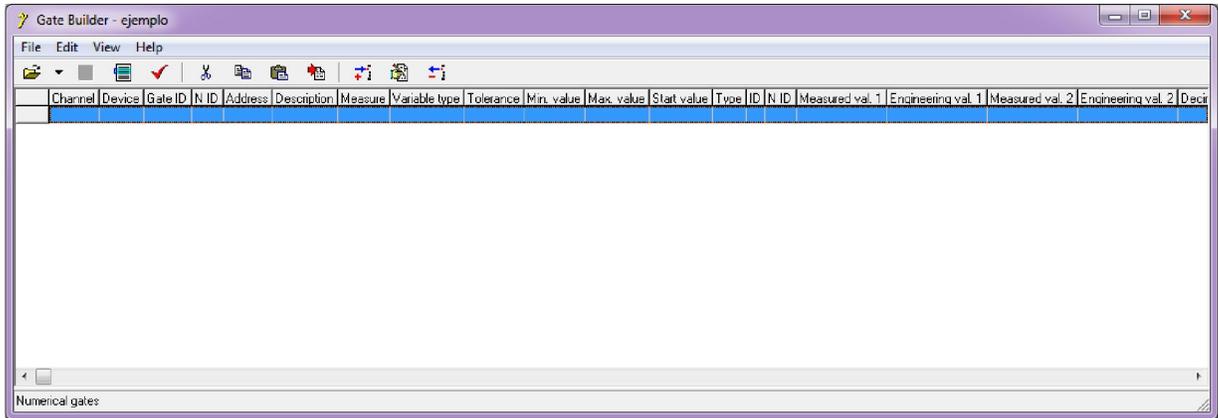
**Holding registers:** son memorias internas que permiten almacenar valores analógicos, son de lectura y/o escritura. Por ejemplo se pueden utilizar para almacenar el valor de temperatura máximo que permite un proceso, contra este valor se compara el valor medido en la entrada analógica y si se excede el mismo se dispara una alarma.

En Winlog Lite utilizamos principalmente dos tipos de TAGs: Digitales (*Digital*) y Analógicos (*Numeric*). Se encuentran dentro de la carpeta GATES.



### 3.3.1. CARGA DE TAGS ANALÓGICOS

Para cargar un TAG analógico el primer paso es hacer doble clic en *Numeric*, se va a abrir a continuación una ventana que muestra el listado de TAGs analógicos (por defecto no existe ninguno).

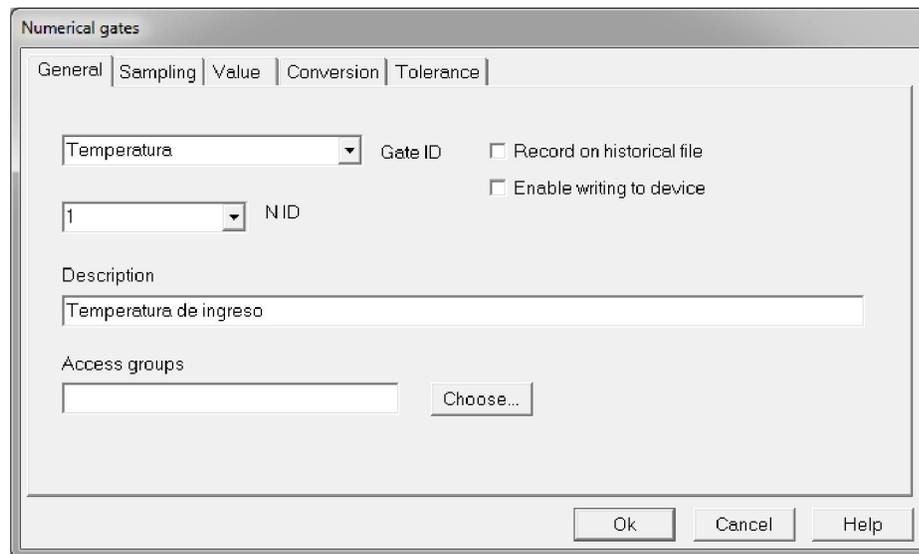


Haciendo doble clic sobre una línea vacía se abrirá la ventana para cargar los datos del TAG, en la lengüeta **General** se cargan los siguientes valores:

**Gate ID:** es el nombre del TAG, en este ejemplo *Temperatura*

**N ID:** es un número que acompaña al Gate ID, en este ejemplo *1*

La combinación de ambos permite cargar TAGs sin duplicación, por ejemplo si se tienen dos temperaturas el primer TAG se llamaría: *Temperatura 1* y el segundo *Temperatura 2*, es decir Gate ID sería el mismo para los dos y se cambiaría el número de ID.



En caso de que el valor de la variable sea de lectura y escritura se debe tildar también “Enable writing to device”.

En la lengüeta **Sampling** se configura:

**Channel:** en este caso es el canal definido en la configuración: 1

**Device:** es el dispositivo que se va a interrogar, también es 1

**Address:** es la dirección de memoria del dispositivo que se va a leer, se conforma de dos partes **X:Y** donde:

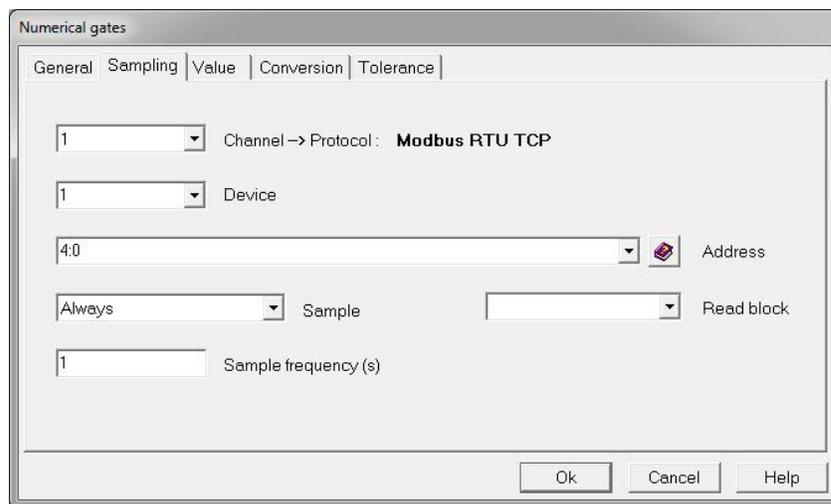
**X:** indica el tipo de entrada a leer, por ejemplo para leer una entrada analógica se indica **4:** y para un Registro de almacenamiento analógica se indica **3:**

**Y:** indica la posición de memoria que debe ser leída, por ejemplo la primera posición es **0**

En nuestro ejemplo utilizamos **4:0**, donde **4:** indica que se quiere leer una entrada analógica y **0** corresponde a la primera dirección de memoria.

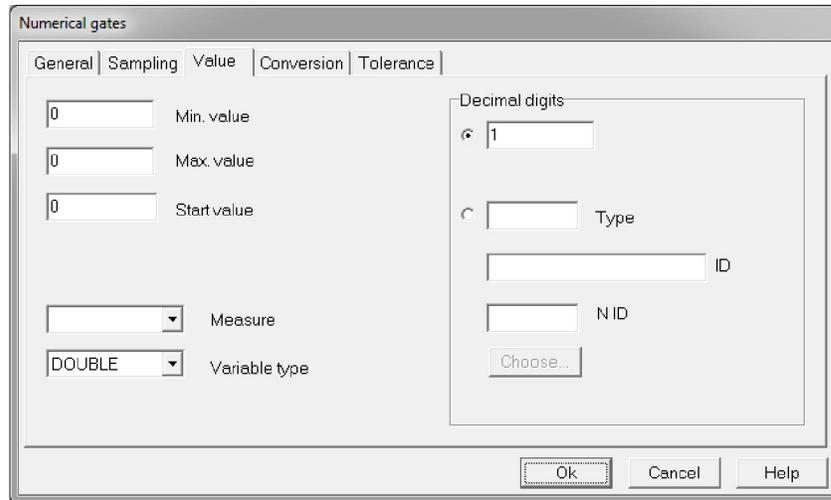
**Sample:** indica en qué forma se va a realizar la toma de datos, habitualmente se elige siempre (*always*)

**Sample frequency:** es la frecuencia, en segundos, con la que se va a realizar el muestreo, en este ejemplo se ha indicado 1 segundo.



En la lengüeta **Value** se pueden especificar:

- *Valor mínimo a leer*
- *Valor máximo*
- *Valor de inicio*
- *Cantidad de decimales*
- *Tipo de variable*



En nuestro ejemplo se deja como está.

En la lengüeta **Conversion** es posible especificar una conversión de escalas, esto es útil ya que en un dispositivo real los valores de ingreso no siempre corresponden con los valores mínimos y máximos del sensor utilizado, por ejemplo pueden ser de 0 a 100 donde 0 equivale a 10 grados Celsius y 100 equivale a 300 grados Celsius, en este caso se indicaría:

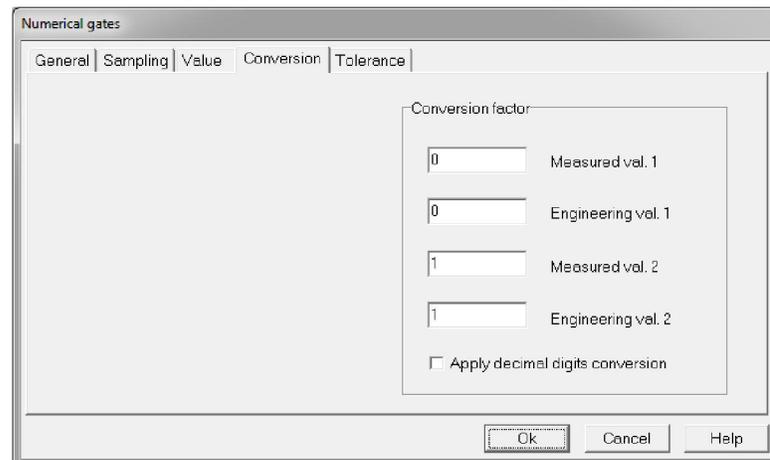
Measured val. 1: 0 (valor medido mínimo)

Engineering val. 1: 10 (valor a representar para el mínimo)

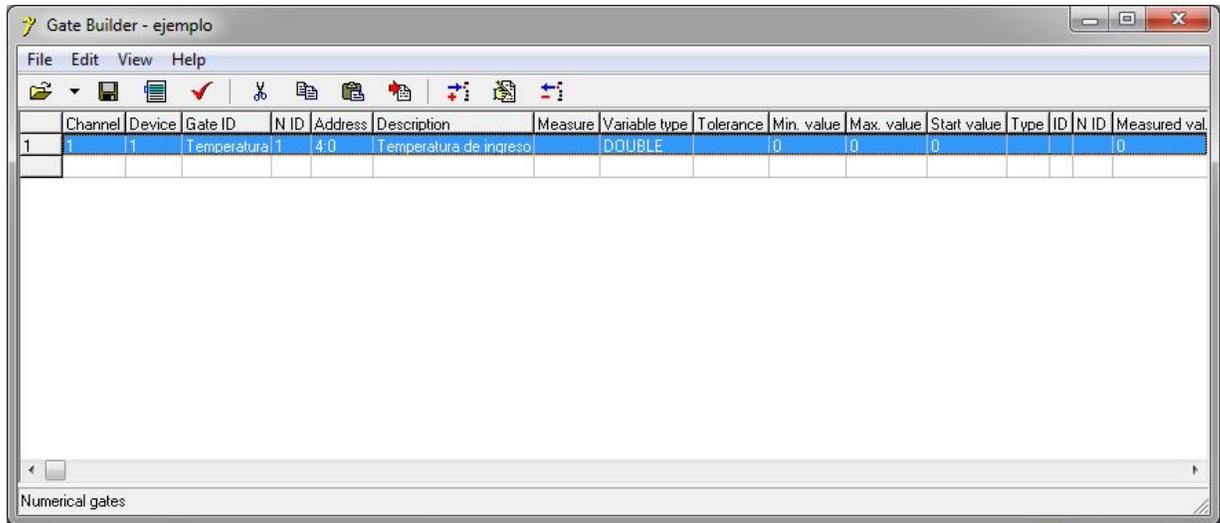
Measured val. 2: 100 (valor medido máximo)

Engineering val. 2: 300 (valor a representar para el máximo)

En nuestro ejemplo no se aplican conversiones por lo que quedan como están.



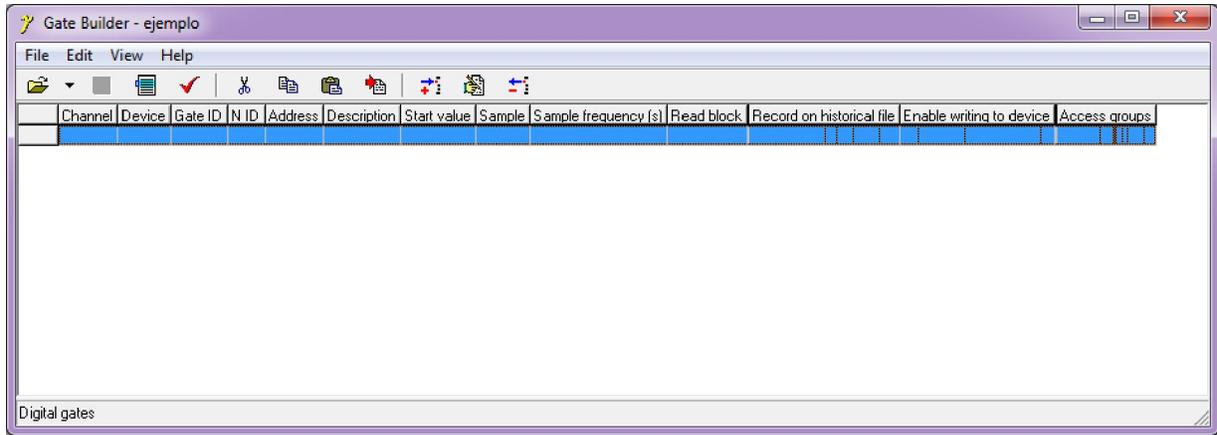
Luego de hacer OK se vuelve a la pantalla inicial, donde aparecerá ahora el TAG creado.



Finalmente se deben guardar los cambios realizados haciendo clic en File → Save Current Table o cerrando la tabla y eligiendo la opción de guardar los cambios realizados.

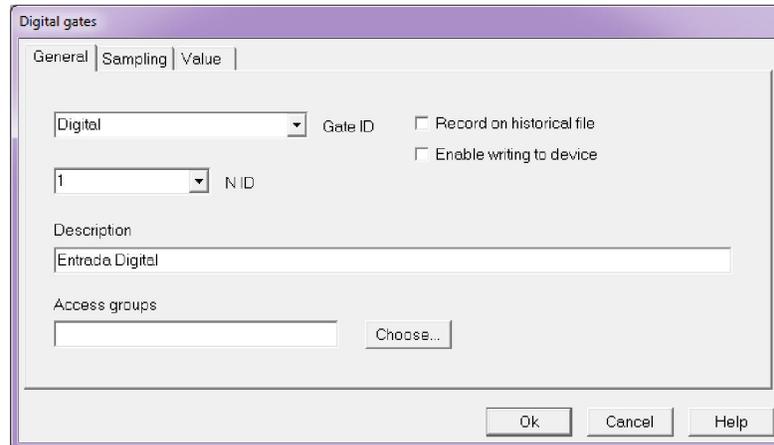
### 3.3.2. CARGA DE TAGS DIGITALES

La carga de TAGs digitales se realiza de una manera muy similar a los analógicos. Haciendo doble clic sobre Digital se abre la ventana que muestra los TAGs digitales.



Haciendo doble clic sobre una línea en blanco aparece la ventana de configuración para el nuevo TAG.

En la lengüeta **General** se elige el Nombre para el TAG (*Gate ID*) y un número que se asociará al mismo (*N ID*). En caso de que el TAG deba ser de lectura y escritura se debe tildar la casilla “*Enable writing to device*”.



En la lengüeta Sampling se indica el número de canal (Channel) y el dispositivo (Device)<sup>2</sup> desde donde se quieren leer los datos.

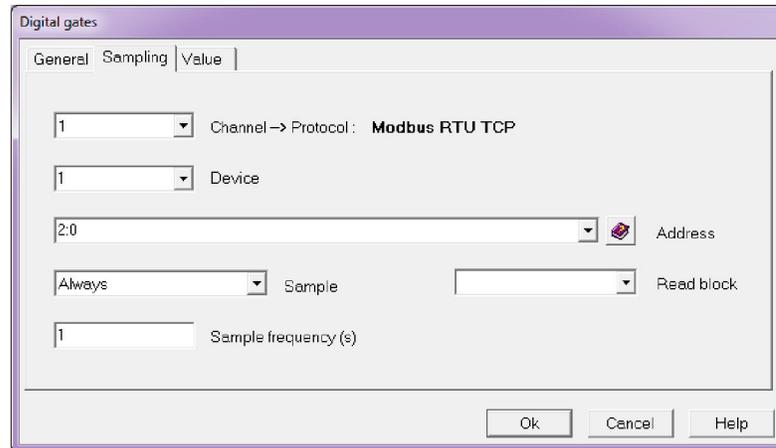
La dirección de memoria (Address) en este caso se va a expresar como:

**1:XX** en caso de que se quieran leer o escribir Memorias digitales (Coil Registers)

**2:XX** en el caso de que se quieran leer Entradas Digitales

En nuestro ejemplo vamos a leer la primera entrada digital por lo que indicamos 2:0

<sup>2</sup> Según lo definido en el punto 3.2 Configuración General



Digital gates

General | Sampling | Value

1 Channel → Protocol: **Modbus RTU TCP**

1 Device

2:0 Address

Always Sample Read block

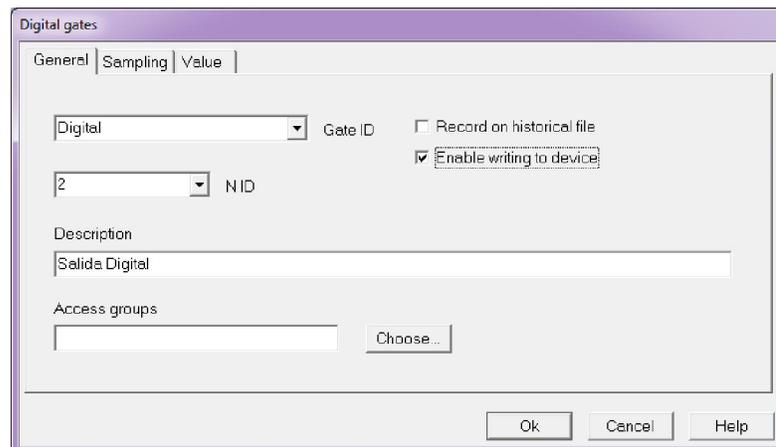
1 Sample frequency (s)

Ok Cancel Help

En *Sample* elegimos *Always*, para que muestree los valores en forma permanente y en frecuencia elegimos 1 segundo.

Finalmente hacemos clic en OK y regresamos a la ventana principal. En esta ventana guardamos los datos y la cerramos (si se cierra y se han realizado cambios va a aparecer una ventana que consulta si se quieren guardar los cambios antes de cerrar).

Para crear una salida digital el procedimiento es el mismo, en este caso debemos asociarlo con una memoria digital (Coil Output) y tildar la opción *Enable writing to device*.



Digital gates

General | Sampling | Value

Digital Gate ID  Record on historical file

2 N ID  **Enable writing to device**

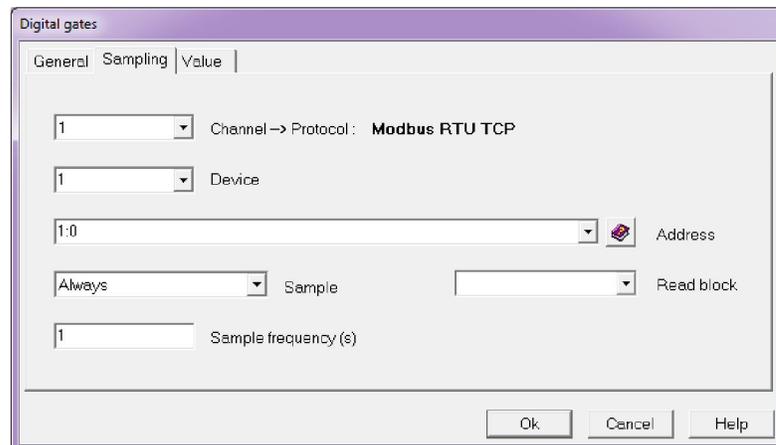
Description

Salida Digital

Access groups

Choose...

Ok Cancel Help



Digital gates

General | Sampling | Value

1 Channel → Protocol: **Modbus RTU TCP**

1 Device

1:0 Address

Always Sample Read block

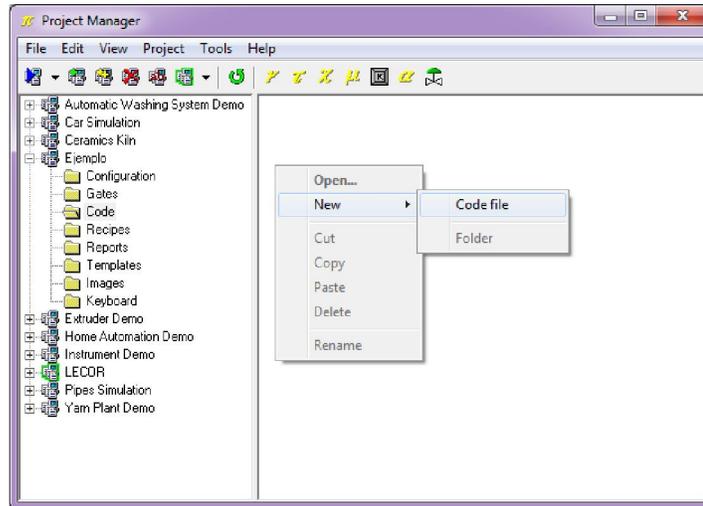
1 Sample frequency (s)

Ok Cancel Help

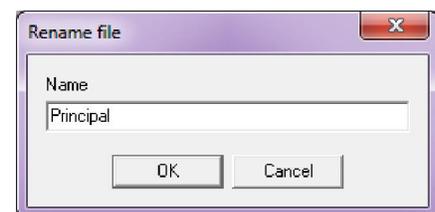
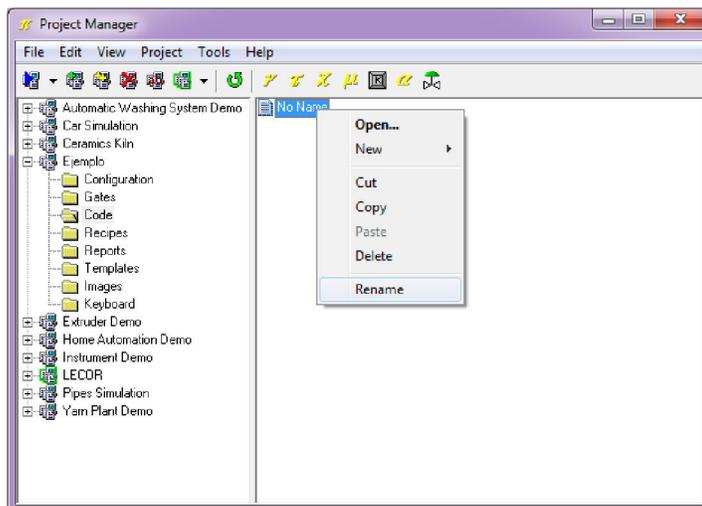
### 3.4. CÓDIGO DE PROGRAMA

Winlog posee un lenguaje de programación similar a Basic que permite realizar operaciones más complejas que las que se pueden conseguir normalmente. Si bien en nuestro ejemplo no vamos a utilizar este tipo de operaciones es necesario ejecutar al menos un comando, es el que indica a Winlog que **Template** (Gráfico) debe abrir al inicio.

Para esto creamos un archivo de programa, seleccionando la carpeta **Code** hacemos clic con el botón derecho en el panel de la derecha y elegimos **New → Code file**.



Va a aparecer un nuevo archivo llamado **No Name**, para colocarle un nombre hacemos clic sobre el archivo con el botón derecho y elegimos **Rename**, en la ventana que aparece colocamos el nombre que deseamos, en el ejemplo es: **Principal**



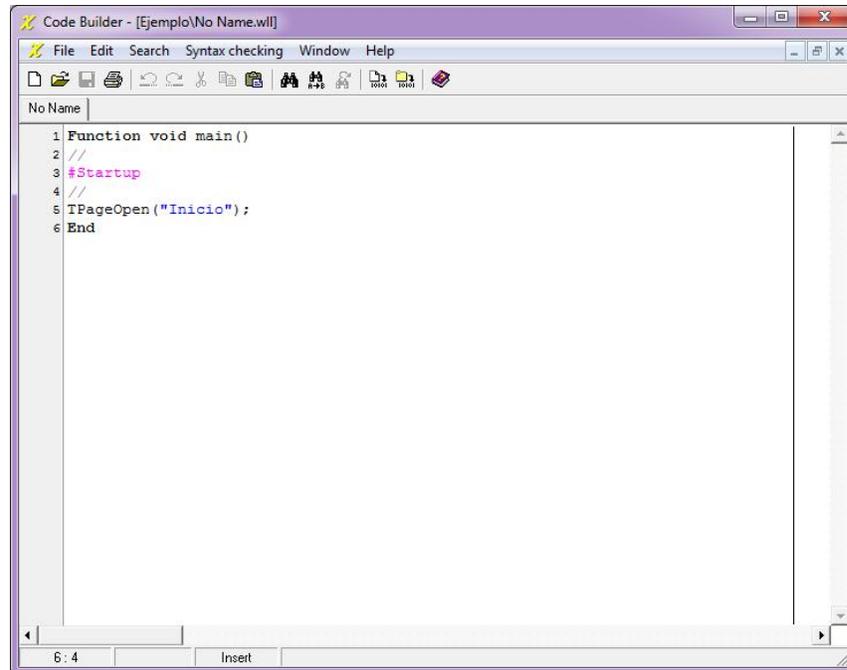
Para editar el contenido del archivo hacemos doble clic sobre el mismo y se abre la ventana de programación, en nuestro caso el programa a utilizar es:

**Function void main ( )**

**#Startup**

**TPageOpen**("Nombre del Template a abrir");

**End**



**Function void main:** declara la función principal

**#Startup:** indica que se ejecutará una sola vez al inicio del programa

**TPageOpen:** indica que Template deseamos abrir, en este caso Inicio

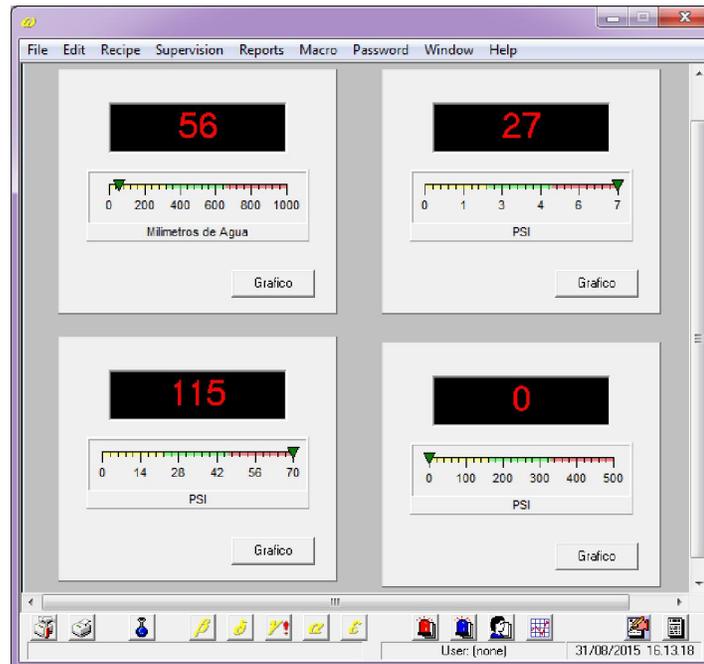
**End:** Indica el final de la función

Una vez finalizado el programa lo guardamos y cerramos la ventana.

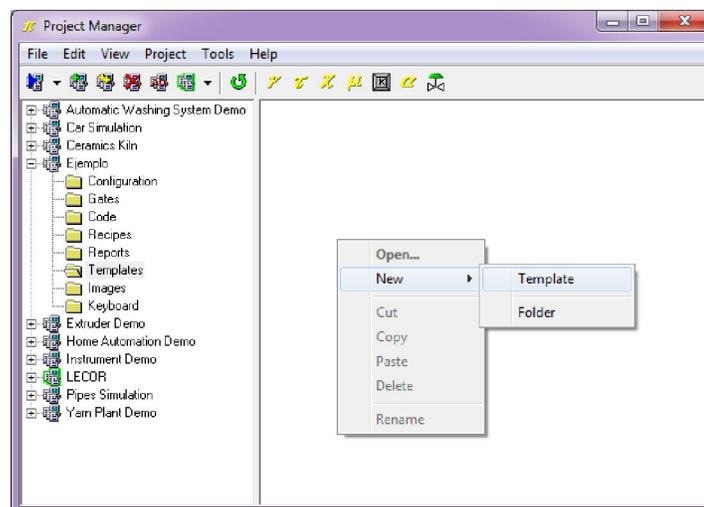
### 3.5. TEMPLATES

Los Templates son las ventanas que se utilizan para visualizar e interactuar con los datos del proceso, por ejemplo valores, gráficos, botones, etc.

Cada uno de estos elementos es un Objeto que se asocia con alguno de los TAGs definidos anteriormente.



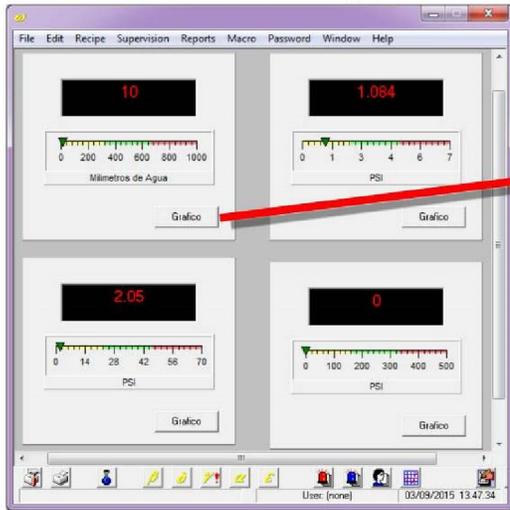
Para crear un Template seleccionamos la carpeta Templates y en el panel de la derecha hacemos clic con el botón derecho y elegimos **New → Template**



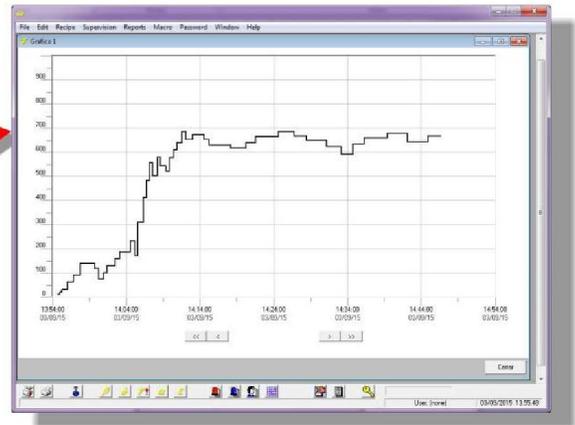
El archivo que aparece se llama por defecto **No Name**, para darle un nombre hacemos clic con el botón derecho sobre el archivo y elegimos la opción **Rename**.

Es posible disponer de varios Templates para un mismo proyecto, uno es el principal que se abre al inicializar el programa y los demás son complementarios. Por ejemplo:

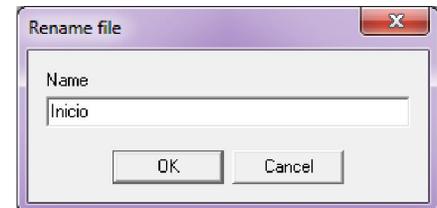
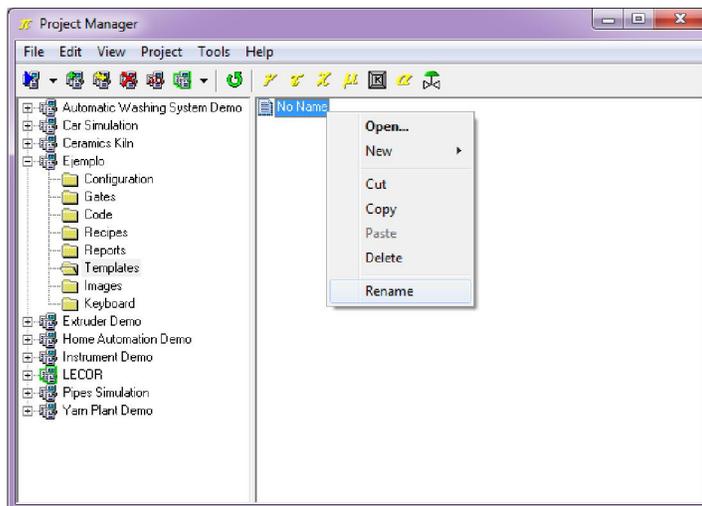
### Template Principal



### Template Auxiliar



Es importante que el nombre del Template que queremos que se abra por defecto corresponda con el que indicamos en el Código de programa (punto 3.4) en nuestro caso es **Inicio**

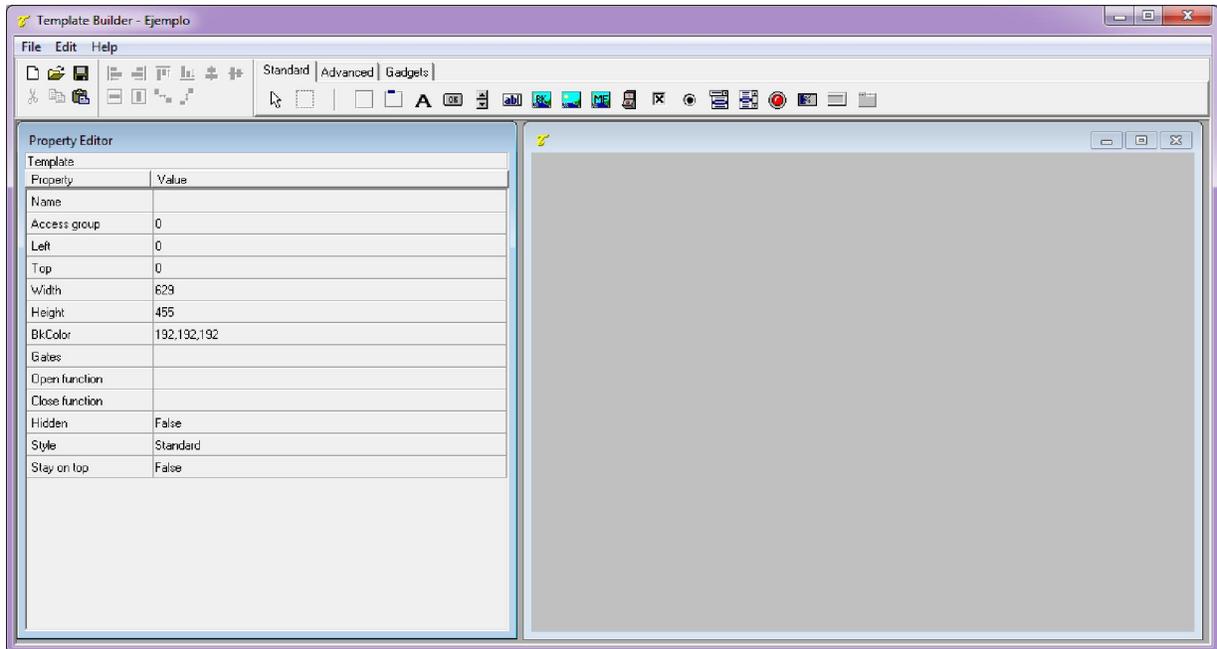


Para modificar el Template hacemos doble clic sobre el nombre del archivo y se abre la ventana de edición.

Esta ventana está organizada de la siguiente manera:

A la izquierda se encuentra la ventana de propiedades, en esta podemos ver la propiedades del objeto seleccionado, por ejemplo, color, tamaño, TAG asociado (Gates), etc.

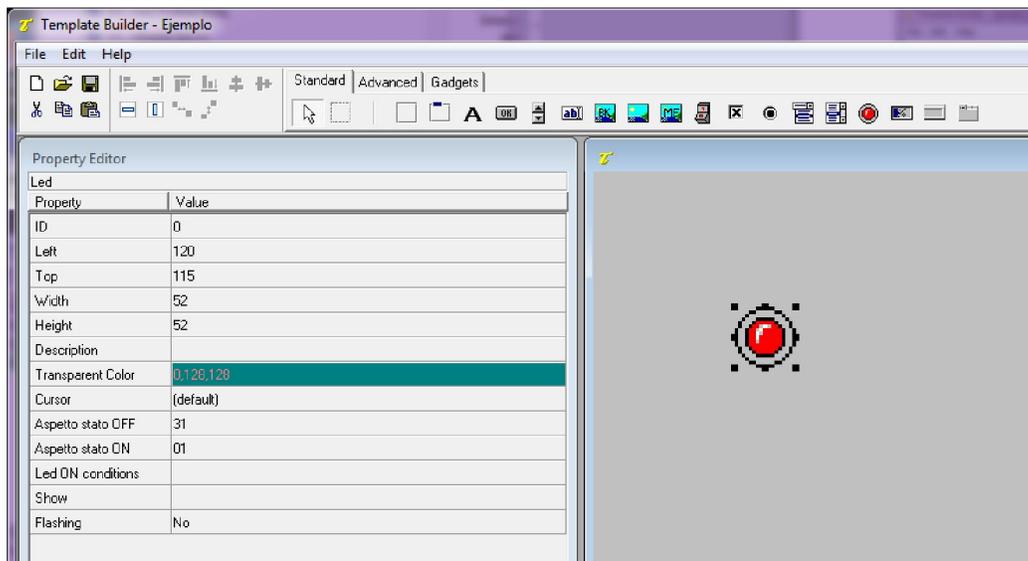
En la parte superior podemos ver tres lengüetas, la que utilizaremos es la Standard, en esta podemos ver los objetos de uso habitual: recuadros, texto, botones, listas, LEDs, barras de nivel (tipo vóumetro), etc.



En esta imagen, la ventana de propiedades está mostrando las propiedades del template en general (está seleccionado el fondo).

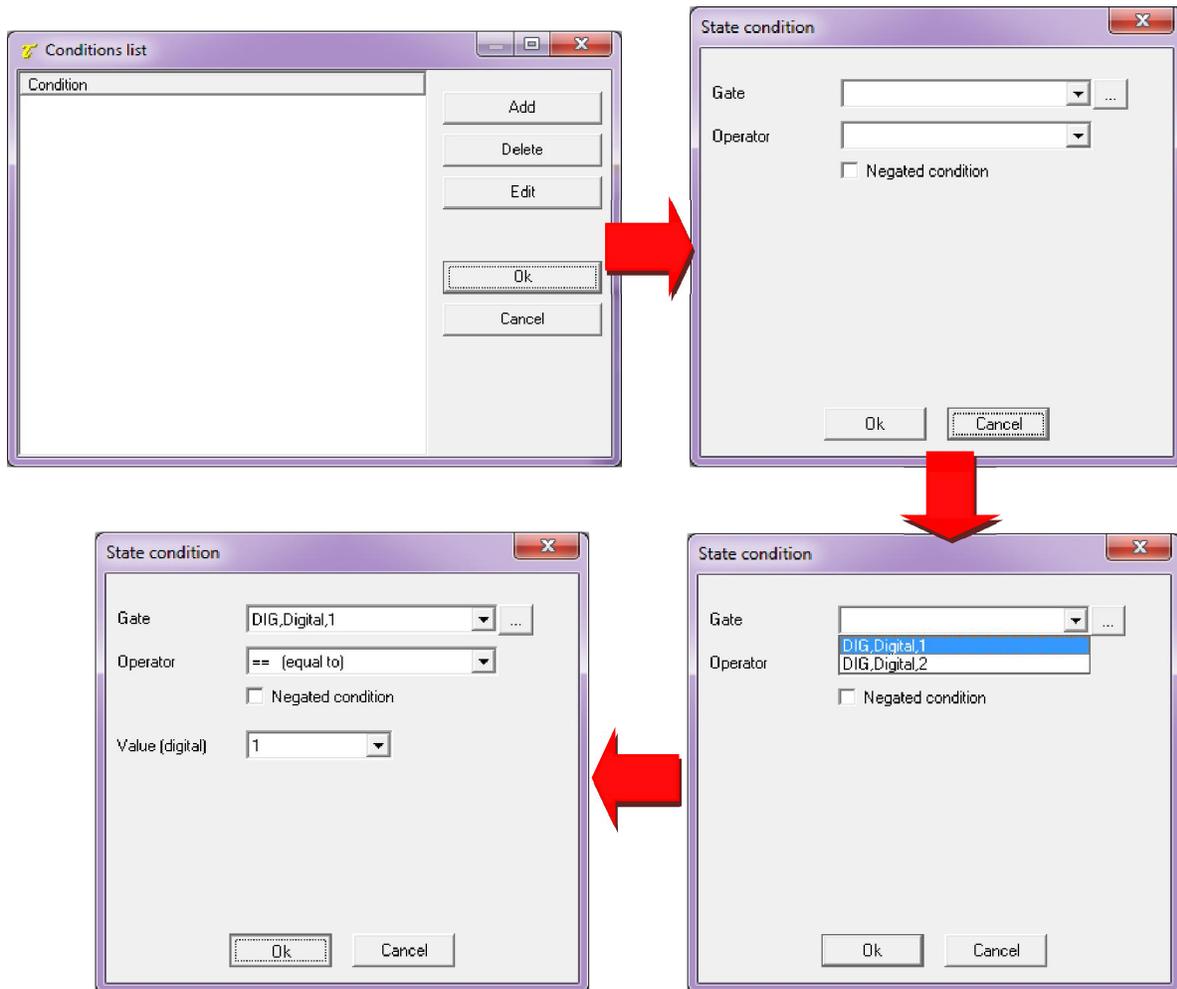
### 3.5.1. LEDs

Vamos a asociar un LED con el TAG: *Digital1* que creamos en el punto 3.3.2. El primer paso es hacer clic sobre el botón de LED  y luego se hace clic sobre el template, va a aparecer un LED al cual le podemos modificar el tamaño y la ventana de propiedades va a mostrar las propiedades del LED.



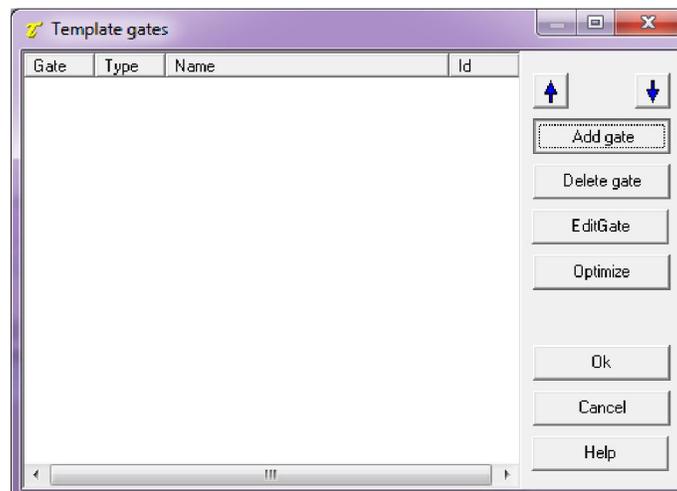
Para asociar este LED con el TAG *Digital1* debemos hacer clic en **Led ON Conditions** (botón ), y en la ventana que aparece seleccionamos la condición de activación, en este caso queremos que el LED se active cuando el estado de la entrada *Digital1* sea 1 por lo que hacemos clic en el

botón Add y en la ventana que aparece elegimos como Gate: *Digital1*, como operador = = y como valor 1.

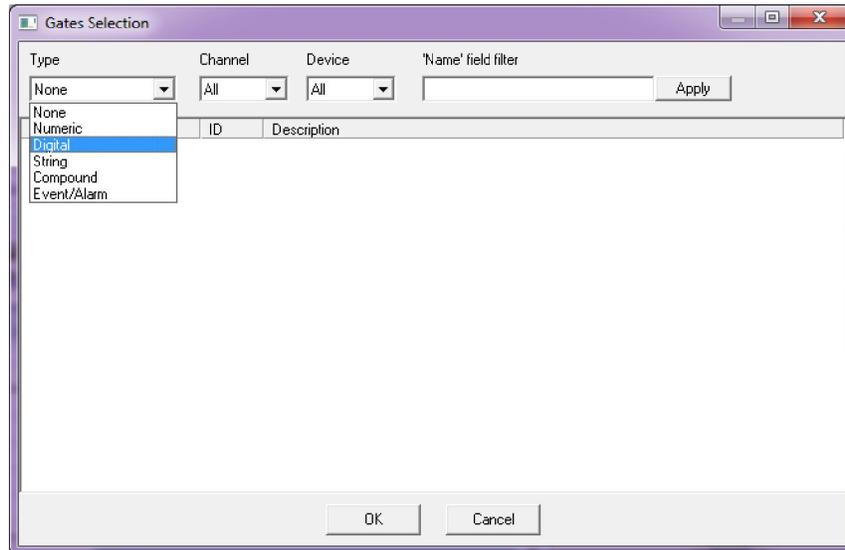


### 3.5.1.1. AGREGAR UN TAG QUE NO SE ENCUENTRA EN LA LISTA

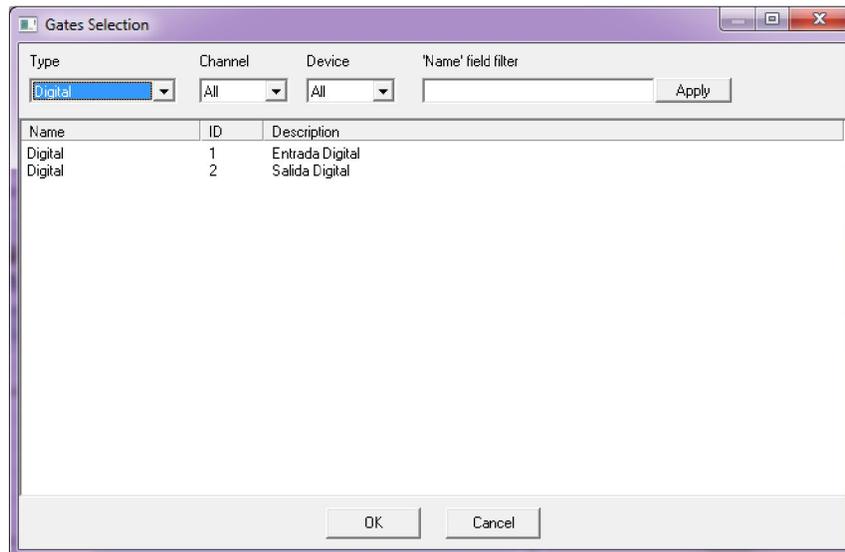
Es posible que en el listado de **Gates** no aparezca el TAG buscado, para agregar el TAG a la lista se hace clic en el botón  y en la ventana que aparece elegimos **Add gate**



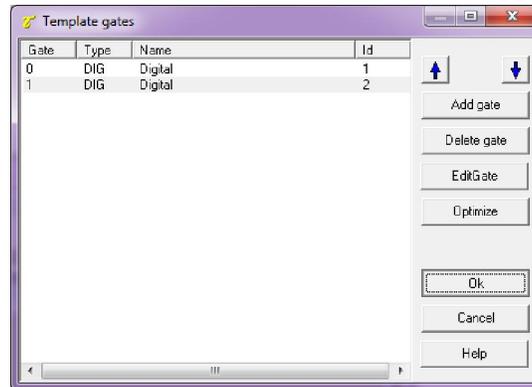
En la ventana que aparece seleccionamos el tipo de TAG que queremos agregar a la lista, en este caso Digital.



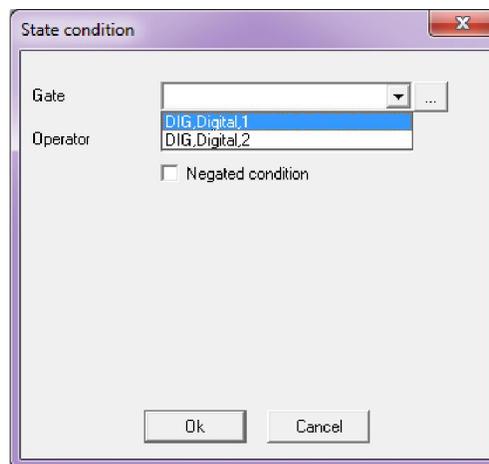
Con esto nos va a mostrar todos los TAGs digitales que hayamos creado, de esta lista elegimos el TAG que nos interesa, si lo deseamos podemos seleccionar y agregar más de un TAG, y hacemos clic en OK.



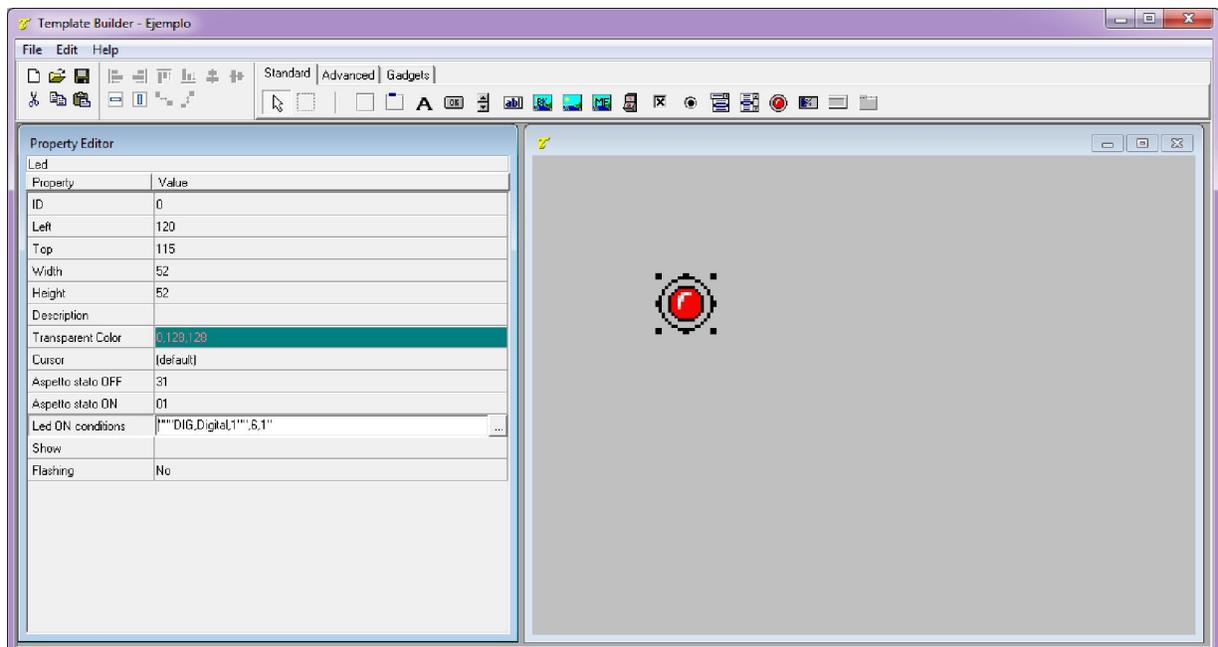
Vamos a volver a la ventana anterior, donde ahora van a figurar todos los TAGs que hayamos agregado.



Elegimos entonces el TAG *Digital1* y hacemos clic en **Ok**, a partir de este momento el TAG aparecerá en la lista de selección.



Luego de hacer clic en OK volveremos a la ventana del template donde veremos que *Led ON conditions* muestra el TAG y la condición de activación.

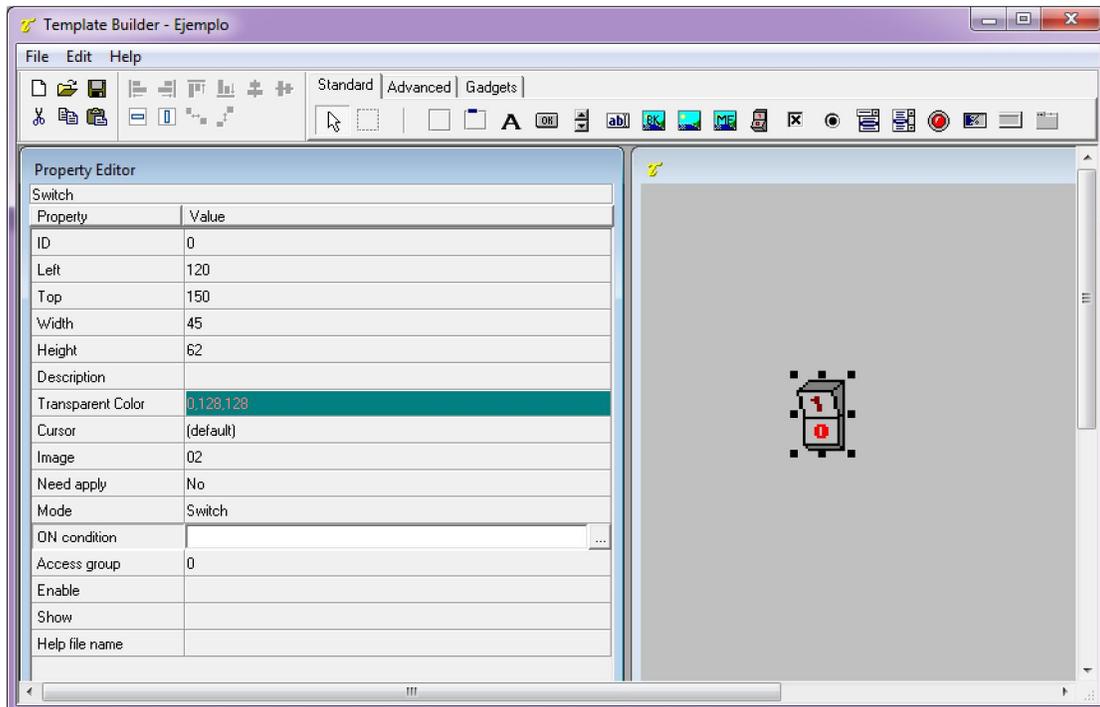


Con esto el LED ha quedado asociado con el TAG Digital1, a partir de ahora si ejecutamos el programa veremos que se abre la ventana con el LED y el mismo reflejará el estado de la entrada digital 1.

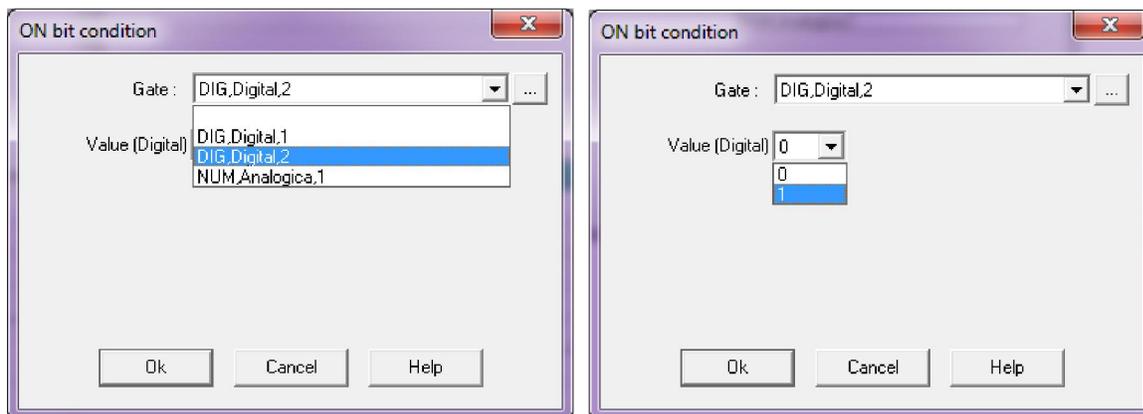
### 3.5.2. LLAVES

Siguiendo un procedimiento similar podemos agregar un botón que envíe información a un TAG definido como entrada/salida digital.

Haciendo clic sobre el botón  y luego sobre el template aparecerá un botón, al que relacionaremos con un Coil Output, para nuestro ejemplo será el TAG *Digital2*



Para asociarlo con el TAG se sigue el mismo procedimiento anterior, haciendo clic en ON Condition se selecciona el TAG y la condición de activación:

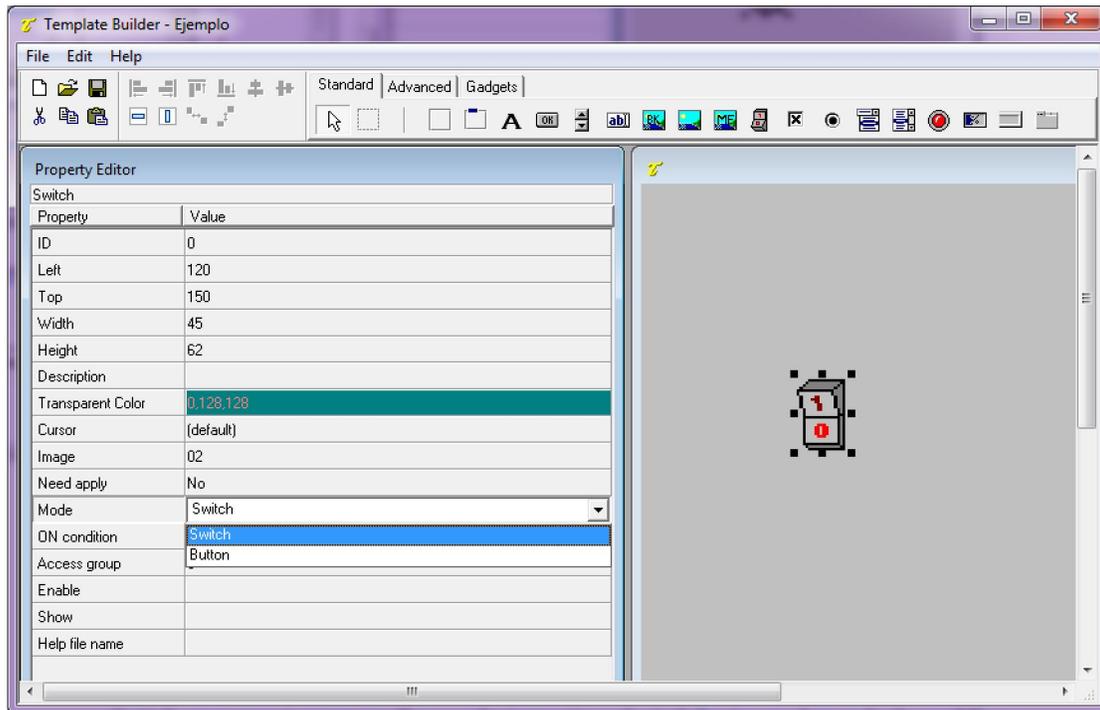


En caso de que el TAG no figure en la lista se sigue el mismo procedimiento descrito anteriormente (3.5.1.1) para agregarlo.

En el caso particular del botón se puede elegir el comportamiento con la propiedad Mode:

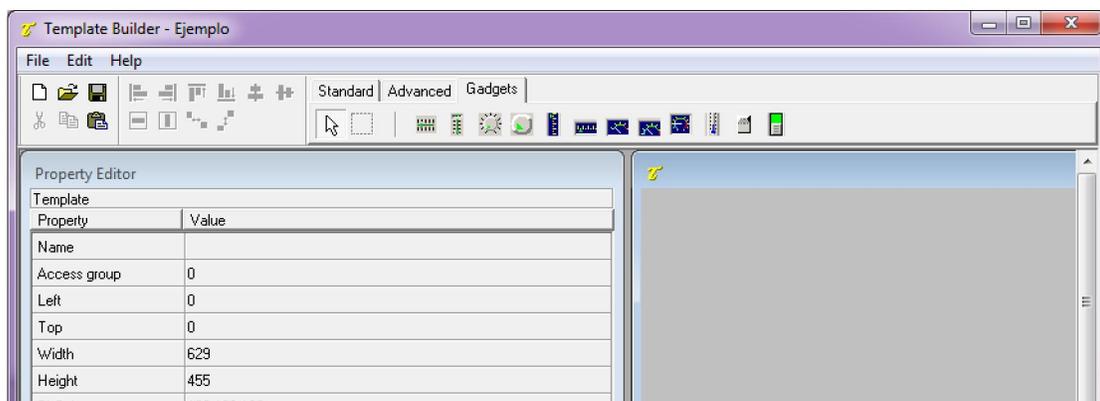
**Switch:** se comporta como una llave, es decir mantiene el estado hasta que el operador lo modifica

**Boton:** se mantiene activado mientras se lo pulsa y desactiva en el momento que se deja de pulsar.

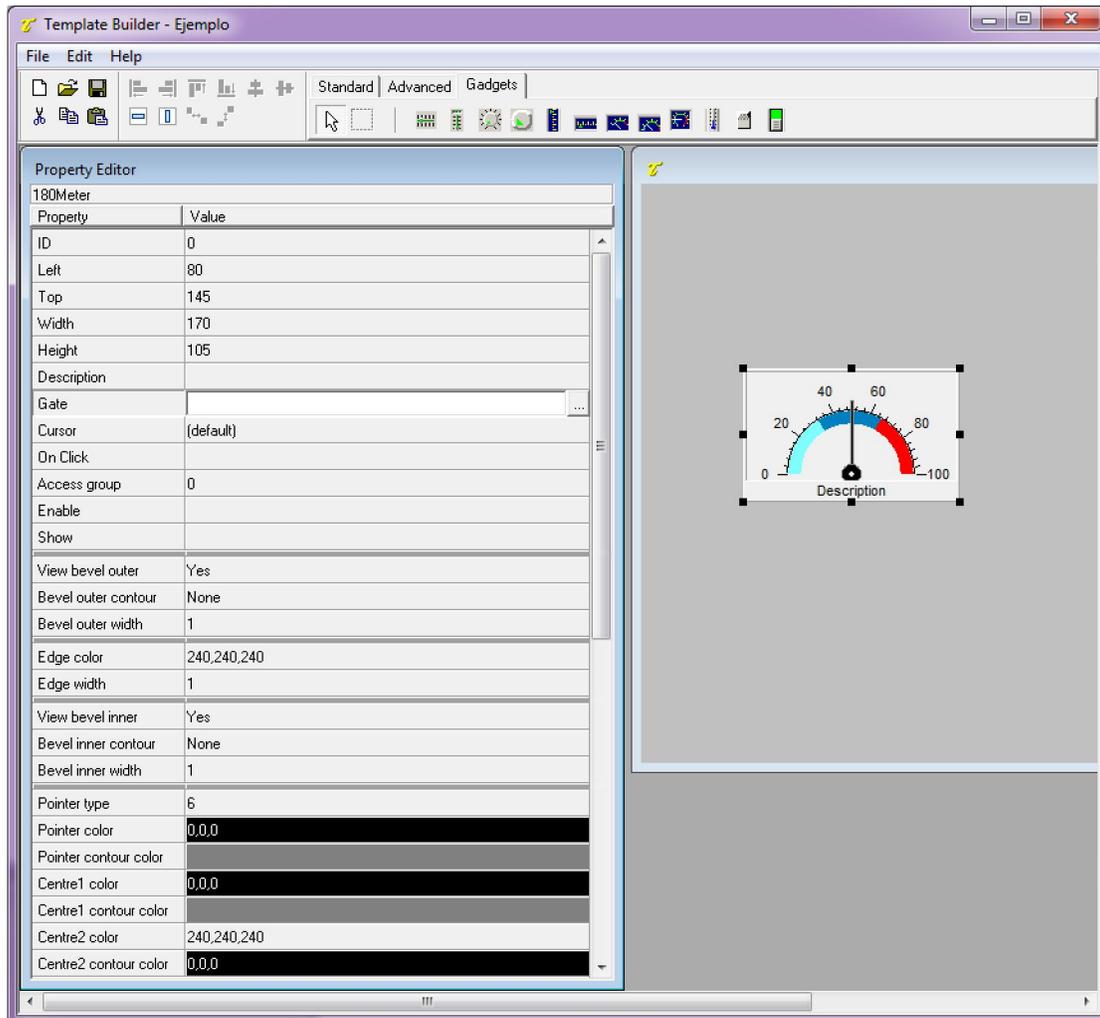


### 3.5.3. VÚMETROS

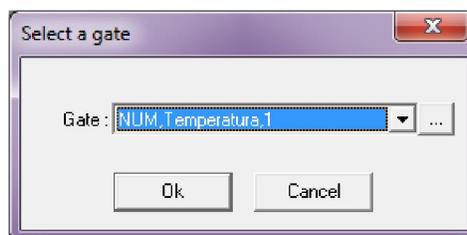
Se pueden agregar indicadores tipo vúmetros para representar valores analógicos, el procedimiento es el mismo que se ha descrito anteriormente para valores digitales. Estos se encuentran en la lengüeta **Gadgets**.



Para este ejemplo elegimos 180 Meter  y hacemos clic en el template. Igual que en los casos anteriores debemos asociar el objeto con un TAG, en este caso analógico. Vamos a utilizar el TAG creado en el punto 3.3.1 llamado *Temperatura1*



Haciendo clic en **Gate** se va a abrir la ventana donde elegimos el TAG, en caso de que no se encuentre en la lista debemos agregar el mismo como se indicó en el punto 3.5.1.1, esta vez eligiendo **Numeric** para que muestre el listado de TAGs Analógicos.

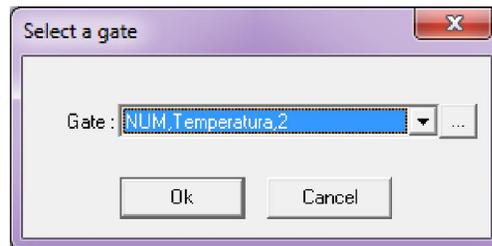
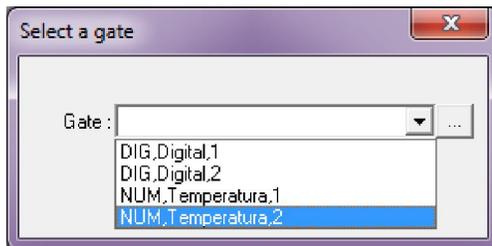
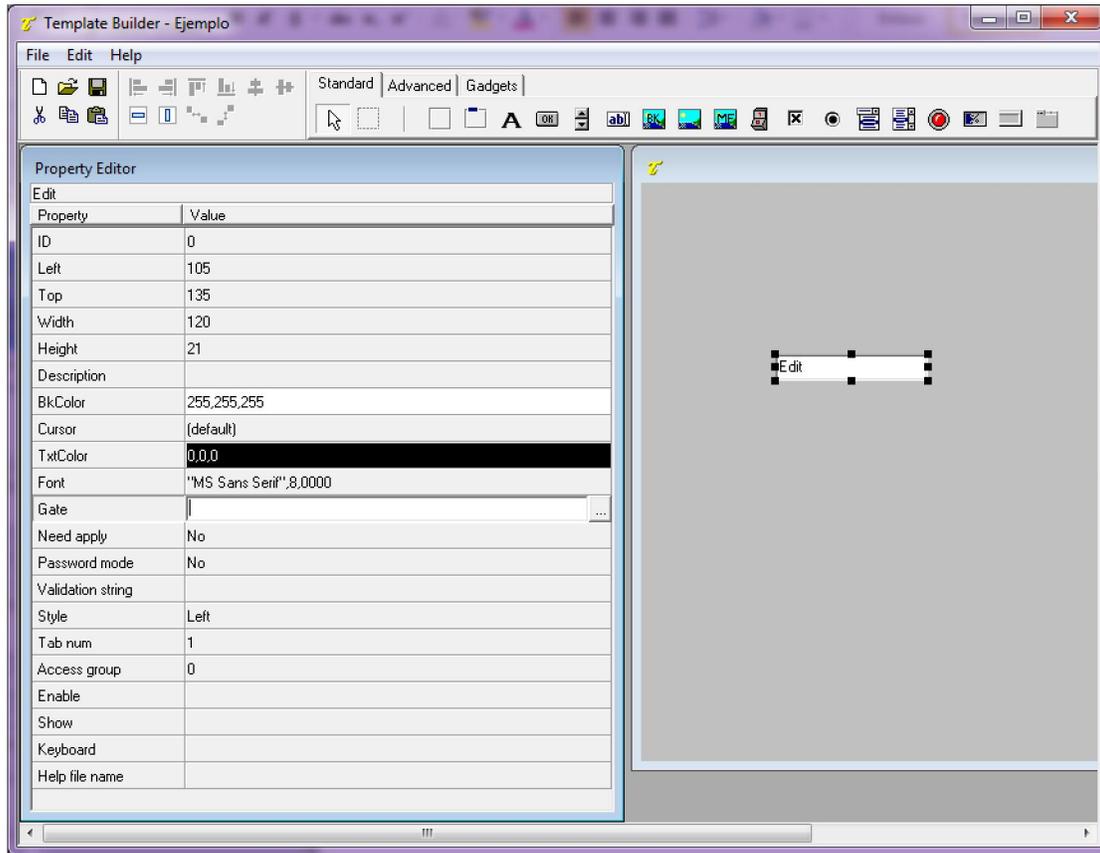


### 3.5.4. CUADRO DE TEXTO (EDIT)

Se pueden utilizar tanto para representar información leída desde una entrada analógica como para enviar un valor hacia el PLC (por ejemplo para fijar el tiempo de funcionamiento de un equipo).

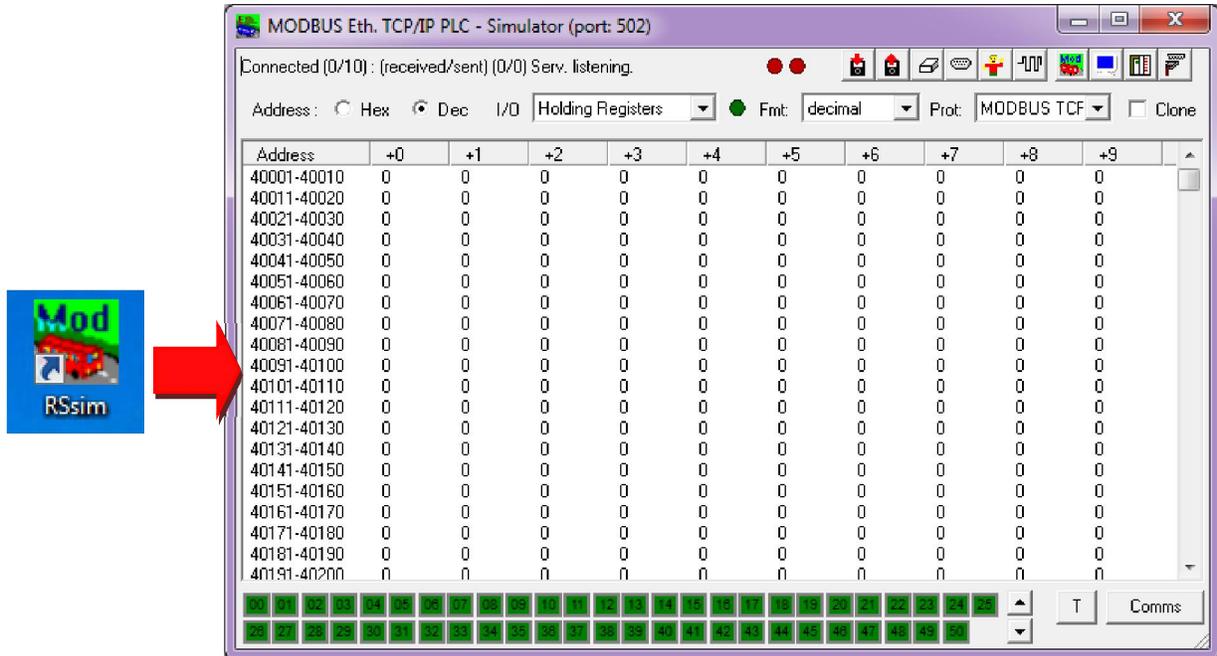
Se encuentra dentro de la lengüeta General. Para insertarlo se hace clic en Edit  y luego en el template.

Para asociarlo con un TAG analógico se hace clic en Gate y se elige el TAG, por ejemplo *Temperatura 2*



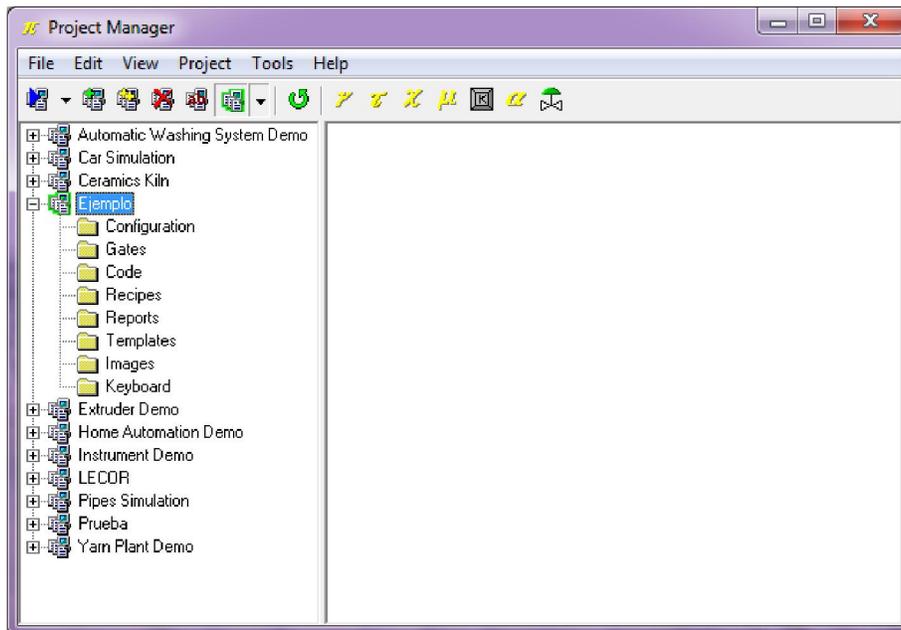
#### 4. EJECUCIÓN DEL PROGRAMA

El primer paso consiste en ejecutar el simulador, haciendo doble clic sobre el ícono del mismo se abrirá la ventana del simulador.



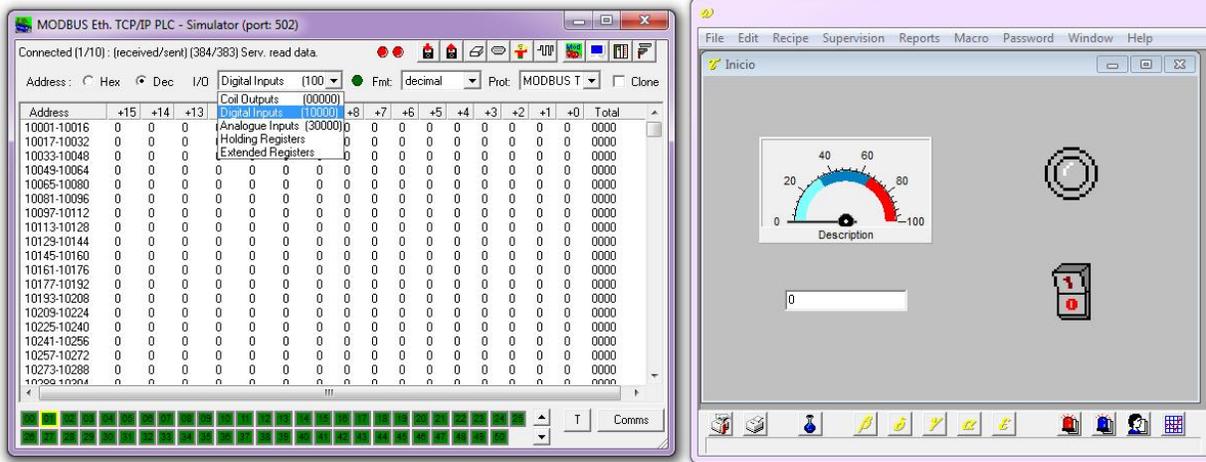
Confirmar que Prot se encuentre en MODBUS TCP/IP

Ejecutar el ejemplo en Winlog, para esto seleccionarlo y hacer clic en el botón Ejecutar .

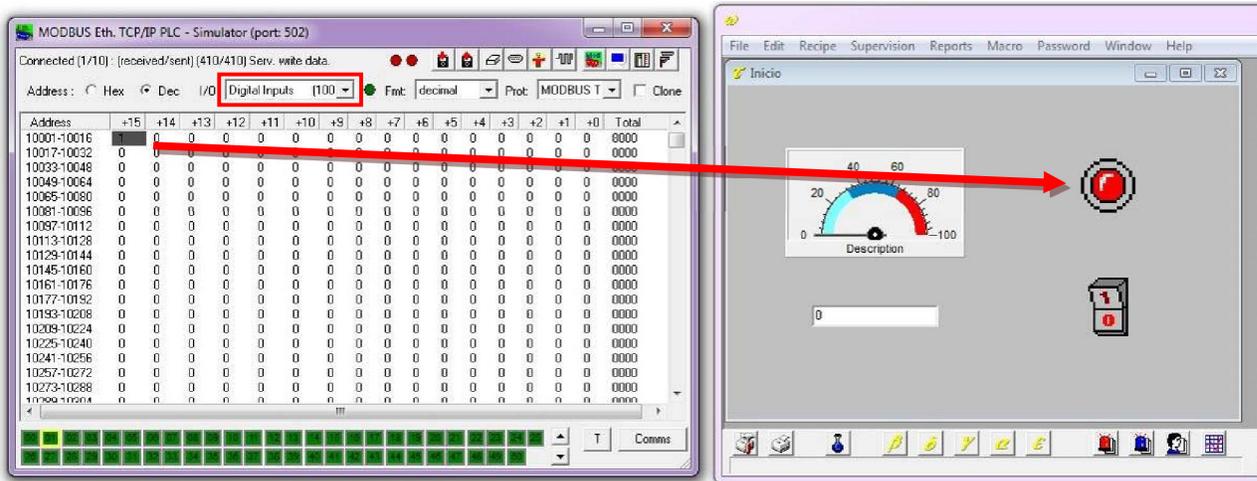


Se va a abrir el Template Inicio.

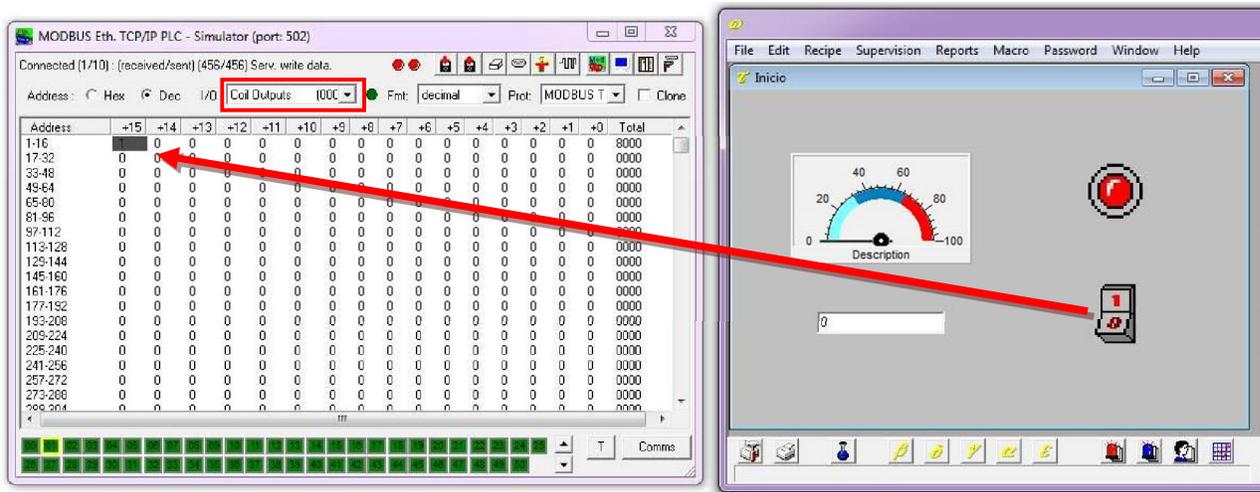
Elegimos **Digital Inputs** en el simulador:



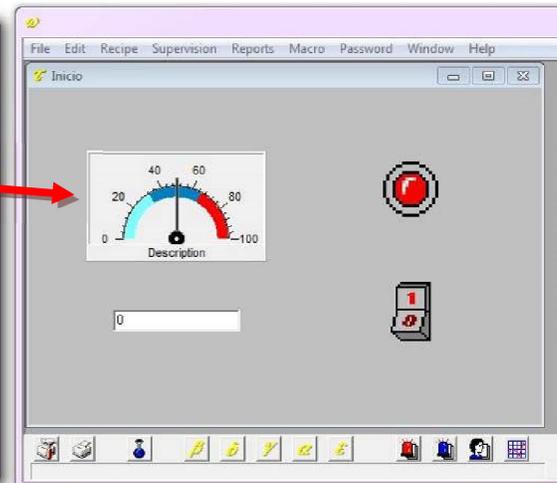
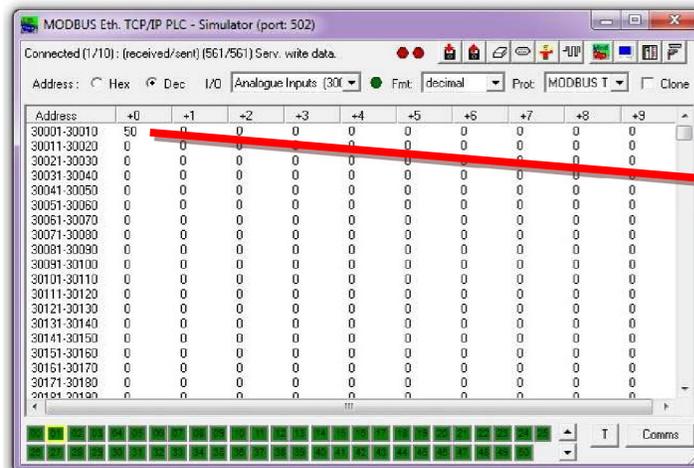
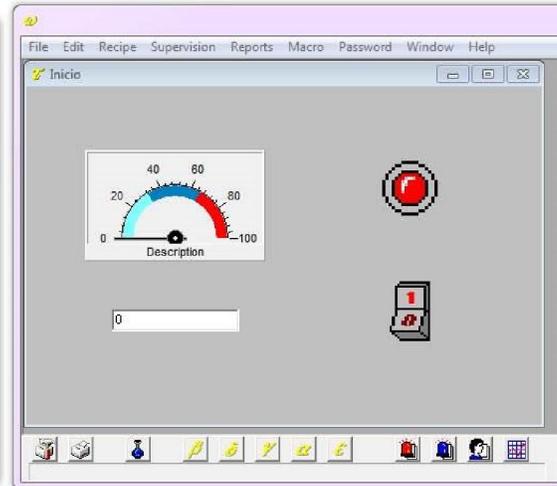
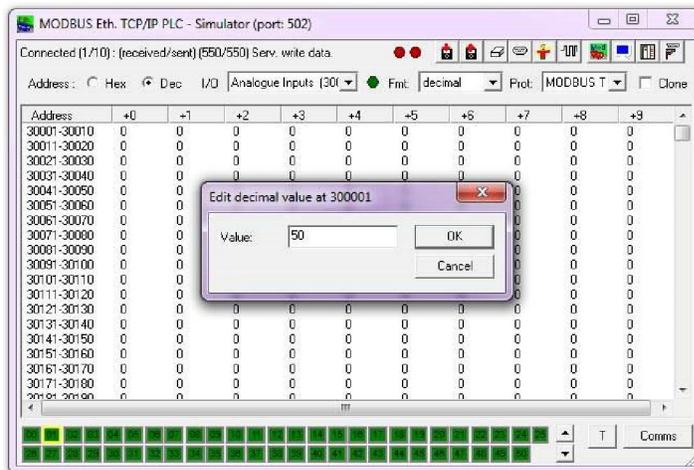
Para activar una entrada digital se hace doble clic sobre la misma, en nuestro caso activamos la primera entrada digital (+15) que está asociada con *Digital 1* y a su vez con el LED.



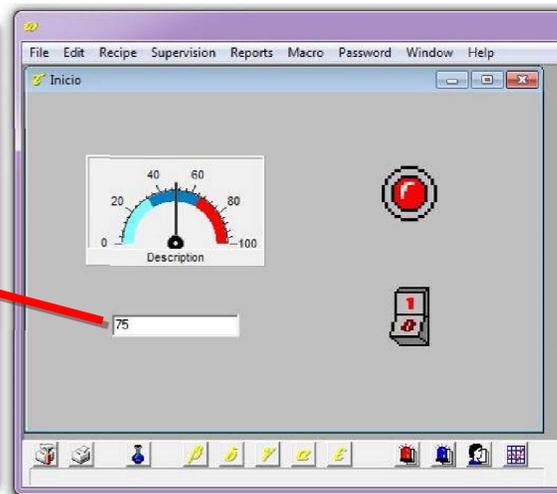
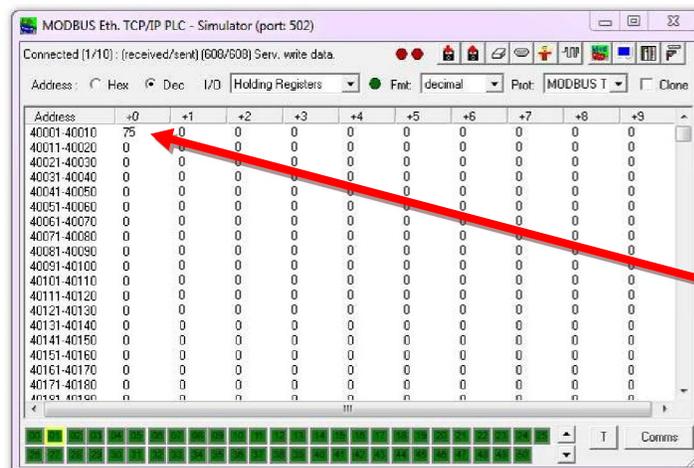
Podemos probar el botón que está asociado con el Coil Output 1 (+15)



La entrada analógica 1 (*Analogue Input +0*) está asociada con el Vúmetro, para cambiar su valor se hace doble clic sobre la posición y en la ventana que aparece se carga el valor.

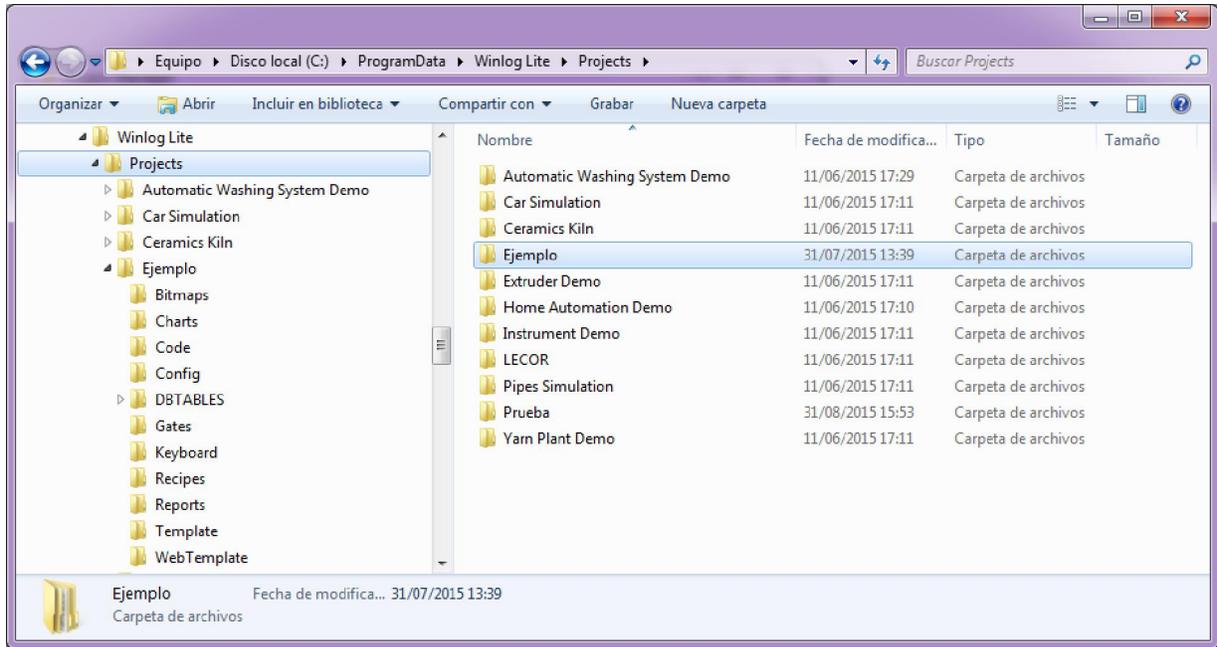


Finalmente podemos enviar un valor hacia el *Holding Register 0* utilizando el cuadro de texto.

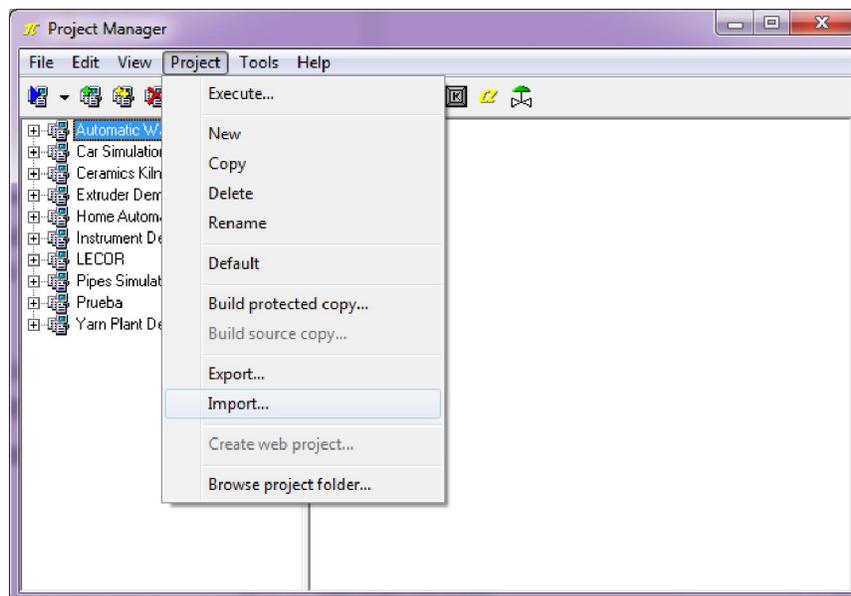


## 5. IMPORTAR UN PROYECTO

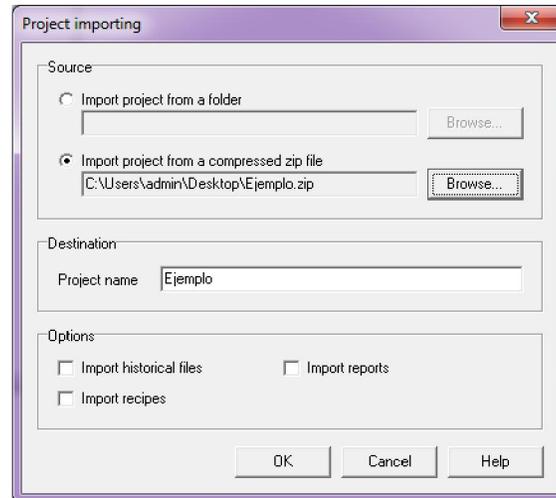
Los proyectos creados en Winlog se almacenan como carpetas en el directorio C:\Archivos de Programa\Winlog Lite\Projects.



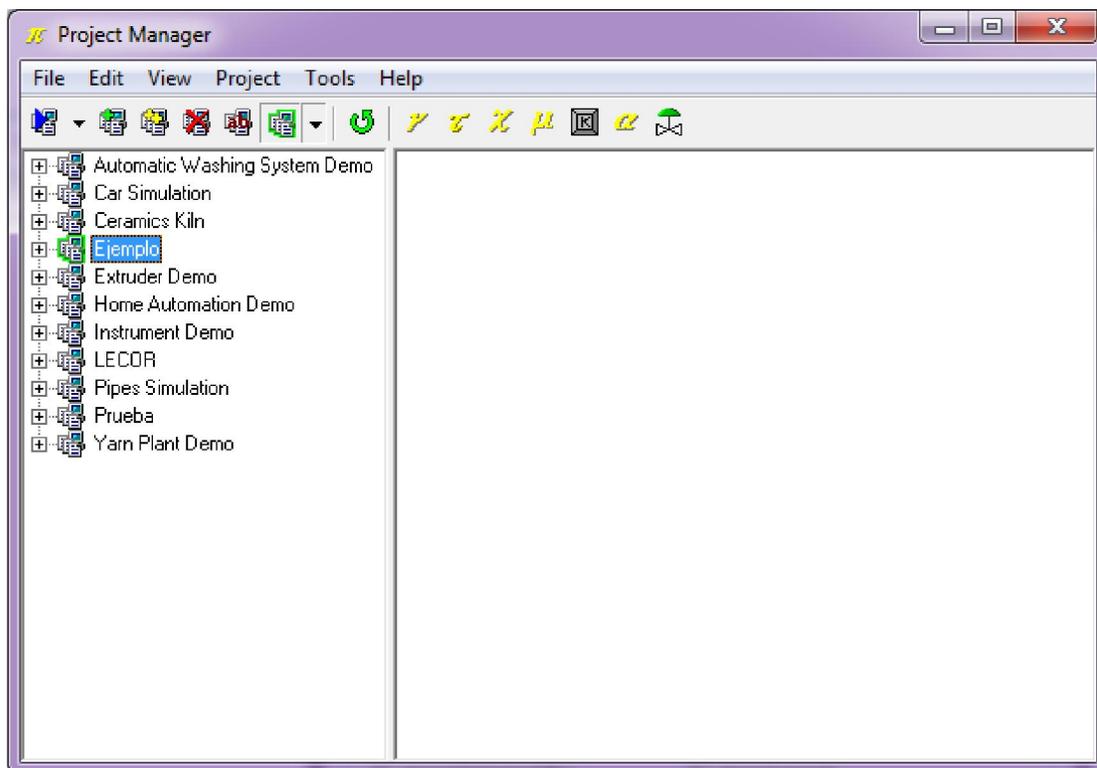
Es posible exportar e importar los mismos copiando directamente estos directorios, o a través de la opción de importar o exportar.



Seleccionando Import aparece la ventana de importación, aquí *seleccionamos Import project from a compressed zip file* y hacemos clic en **Browse**.



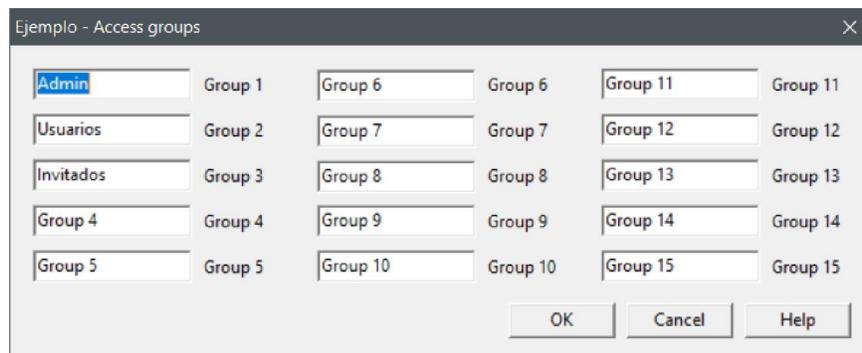
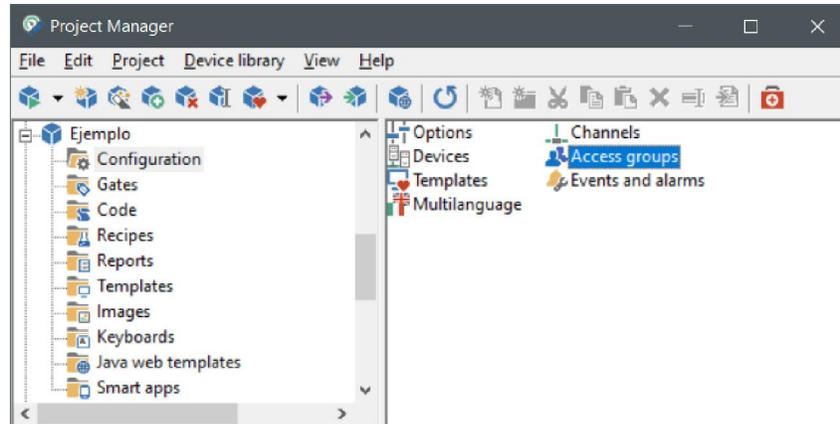
Buscamos el archivo comprimido que contiene el proyecto y hacemos clic en Ok. Winlog importará el proyecto y lo agrega a la lista de proyectos disponibles.



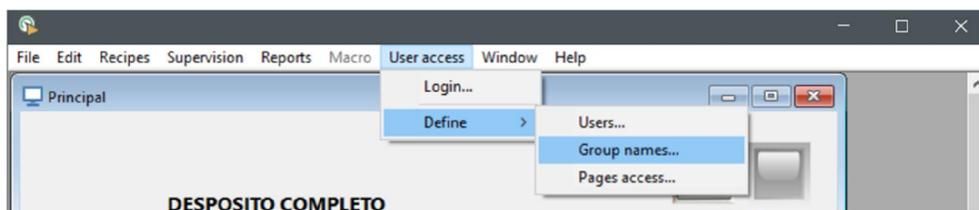
## 6. USUARIOS Y GRUPOS

En Winlog es posible definir usuarios y Grupos a los que pertenecen estos usuarios, esto tiene por objeto permitir acceso a los elementos en base al grupo al que pertenecen los usuarios, por ejemplo se puede crear en un template un botón de arranque que solo pueda ser activado por usuarios del grupo Administrador, de esta forma solo los usuarios que pertenecen al grupo administrador pueden dar arranque a un proceso.

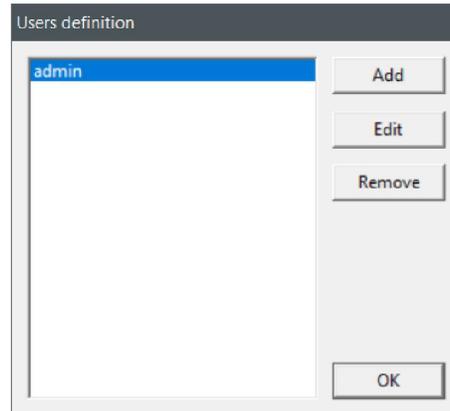
Por defecto Winlog posee 15 Grupos, es posible cambiar el nombre de los mismos desde Configuración → Access Groups



La creación de usuarios, y su asignación a los grupos definidos se realiza desde la ventana de ejecución de Winlog (aun cuando no exista comunicación con la maqueta) desde User Access, en este menú se puede realizar el Login con un usuario existente, definir usuarios nuevos, cambiar nombres de grupos y asignar a que páginas tiene acceso cada usuario.

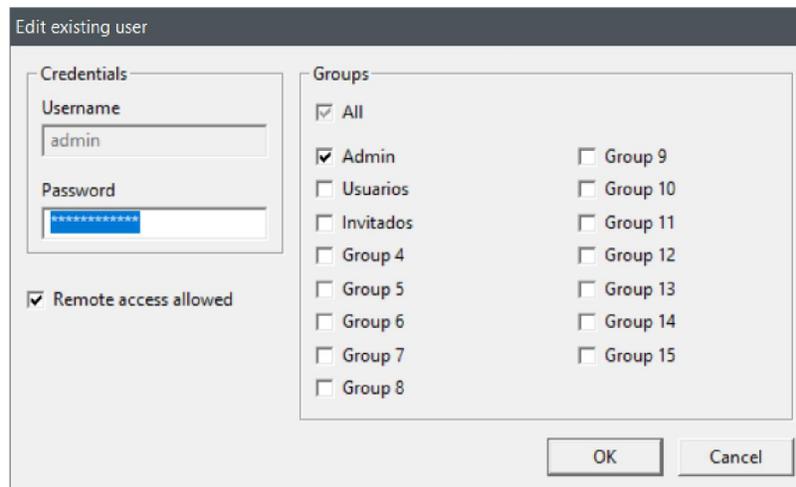


En la ventana que se abre al seleccionar usuarios es posible agregar, modificar o borrar usuario:

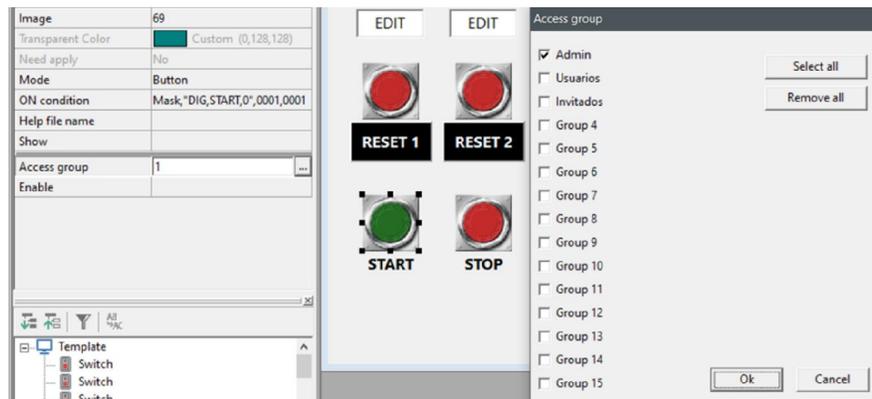


Al agregar o modificar un usuario se puede indicar a que grupos pertenece, la clave de acceso y si tiene permitido acceso remoto o no.

En este ejemplo el usuario admin pertenece al grupo Admin (Administradores) y tiene permitido el acceso remoto)

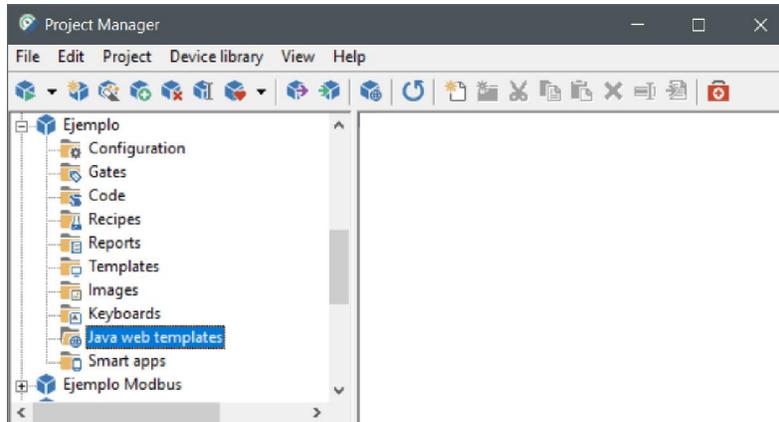


Es posible entonces asignar a los elementos de un template un grupo al que pertenecen, por ejemplo asignar el botón Start al grupo Admin, de manera que solo los usuarios asignados a este grupo puedan interactuar con él.

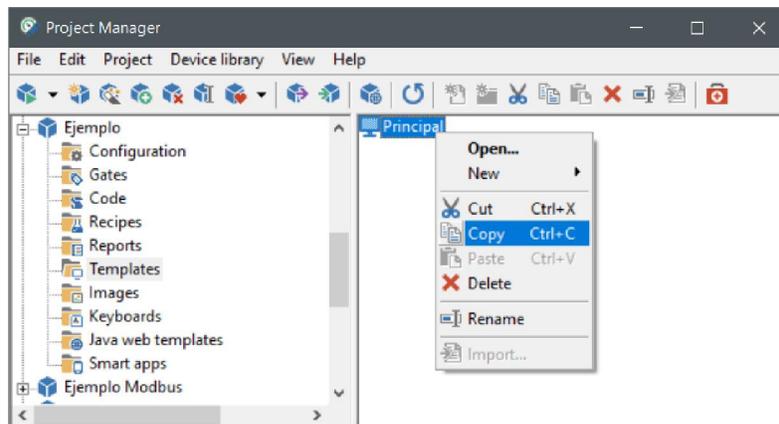


## 7. JAVA WEB TEMPLATES

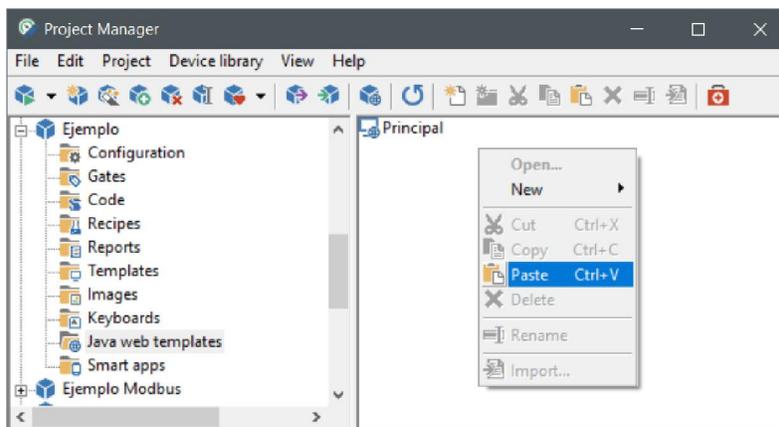
Permite armar ventanas de supervisión que pueden ser accedidas desde una PC en forma remota a través de un navegador y un cliente Java. Requiere tener Java instalado en la PC cliente.



Es posible utilizar el mismo template diseñado en Templates (Punto 3.5) para esto se selecciona el template, con el botón derecho se elige Copiar

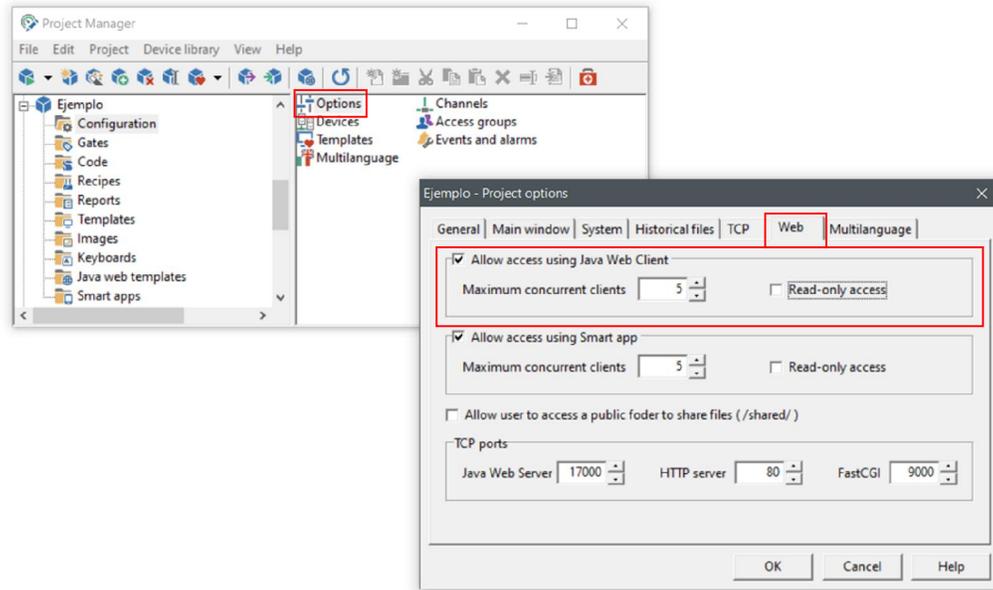


En Java web templates con el botón derecho se selecciona Pegar:



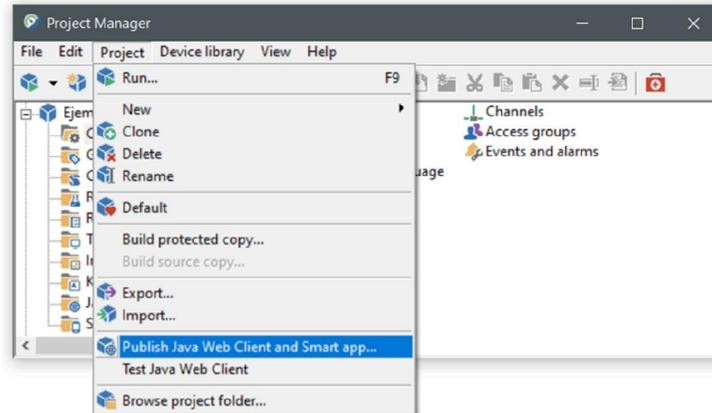
El template puede ser editado del mismo modo que un template convencional.

Con el template creado se debe activar el acceso usando el cliente Java desde opciones → Web → Allow Access using Java Web Client



Por defecto el acceso es de solo Lectura, si es necesario modificar valores en forma remota se debe destildar Read-only Access, también es posible indicar el número máximo de usuarios que pueden estar conectados al mismo tiempo.

Finalmente es necesario Publicar el Cliente, esto se consigue llenando a Project → Publish Java Web Client

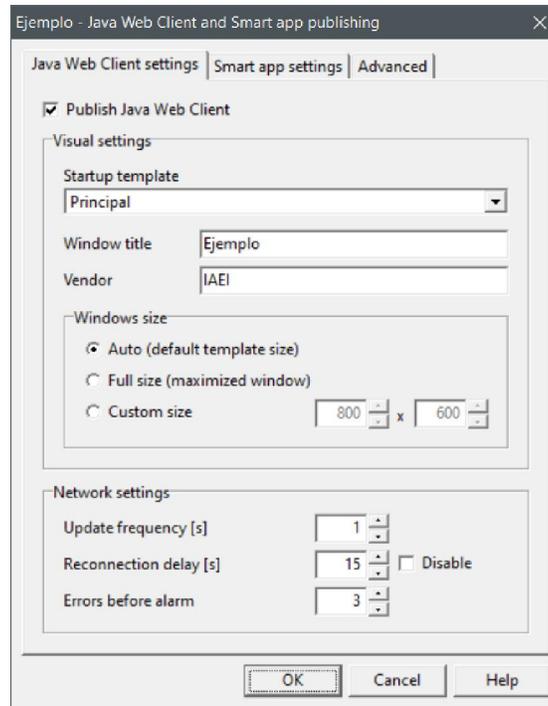


En la ventana que se abre seleccionar la lengüeta Java Web Client settings y tildar Publish Java Web Client.

En caso de que exista mas de un template se puede elegir el template que se muestra por defecto (Startup Template)

Window Title (el título de la ventana) y Vendor (Nombre del creador del template) pueden ser indicados libremente por el usuario.

Finalmente se puede indicar la frecuencia de actualización de valores, por defecto son 5 segundos, mientras mas bajo es este valor mas frecuentemente se actualizarán los datos mostrados. En el ejemplo se ha elegido un tiempo de actualización de 1 segundo.



En estas condiciones ya se puede ejecutar el proyecto (Punto 4 de esta guía)

Utilizando un navegador web acceder a la dirección de red de la PC donde está instalado Winlog, en este ejemplo se utiliza un navegador en la misma PC donde se ejecuta Winlog, por lo que la dirección de red utilizada es la dirección local (dirección de loopback 127.0.0.1)

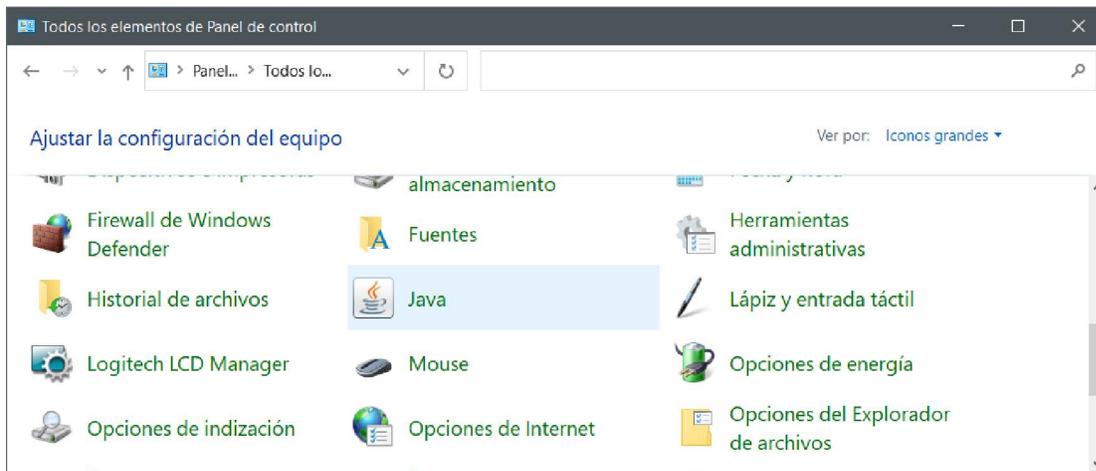


Si el servidor Web Java se está ejecutando correctamente debería estar habilitada la opción Java Web Client.

Al hacer click sobre ella el navegador va a mostrar instrucciones sobre como habilitar Java para el client, esto es debido a que la seguridad de Java no permite la activación del cliente por defecto, por lo que es necesario agregar la dirección del servidor de winlog a la lista de direcciones permitidas.

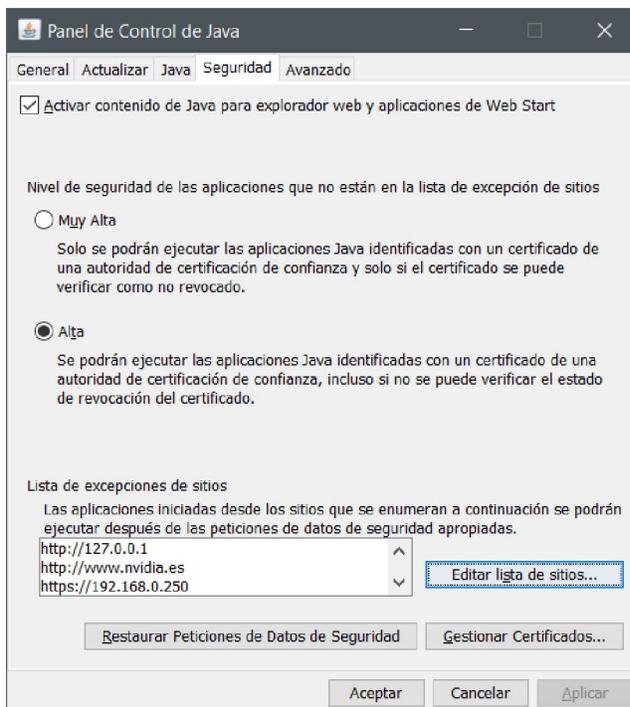
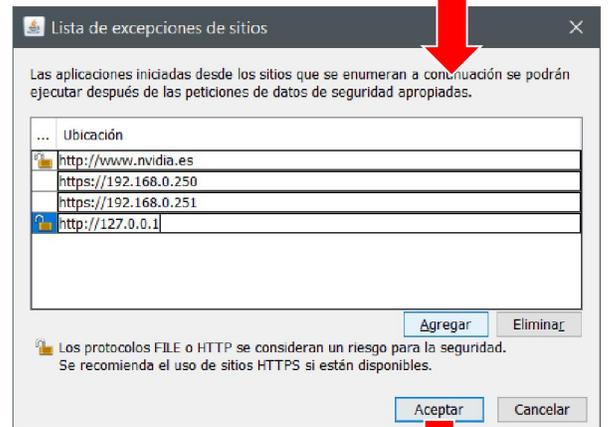
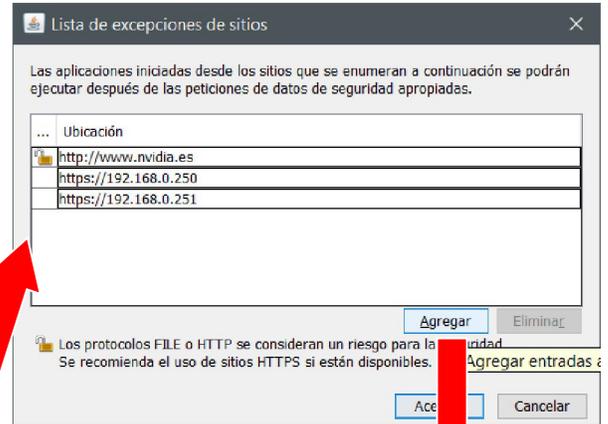
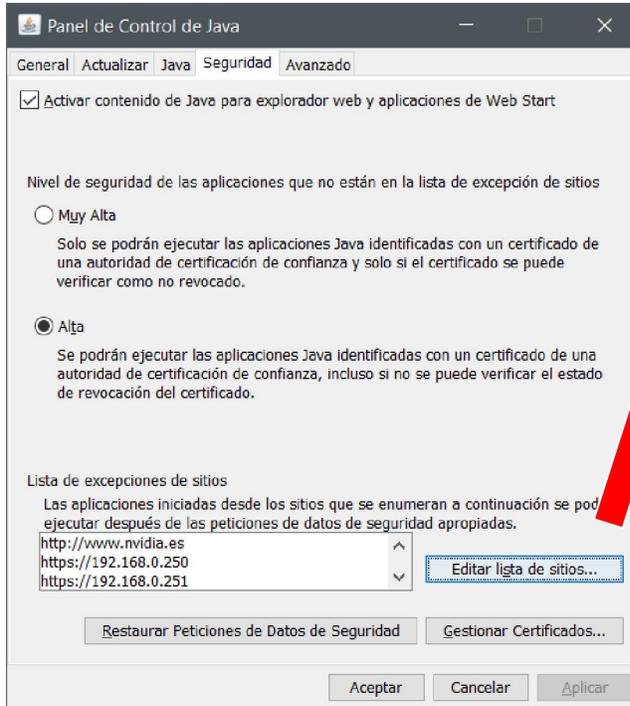


Si Java está instalado en el equipo, en el panel de control de Windows debería figurar un ícono llamado Java (esta operación se realiza por

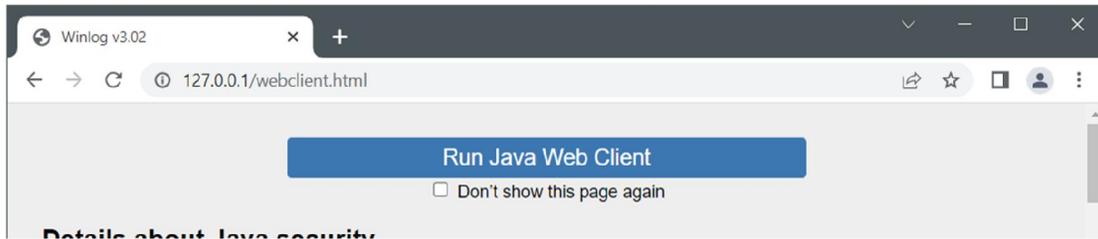


Haciendo clic sobre el mismo se abre la ventana de configuración, en la lengüeta Security se encuentra el listado de direcciones permitidas.

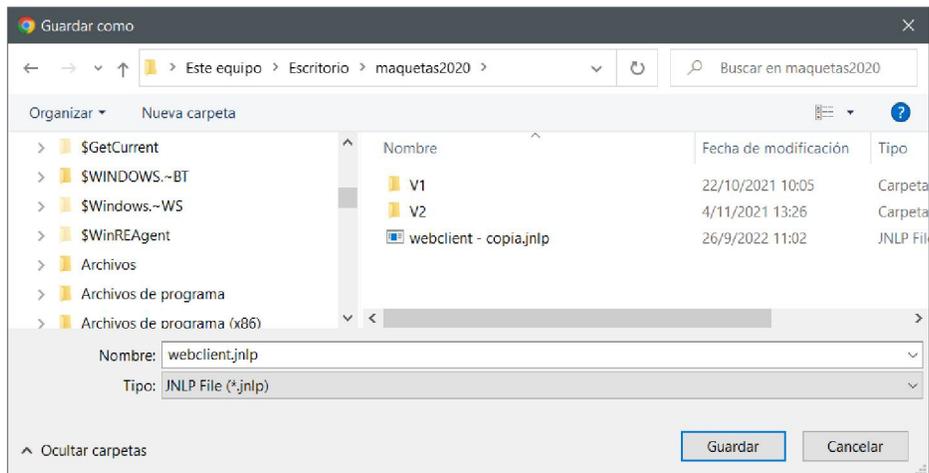
Usando el botón Editar lista de sitios se agrega la dirección de red de la PC que está ejecutando Winlog, como mencionamos anteriormente en este ejemplo se ejecuta en la misma PC por lo que la dirección que se debe agregar es: <http://127.0.0.1>



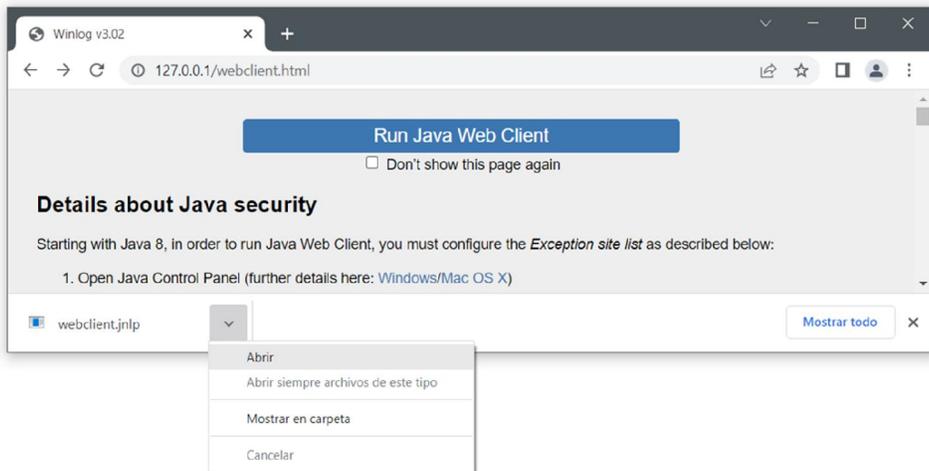
Finalmente se puede ejecutar el cliente Java:



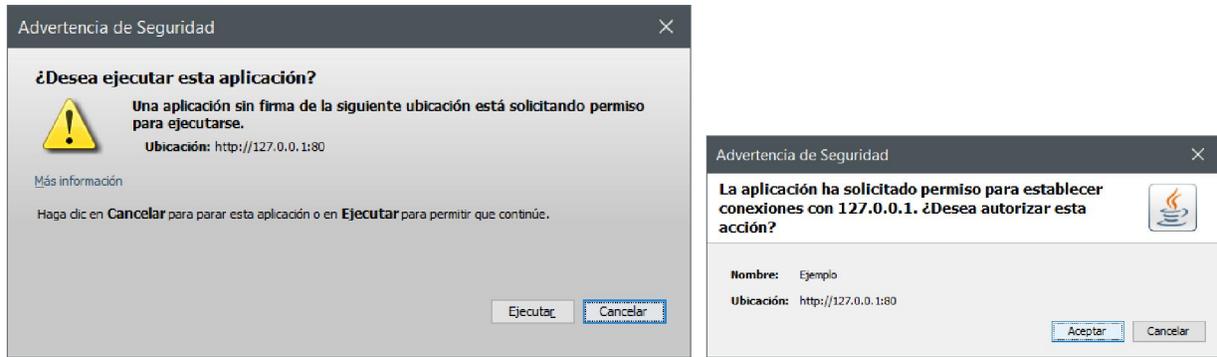
Va a solicitar un directorio temporal de destino para descargar el cliente java:



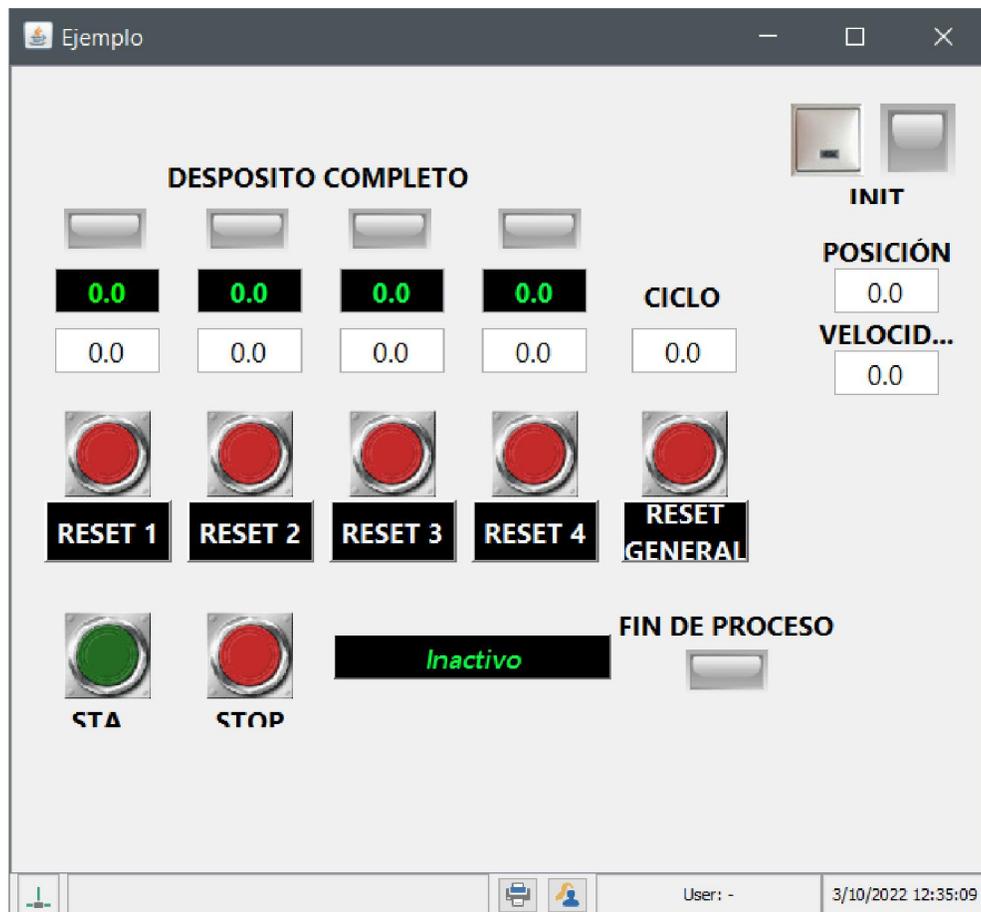
Una vez descargado se ejecuta:



Ejecutar esta aplicación y permitir las conexiones de red.



Finalmente tendremos acceso remoto al SCADA mediante el cliente Java.



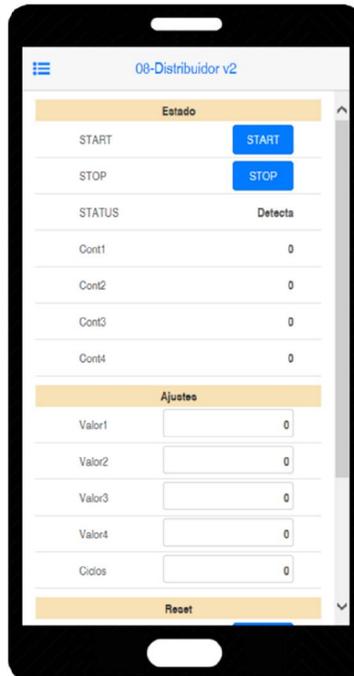
## 8. SMART APPS

SmartApps permite diseñar ventanas de supervisión para acceder en forma remota desde cualquier dispositivo móvil, por ejemplo, un celular o una Tablet.

Las ventanas generadas son más simples que las generadas en Java Web Client, pero tienen la ventaja que no requieren Java, lo que las hace compatible con cualquier tipo de dispositivo que disponga de un navegador web.

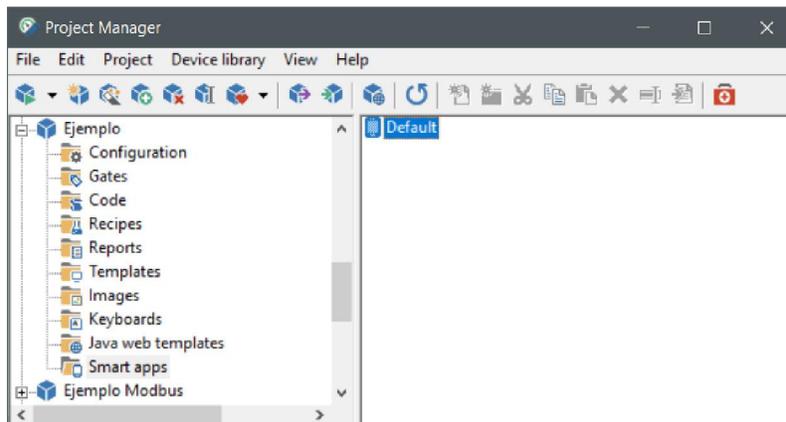
En este caso no se puede reutilizar un template existente como en Java Web Client.

En este ejemplo pueden verse botones para inicio y parada, indicadores de estado del proceso y estado de contadores, con la posibilidad de modificar las consignas de los mismos.

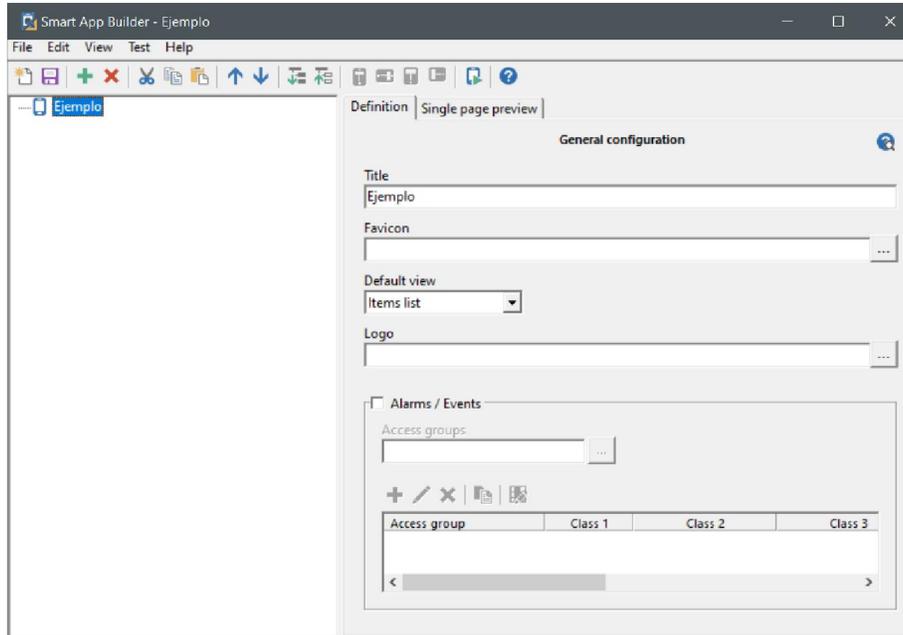


### 8.1. CREACIÓN DE UNA SMARTAPP

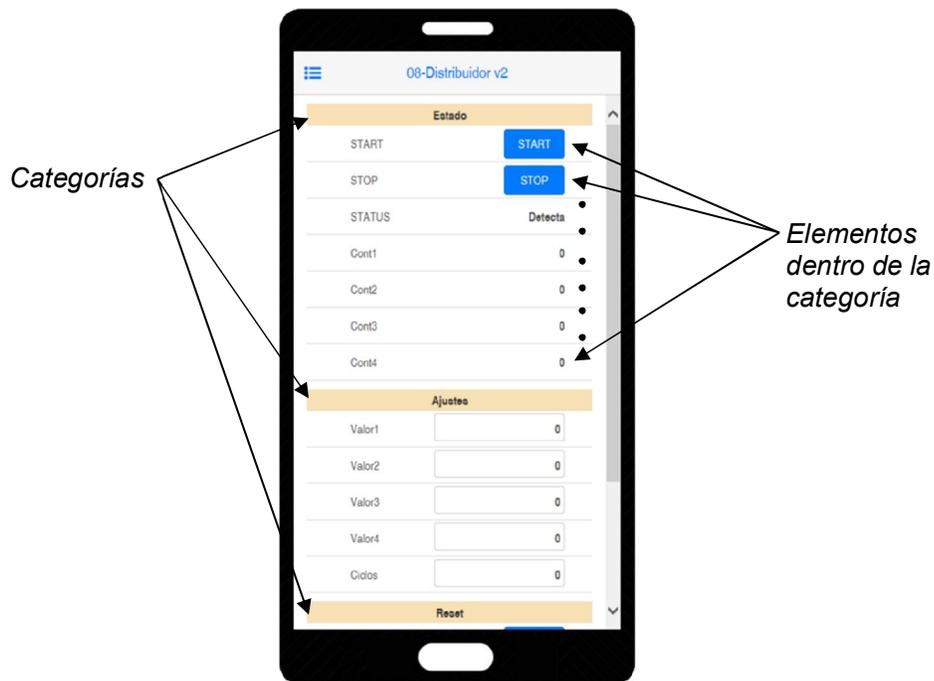
Vamos a la categoría SmartApp y hacemos doble clic en Default.



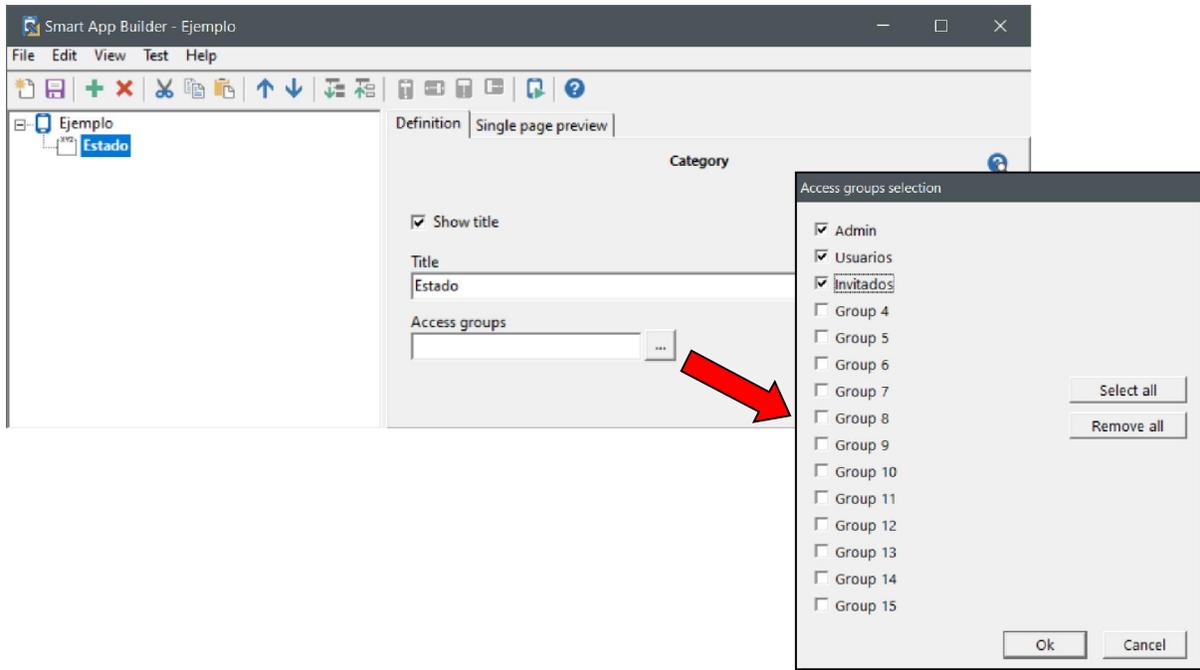
En la ventana que se abre vemos la configuración general, en ella podemos indicar el nombre del proyecto (en este caso Ejemplo) asociarle un ícono (es optativo), la vista por defecto (ver el listado de elementos de control, como en la imagen de ejemplo más arriba, o ver solamente el logo de la aplicación), podemos seleccionar una imagen como Logo de la aplicación, y definir la visualización de alarmas y eventos.



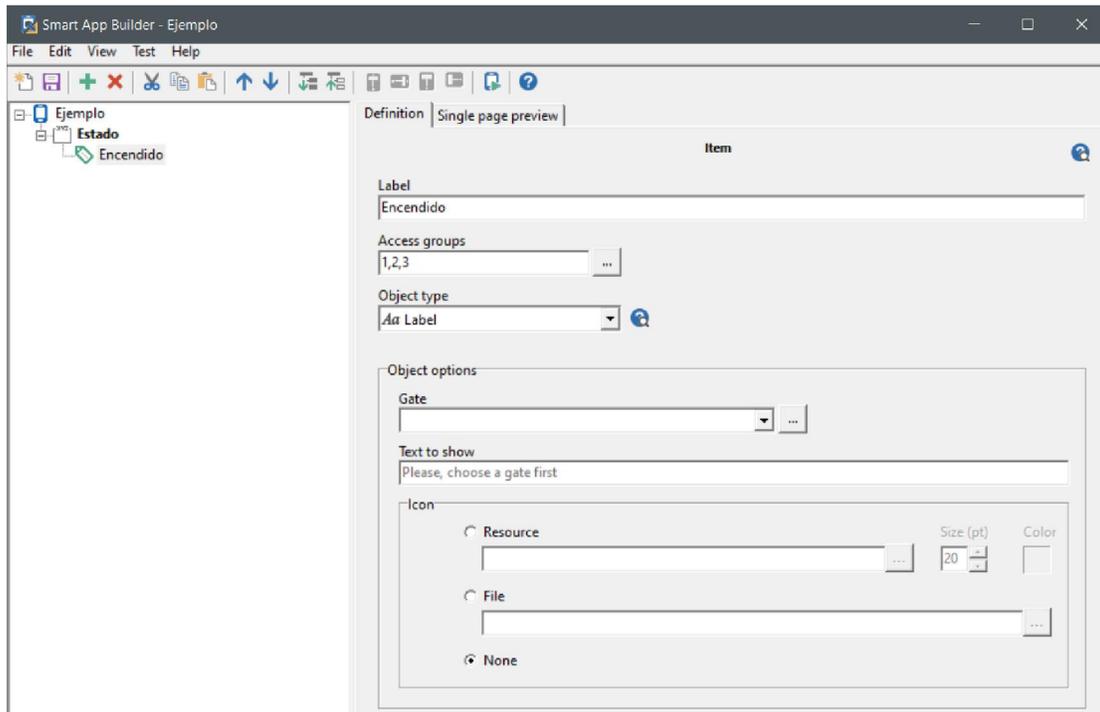
Haciendo clic en el signo + agregamos una categoría dentro de la cual podemos crear elementos para visualizar. Podemos tener varias categorías que agrupen diferentes elementos.



En la ventana de categoría indicamos el nombre de la categoría y que grupos tienen acceso a la misma (ver punto 6). Los grupos se seleccionan haciendo clic en el botón y en la ventana que aparece se tildan los grupos deseados.



Con la categoría seleccionada se puede hacer clic nuevamente sobre el signo + para comenzar a agregar elementos.



Debemos asignarle un nombre al elemento (Label), indicar los grupos que tienen acceso al mismo (igual que como hicimos anteriormente)

Finalmente debemos indicar el tipo de objeto que queremos utilizar, por ejemplo si queremos un botón, un mensaje de texto, ver el estado de un contador, etc.

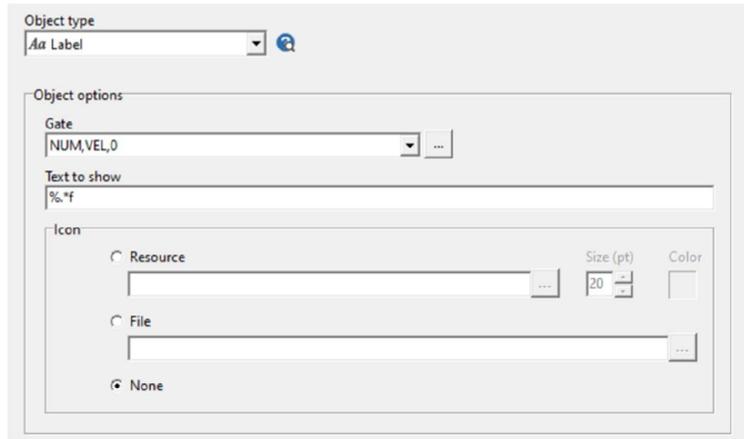
Tenemos dos tipos de objetos:

## 8.2. ELEMENTOS DE SOLO LECTURA

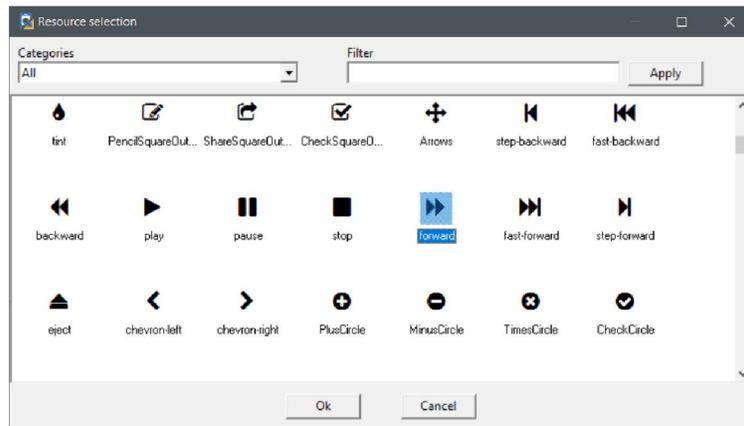
Nos permiten visualizar un estado pero no modificarlo, dentro de esta categoría tenemos:

### 8.2.1. LABELS:

Nos permiten visualizar el valor almacenado en un Tag, por ejemplo el valor de una velocidad de desplazamiento.



Icon es optativo y es más bien estético, para indicar al usuario gráficamente el tipo de valor a visualizar. Se puede agregar iconos de una base incluida con Winglog (Resource)



O desde una imagen creada por el usuario (File).

### 8.2.2. N-STATES LABEL:

Se utilizan para asociar distintos estados para cada valor de un Tag. Por ejemplo podemos tener un Tag llamado Estado que representa distintos estados del sistema para cada valor que toma, por ejemplo si tiene el valor 1 significa que el sistema está Inactivo, si tiene valor 2 se encuentra detectando, etc.

Igual que en el caso anterior debemos indicar un nombre para el elemento (Label), definir que grupos tienen acceso, y elegir como tipo de objeto N-States Label.

Dentro de opciones del objeto seleccionamos el Tag que tiene el valor a utilizar, en este ejemplo se llama Estado, y asociar un texto a mostrar para cada valor que vaya tomando.

Label  
STATUS

Access groups  
1,2,3

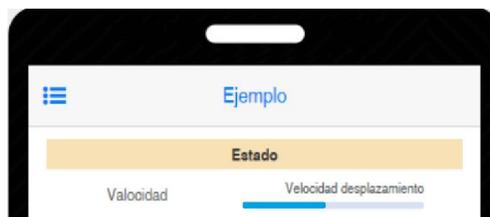
Object type  
N-States label

Object options  
Gate  
NUM,ESTADO,0

| Gate value | Text to show | Icon | Resource / Icon file name | Size (pt) |
|------------|--------------|------|---------------------------|-----------|
| 0          | Inactivo     |      |                           |           |
| 1          | Detecta      |      |                           |           |
| 2          | Busca        |      |                           |           |
| 3          | Evalua       |      |                           |           |
| 4          | Traslada     |      |                           |           |
| 5          | Deposita     |      |                           |           |
| 6          | Finalizado   |      |                           |           |
| 7          | Espera       |      |                           |           |

### 8.2.3. GAUGE:

Nos permite ver el valor numérico almacenado en un Tag, en forma gráfica.



Nuevamente indicamos un nombre (Label), definimos los grupos que pueden acceder, y como tipo de objeto elegimos Gauge.

Label  
Valoocidad

Access groups  
1,2,3

Object type  
Gauge

Object options  
Gate  
NUM,N,1

Text to show  
Velocidad desplazamiento

Width: 150 | Measure unit: (px) pixels

Minimum value: 0 | Maximum value: 40

Icon  
 Resource  
 File  
 None

En Opciones de objeto Seleccionamos el Tag que vamos a leer, en este caso debe ser un Tag de tipo Numérico, no digital.

Podemos elegir un texto para mostrar por encima (es opcional), el ancho de la línea, y los valores mínimo y máximo que va a tomar el indicador.

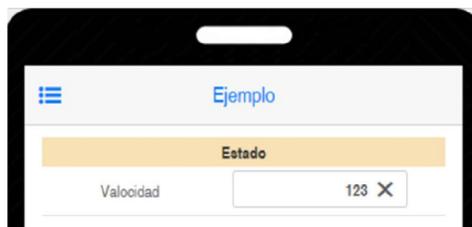
Como en el caso de los Labels es posible asignar un ícono, solo a modo descriptivo.

### 8.3. ELEMENTOS DE LECTURA/ESCRITURA

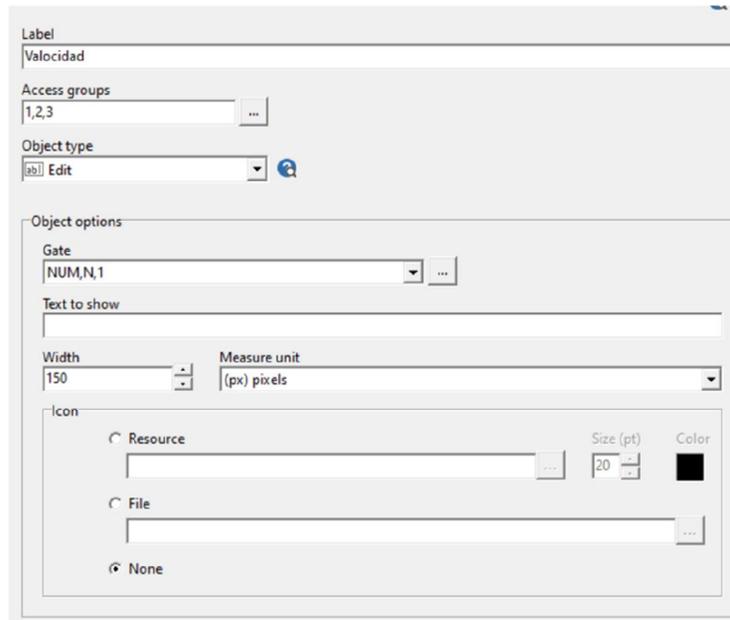
Permiten modificar el estado del proceso, dentro de esta categoría tenemos:

#### 8.3.1. EDIT

Similar a Label, pero permite escribir valores que serán enviados al Tag asociado con el elemento



Asignamos un nombre (Label), asociamos los grupos de acceso, en tipo de objeto elegimos Edit

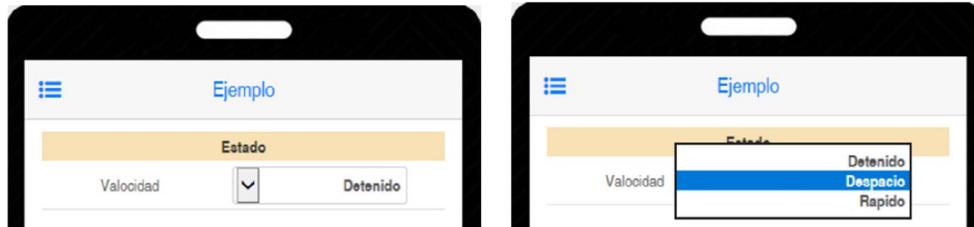


En opciones de objeto asociamos el Tag (debe ser de lectura/escritura), podemos asignar un texto para mostrar (optativo) el ancho del casillero y un ícono.

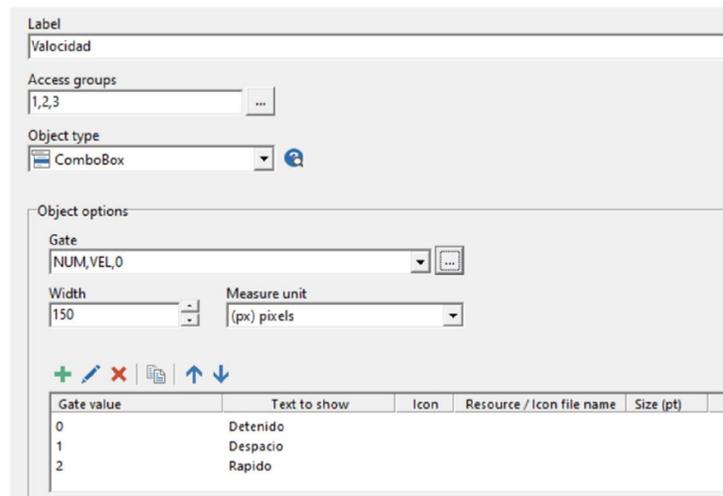
### 8.3.2. COMBO BOX

Permite asignar un valor a un Tag mediante un cuadro de texto desplegable, por ejemplo podemos tener un Tag llamado Velocidad que cuando toma el valor 0 detiene el movimiento de un carro, con valor 1 se mueve lentamente, y con valor 2 se mueve rápidamente.

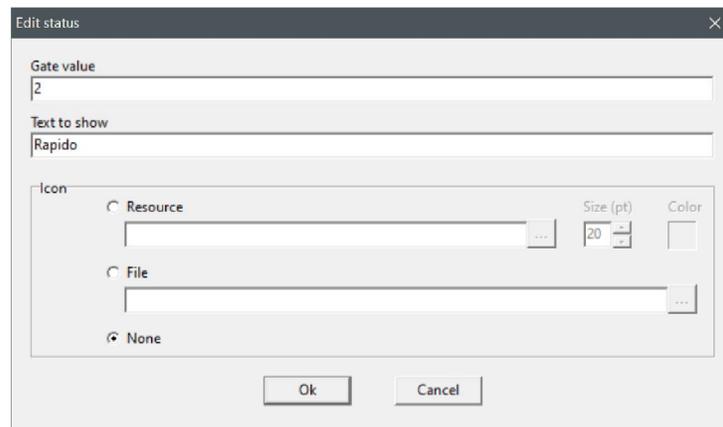
Combo Box permite asociar valores con un texto descriptivo, de manera que si el usuario selecciona Despacio se envía el valor 1 al Tag Velocidad, si selecciona Rápido se envía el valor 2, etc.



Nuevamente debemos indicar un nombre (Label), definir los grupos de acceso y seleccionar como tipo de objeto ComboBox.



Asociamos el Tag correspondiente, en este ejemplo VEL (recordar que el mismo tiene que haber sido creado como Lectura/Escritura), podemos elegir el ancho del recuadro, y finalmente utilizando el signo + asociamos distintos valores a líneas de texto.

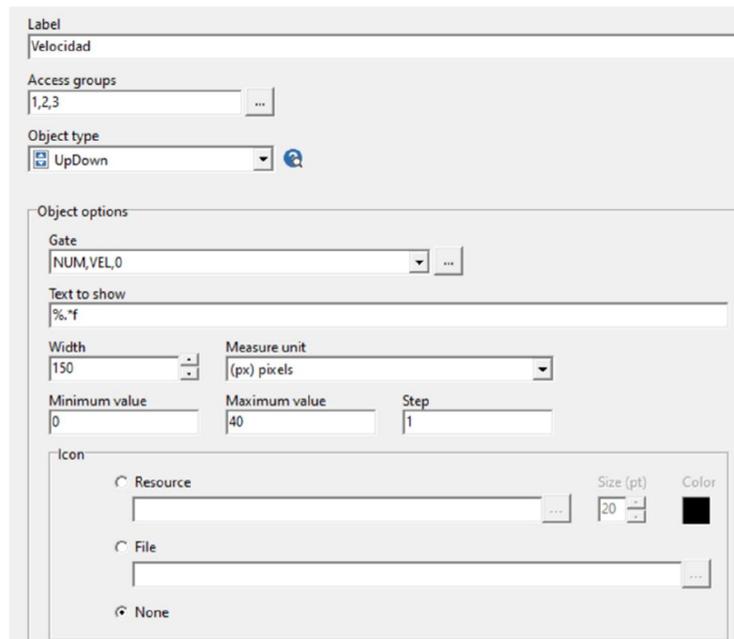


### 8.3.3. UPDOWN

Permite incrementar o decrementar el valor de un Tag empleando flechas arriba/abajo.



Debemos asignarle un nombre (Label), definir los grupos de acceso y seleccionar como tipo de objeto UpDown.



The configuration window for an UpDown object includes the following fields:

- Label:** Velocidad
- Access groups:** 1,2,3
- Object type:** UpDown
- Object options:**
  - Gate:** NUM,VEL,0
  - Text to show:** %.\*f
  - Width:** 150
  - Measure unit:** (px) pixels
  - Minimum value:** 0
  - Maximum value:** 40
  - Step:** 1
- Icon:**
  - Resource
  - File
  - None

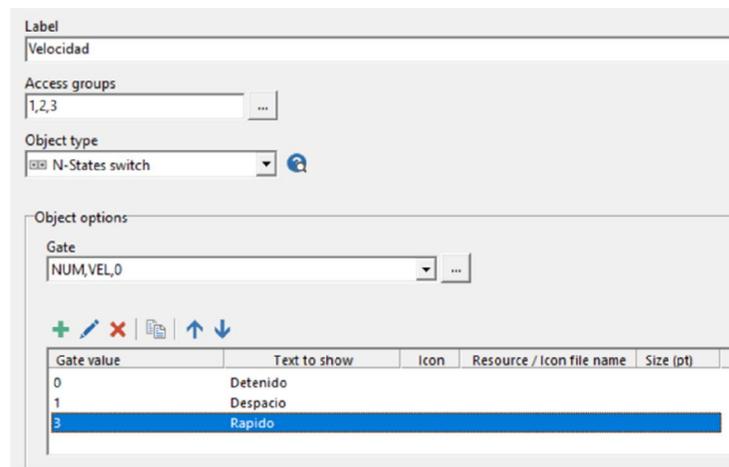
En opciones de objeto seleccionamos el Tag (debe ser de lectura/escritura). Podemos asignar un texto a mostrar (por defecto muestra el valor del tag), indicar el ancho del cuadro, y finalmente podemos indicar el valor mínimo, máximo y de a cuanto se incrementa/decrementa el valor, en este ejemplo el mínimo es 0, el máximo es 40 y con cada clic el valor sube o baja en 1.

### 8.3.4. N-STATES

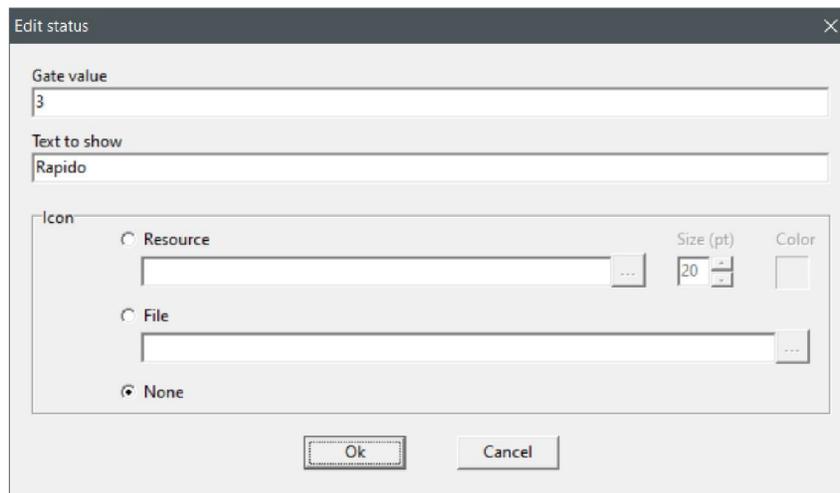
Permite seleccionar rápidamente entre distintos estados, cada estado envía un valor hacia el Tag asociado (similar a 8.3.2 ComboBox)



Se le asigna un nombre (Label), se seleccionan los grupos de acceso y como tipo de objeto seleccionar N-States.

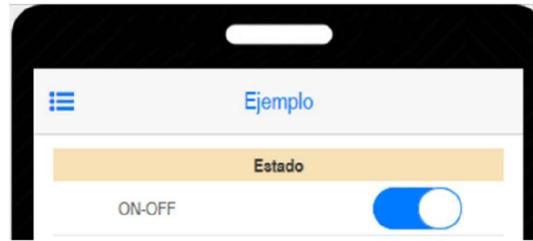


En opciones de objeto seleccionamos el Tag a asociar (debe ser Lectura/Escritura) y utilizando el signo + agregamos distintos estados asignando un valor y un texto a cada uno.

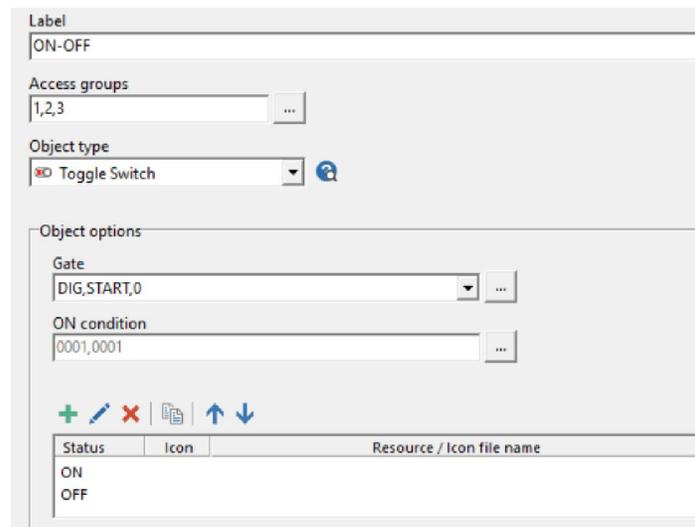


### 8.3.5. TOGGLE SWITCH

Es switch de dos estados, para trabajar con Tags de tipo Digital.



Le asignamos un nombre (Label), seleccionamos los grupos de acceso y como tipo de objeto elegimos Toggle Switch.



En opciones de objeto seleccionamos el Tag, debe ser de tipo Digital, y Lectura/escritura.

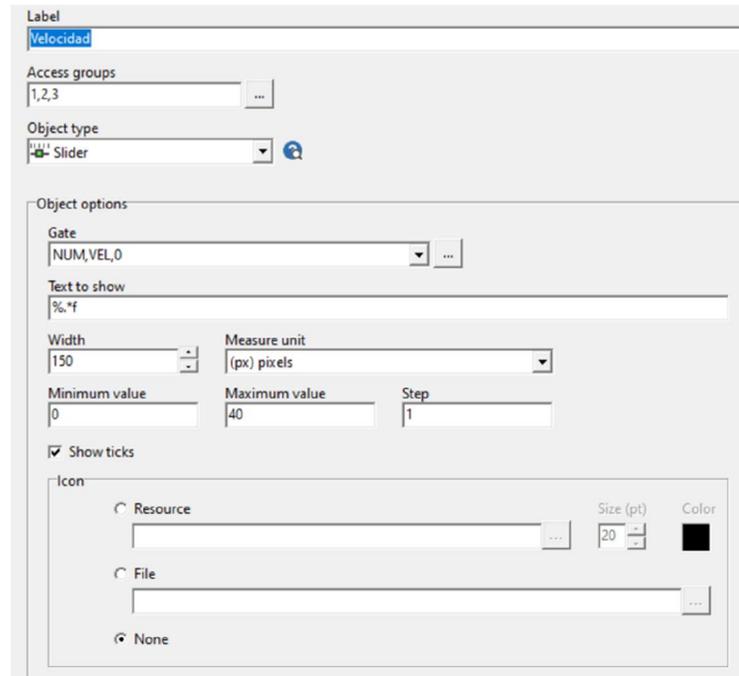
La condicon de encendido se pude modificar (por defecto On envía al Tag el valor 1) y se puede asociar un ícono para mostrar al usuario cuando el siwtch está en estado On o Off.

### 8.3.6. SLIDER

Permite asignar un valor a un Tag empleando un deslizador.



Especificamos un nombre (Label), seleccionamos los grupos de acceso y como tipo de objeto elegimos Toggle Switch.

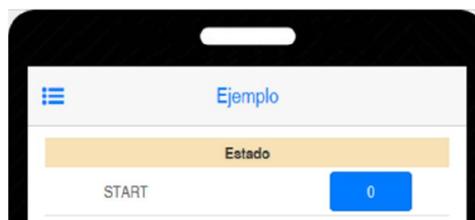


En opciones de objeto seleccionamos el Tag a asociar (debe ser Numérico, Lectura/Escritura), Por defecto el texto a mostrar es el valor del Tag.

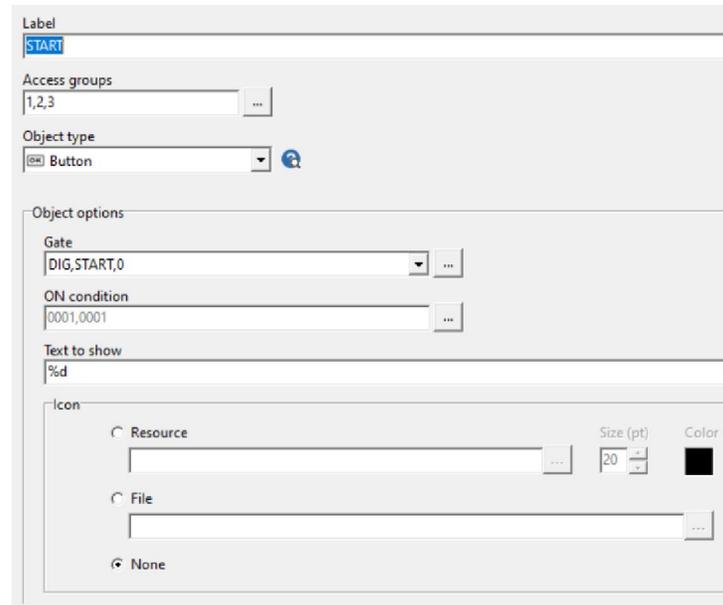
Se puede indicar el ancho de la barra deslizadora junto con los valores mínimo, máximo y el incremento, en este ejemplo el mínimo está seteado en 0, el máximo en 40 y el incremento es de 1.

### 8.3.7. BUTTON

Permite alternar entre dos valores. Al contrario que el switch, no tiene memoria, es decir que vuelve a su estado original luego de ser presionado.



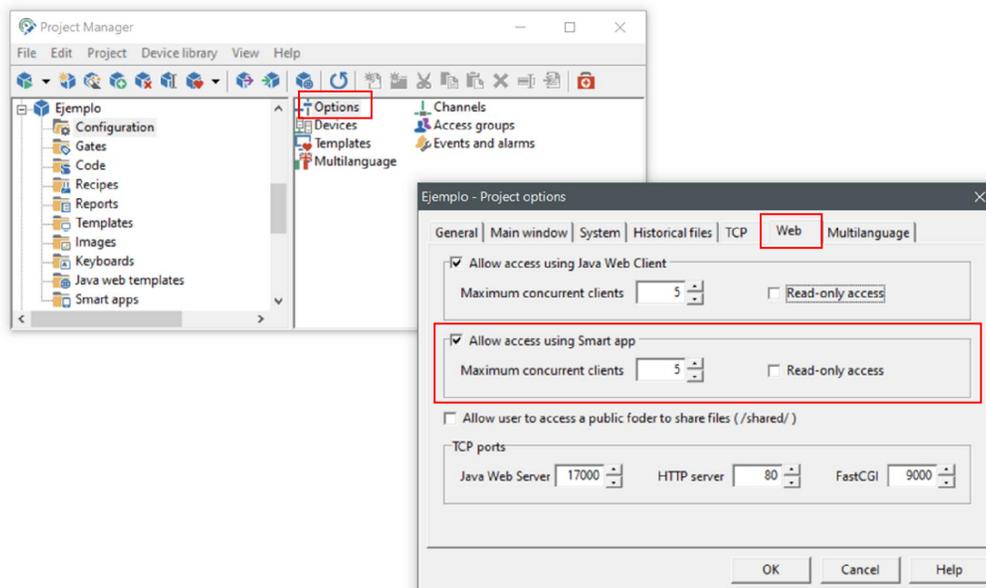
Se indica un nombre (Label), se le deben asociar los grupos de acceso y como tipo de objeto se elige Button.



En opciones de objeto se selecciona el Tag, que debe ser de tipo Digital, lectura/escritura.  
 Por defecto para el estado On envía un 1, pero se puede modificar este valor de ser necesario.  
 El texto a mostrar es el valor del Tag (0 o 1), puede reemplazarse por un texto estático, por ejemplo START.

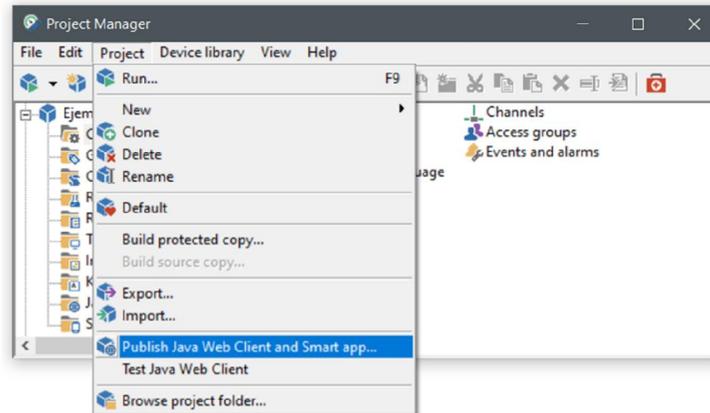
#### 8.4. EJECUCIÓN DE LA SMART APP

Con la aplicación creada se debe habilitar el acceso a la SmartApp desde opciones → Web → Allow Access using Smart App

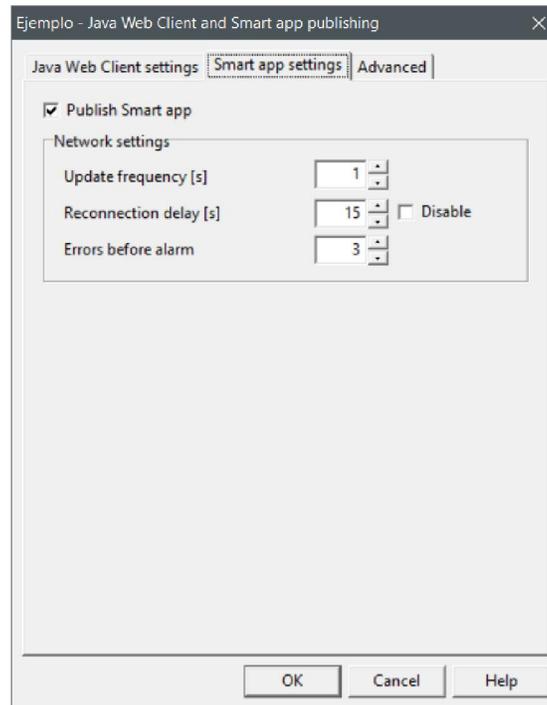


Por defecto el acceso es de solo Lectura, si es necesario modificar valores en forma remota se debe destildar Read-only Access, también es posible indicar el número máximo de usuarios que pueden estar conectados al mismo tiempo.

Finalmente es necesario Publicar la SmartApp, esto se consigue llando a Project → Publish Java Web Client and Smart app (es necesario volver a publicar la SmartApp si se han realizado modificaciones a la misma, para que tomen efecto los cambios)



En la ventana que se abre seleccionar la lengüeta Smat App settings y tildar Publish Smart App.



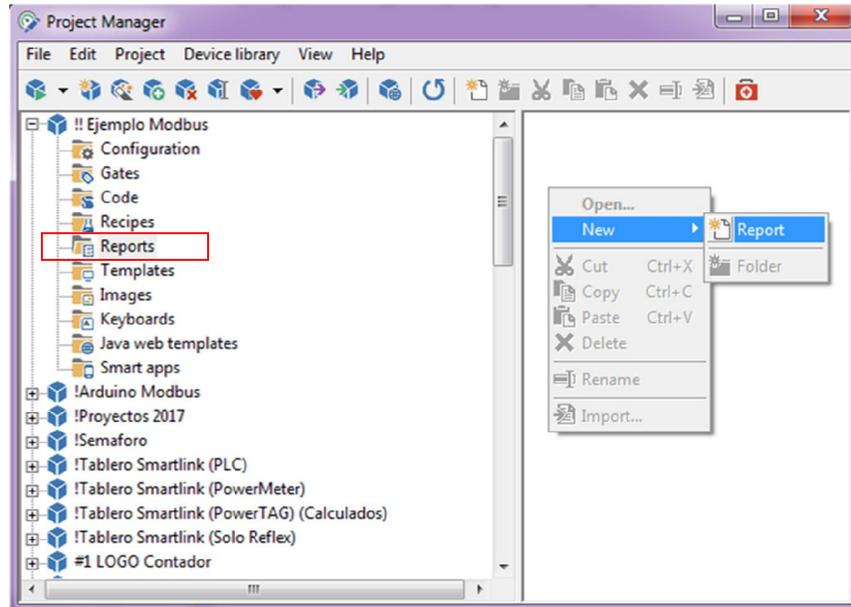
Se puede indicar la frecuencia de actualización de valores, por defecto son 5 segundos, mientras mas bajo es este valor mas frecuentemente se actualizarán los datos mostrados. En el ejemplo se ha elegido un tiempo de actualización de 1 segundo.

## ANEXO I

### 9. GENERACIÓN DE REPORTE.

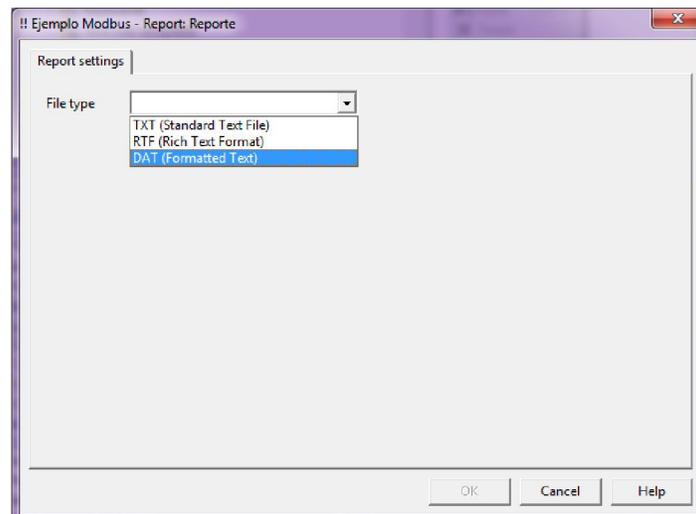
Winlog permite la generación de reportes, dicho de otra manera, permite generar un archivo donde se almacenen los valores de los TAGs que uno desee. Esto permite exportar estos datos para ser procesados por otro tipo de programas (por ejemplo una planilla de cálculo).

Los reportes se crean desde “Reports”, en el panel de la derecha se puede crear un nuevo reporte:



En este ejemplo se le ha dado el nombre: **Reporte**

Haciendo doble clic sobre Reporte se puede acceder a las propiedades del mismo, dentro de estas se define el tipo de reporte a crear:



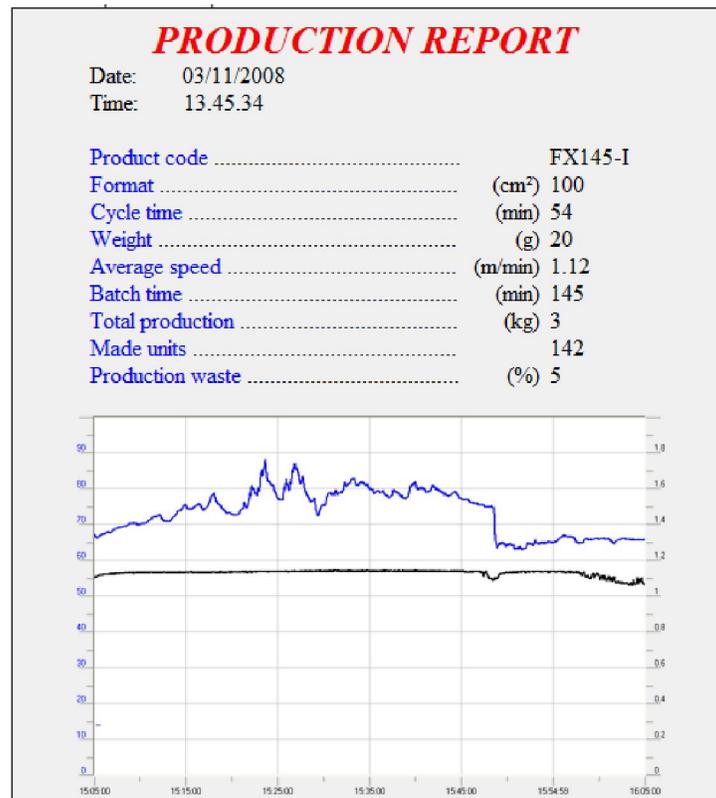
Los Reportes TXT se utilizan para generar informes con valores de las variables elegidas, están orientados mas a la presentacion de un reporte de situación que a un histórico de a variable, como el nombre lo indica, el reporte es generado en texto sin formato.

```

***** PRODUCTION REPORT *****
Date: 03/11/2008   Time: 13.45.34

Product code ..... FX145-I
Format ..... (cm²) 100
Cycle time ..... (min) 54
Weight ..... (g) 20
Average speed ..... (m/min) 1.12
Batch time ..... (min) 145
Total production ..... (kg) 3
Made units ..... 142
Production waste ..... (%) 5
    
```

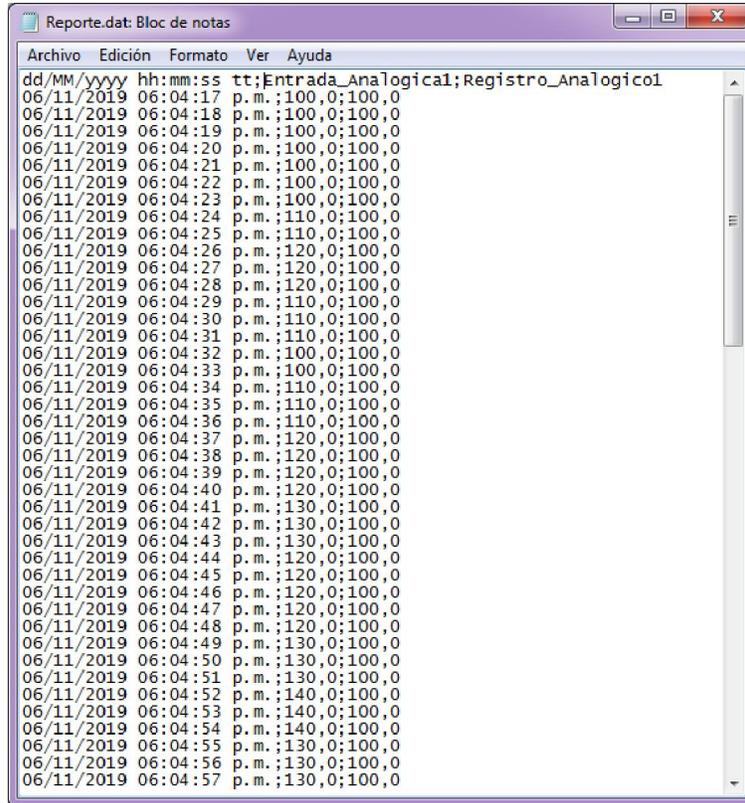
Un reporte RTF es similar, pero permite generar los reportes usando texto con formato y con la posibilidad de insertar gráficos.



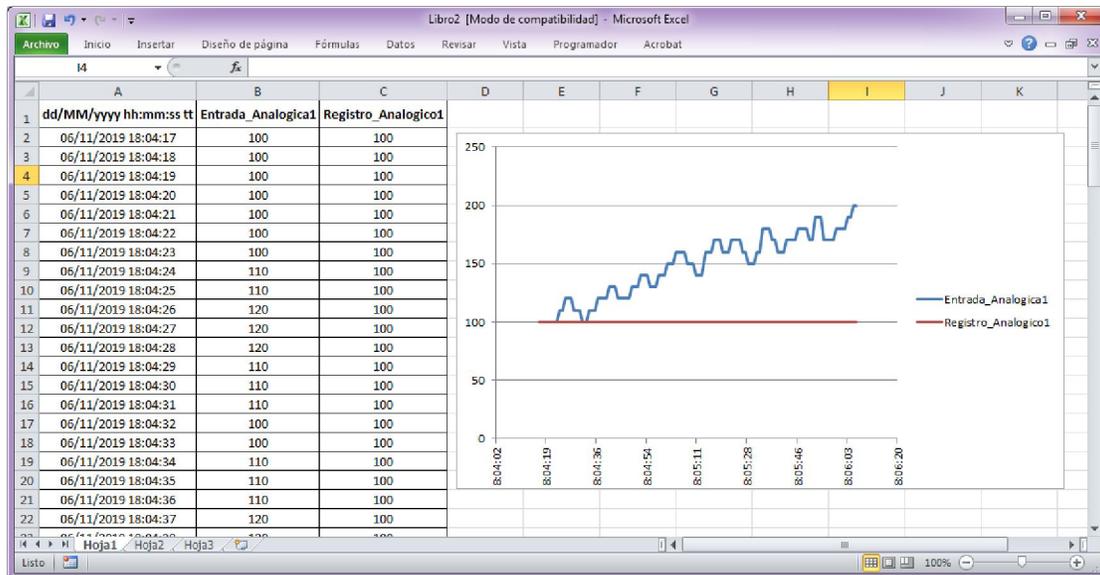
Los reportes DAT permiten generar archivos donde se almacenan los valores que ha tomado una o mas variables durante el tiempo, estos pueden ser almacenados en formato separado por comas y pueden ser fácilmente importados por otras aplicaciones.

En el ejemplo a continuación pueden verse tres columnas separadas por ;

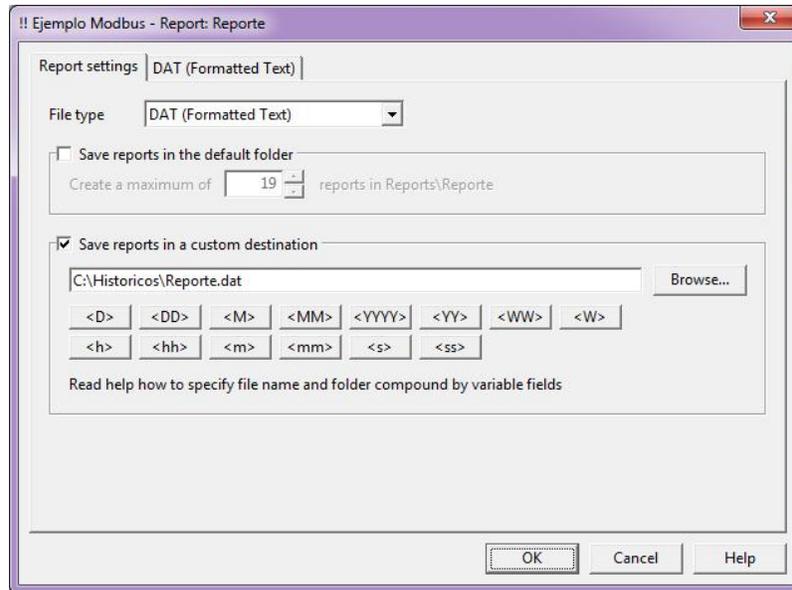
La primera columna contiene la fecha y hora (*dd/MM/yyyy hh:mm:ss tt*) la segunda columna contiene los valores del TAG ***Entrada\_Analogical1*** y la tercer columna contiene los valores del TAG ***Registro\_Analogico1***



Estos valores pueden ser importados por otra aplicación, por ejemplo una planilla de cálculo:



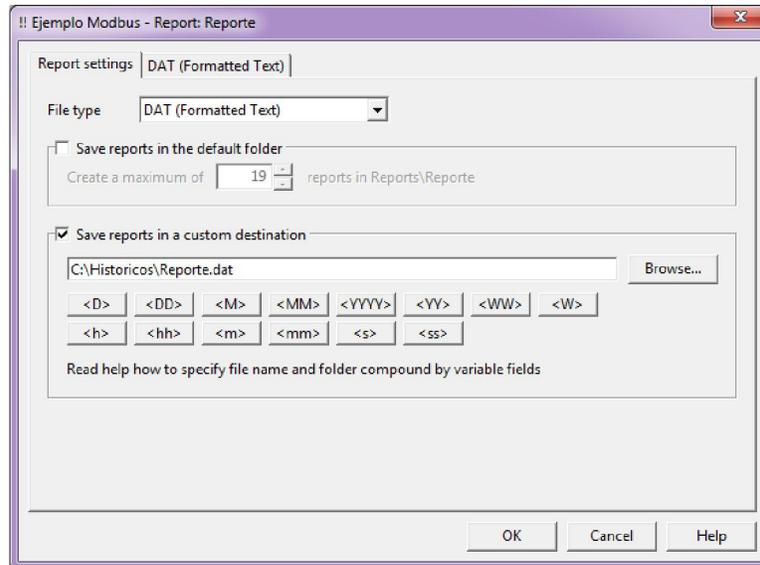
Una vez elegido el tipo de reporte (en este caso DAT) se puede elegir el directorio donde se desea almacenar la información (si no se elige esta opción el reporte se almacenará dentro del directorio de Winlog: C:\ProgramData\Winlog Lite 3\Projects\*<Nombre del proyecto>*\Reports)



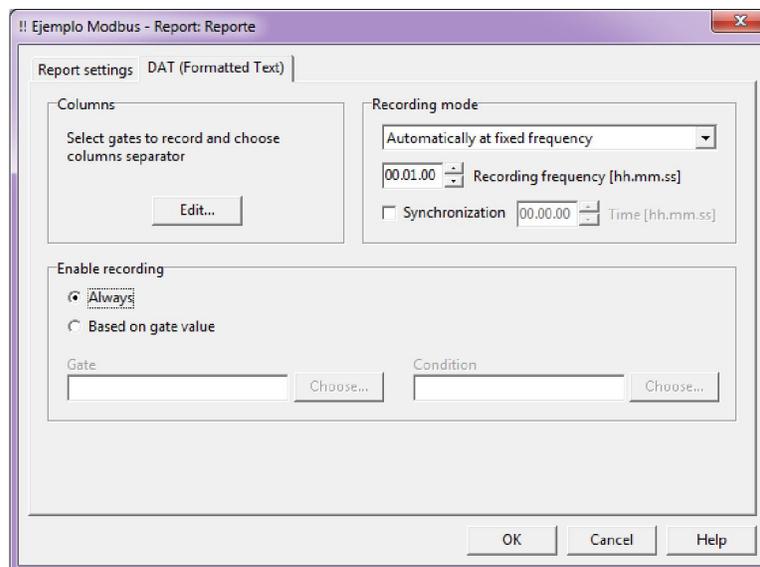
En la lengüeta DAT (Formatted Text) se eligen:

- Los TAGs que serán almacenados.
- La frecuencia de la toma de datos, que puede ser:
  - Automática, con una frecuencia Fija. Hay que tener en cuenta en este punto que una frecuencia (por ejemplo almacenar un dato cada segundo) generará archivos de gran tamaño.
  - Manual, en este caso se define la frecuencia mediante un Script.
- Cuando se comenzará la toma de datos:
  - Always: se empiezan a tomar datos desde el momento que se ejecuta el SCADA
  - Based on Gate Value: los datos se comenzarán a tomar cuando un TAG cumpla con una condición, por ejemplo mientras una llave (TAG digital) sea igual a uno (1).

En este ejemplo los datos se almacenarán en el directorio llamado ***Histórico*** ubicado en el disco **C:** el nombre del archivo será ***Reporte.dat***

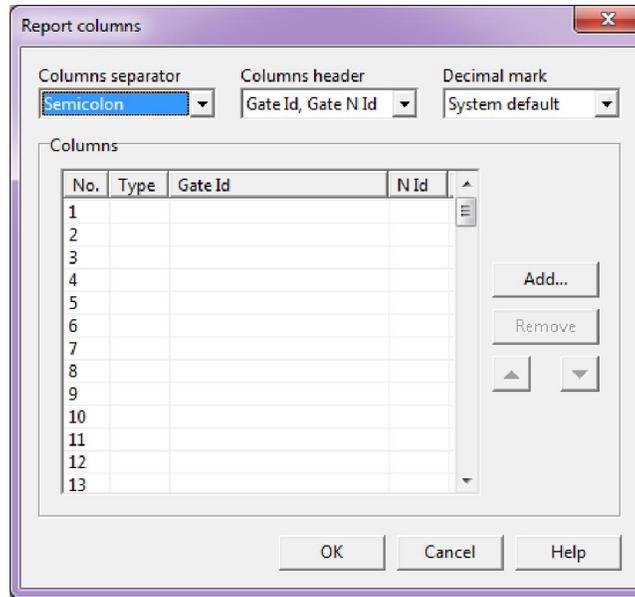


Los datos se almacenarán en forma permanente (Always) y con una frecuencia de 1 minuto.



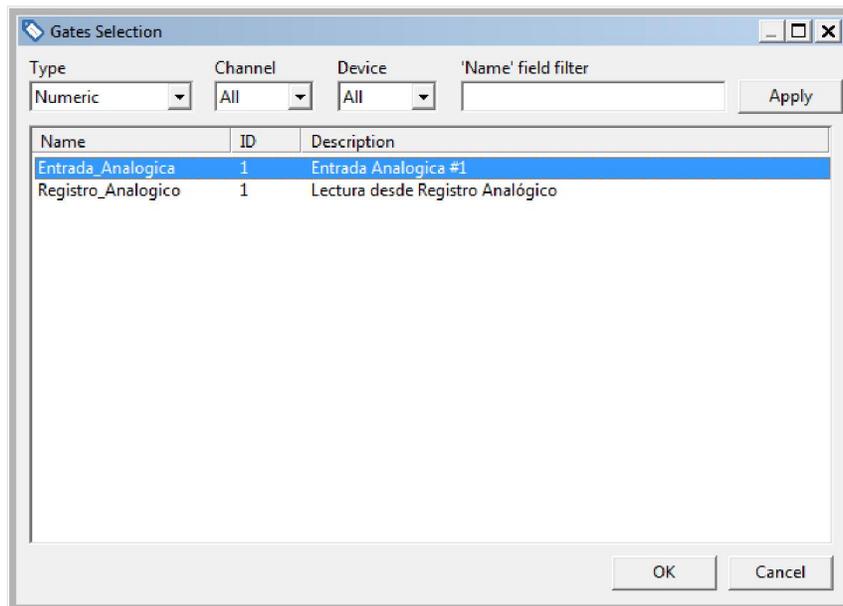
Los TAGs que serán almacenados se elijen haciendo clic en el botón **Edit...**

En la ventana que aparece se elije el separado de columnas, se recomienda punto y coma (;) o tabulador.

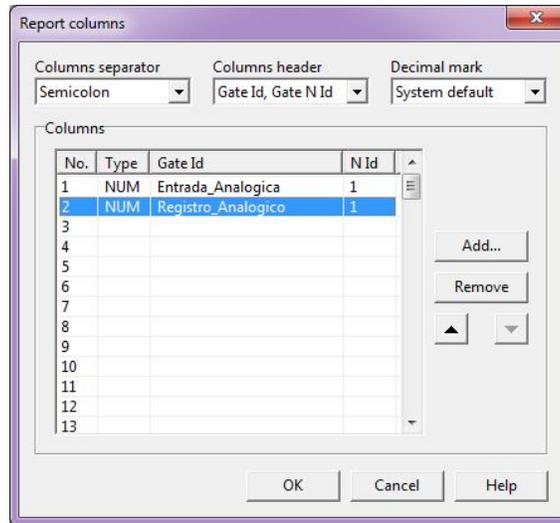


Para agregar los TAGs se hace clic en el botón **Add...**

En la ventana de selección de TAGs el que se desean almacenar, por ejemplo en nuestro caso el TAG numérico (Analógico) **Entrada\_Analogica1**



Se repite el proceso para los TAGs que se desee agregar.



En estas condiciones se da OK a las ventanas del reporte y el mismo queda listo para comenzar a almacenar los valores cuando se ejecute el SCADA.