



D-AUT-002

Versión: **2**
Junio 2016

Gerencia de Comunicaciones y
Automatismos
Dirección de Apoyo Logístico
Agua y Saneamientos Argentinos

Normas de programación de PLC



Contenido

1. OBJETIVO	3
2. ALCANCE.....	3
3. REFERENCIAS	3
4. DEFINICIONES Y ABREVIATURAS	3
5. DESARROLLO	4
5.1. SOFTWARE DE PROGRAMACIÓN.	4
5.2. LENGUAJE DE PROGRAMACIÓN.....	4
5.3. AUTÓMATAS LÍNEA SCHNEIDER ELECTRIC	4
5.3.1. CONFIGURACIÓN DEL PROYECTO	4
5.3.2. ESTRUCTURA DE LA APLICACIÓN	6
5.3.3. SUBROUTINAS	9
5.3.4. ORGANIZACIÓN DE TABLAS DE PALABRAS PARA COMUNICACIÓN. ..	10
5.3.5. VARIABLES	10
5.3.6. NOMENCLATURA DE LAS ENTRADAS/SALIDAS FÍSICAS.	11
5.3.7. NOMENCLATURA DE VARIABLES	13
5.3.8. NOMENCLATURA DE VARIABLES DE USO GENERAL.	13
5.3.9. USO DE LAS VARIABLES A COMUNICAR AL SCADA.....	14
5.3.10. IDENTIFICACIÓN DE LAS PALABRAS A COMUNICAR AL SCADA	14
5.3.11. CONFIGURACIÓN DE LAS COMUNICACIONES.....	16
5.3.12. CONFIGURACIÓN DE ENTRADAS/SALIDAS ANALÓGICAS.	19
5.3.13. VARIABLES MÍNIMAS A PROGRAMAR	21
5.3.14. ALARMAS Y FALLAS MÍNIMAS A PROGRAMAR.....	22



1. Objetivo

El presente documento define los lineamientos básicos que debe seguir la empresa contratista que ha sido adjudicada para el desarrollo de las lógicas de todos los PLC que se encuentren dentro del alcance del proyecto.

Estos lineamientos los deben respetar tanto los PLC que conformen el sistema de control de planta (para el caso que aplicara) como así también los PLC que formen parte de un equipo “paquetizado”.

El mismo define cuestiones generales que se deben respetar para todo tipo de aplicaciones. Cualquier duda o consulta por el desarrollo de la aplicación se deberá realizar al Departamento de Automatismos y Sistemas de Control (AySC) canalizada a través de la inspección de obra, así como también las entregas parciales de la aplicación. Se deberá enviar la aplicación para aprobación en un estado de avance inicial (Definición y creación de variables) a AySC, de modo de validar que se cumpla la presente normativa.

2. Alcance

Este procedimiento se aplica para todos los desarrollos de programas de PLC que deban instalarse en cualquiera de los establecimientos de Agua y Saneamiento dentro del radio de acción de la empresa.

3. Referencias

D-AUT-002 Normas de implementación Topkapi.

D-AUT-005 Normas de implementación de aplicaciones HMI.

4. Definiciones y Abreviaturas

AySC: Departamento de Automatismos y Sistemas de Control.

HMI: Interface Hombre-Máquina (Human Machine Interface)

PLC: Controlador lógico programable.

ET: Especificación Técnica.

HARDWARE: Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

SOFTWARE: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

SCADA: Supervisión, Control y Adquisición de datos (Supervisory Control and Data Acquisition).

TOPKAPI: Sistema SCADA utilizado en AySA.



5. Desarrollo

5.1. Software de Programación.

Cualquiera sea el software a utilizar para programación de PLC (Ej.: Unity Pro, PL7 Pro, Concept, Twido Suite, SoMachine, etc.), deberá ser la misma versión que al momento esté vigente en AySA; de utilizarse una versión superior, la empresa contratista encargada de la programación, deberá entregar una copia original del software, con los derechos y/o Licencias correspondientes a nombre de Agua y Saneamientos Argentinos S.A., al Dto. Automatismos y Sistemas de Control.

5.2. Lenguaje de programación

Todos los programas a desarrollar deberán ser desarrollados en lenguaje **Ladder exclusivamente**.

En caso de que el proceso amerite otro tipo de lenguaje (Procesos secuenciales) se deberá realizar la consulta a AySC quien aprobará o no su uso, en función del justificativo del mismo.

5.3. Autómatas línea Schneider Electric

5.3.1. Configuración del proyecto

Todo proyecto debe estar configurado como para que sean posibles las siguientes acciones:

- 1) Conectarse al PLC y transferir el programa a una PC sin tener el programa original.
- 2) Realizar modificaciones online.
- 3) Crear nuevos bloques de datos online.
- 4) Permitir matrices dinámicas

Para garantizar estas posibilidades, el proyecto debe tener los ajustes configurados como se muestra a continuación:

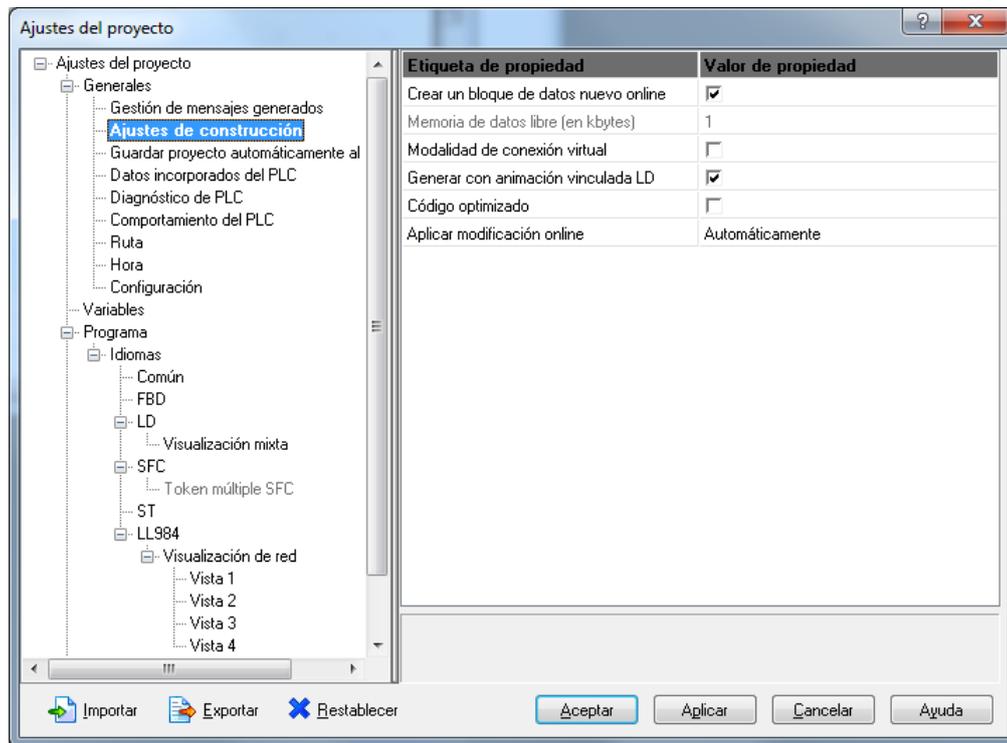


Figura 1 – Ajustes del proyecto

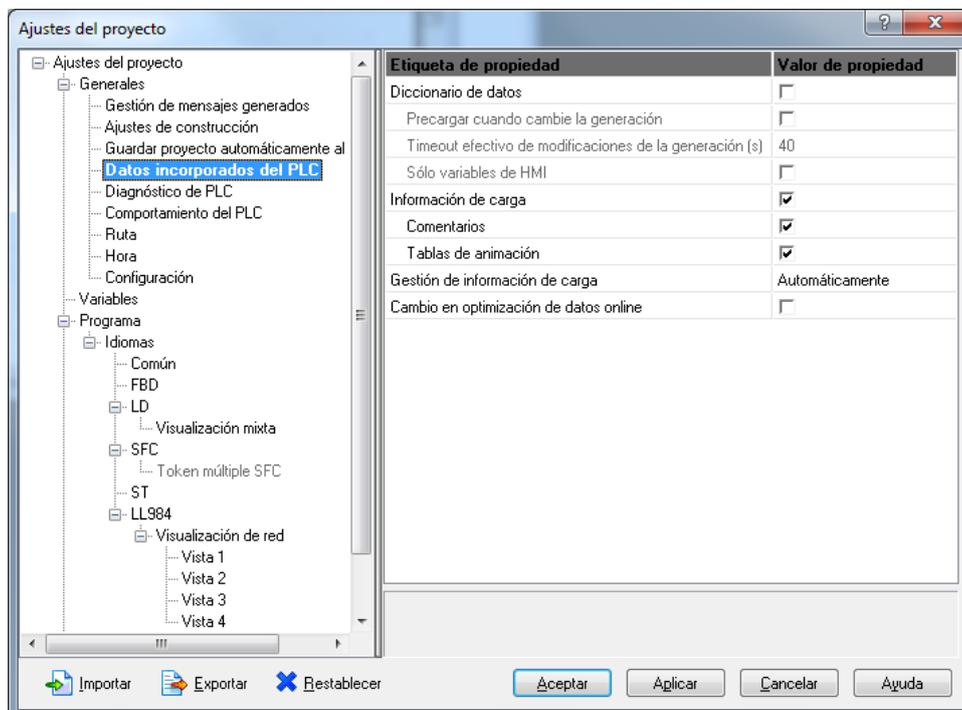


Figura 2 – Ajustes del proyecto

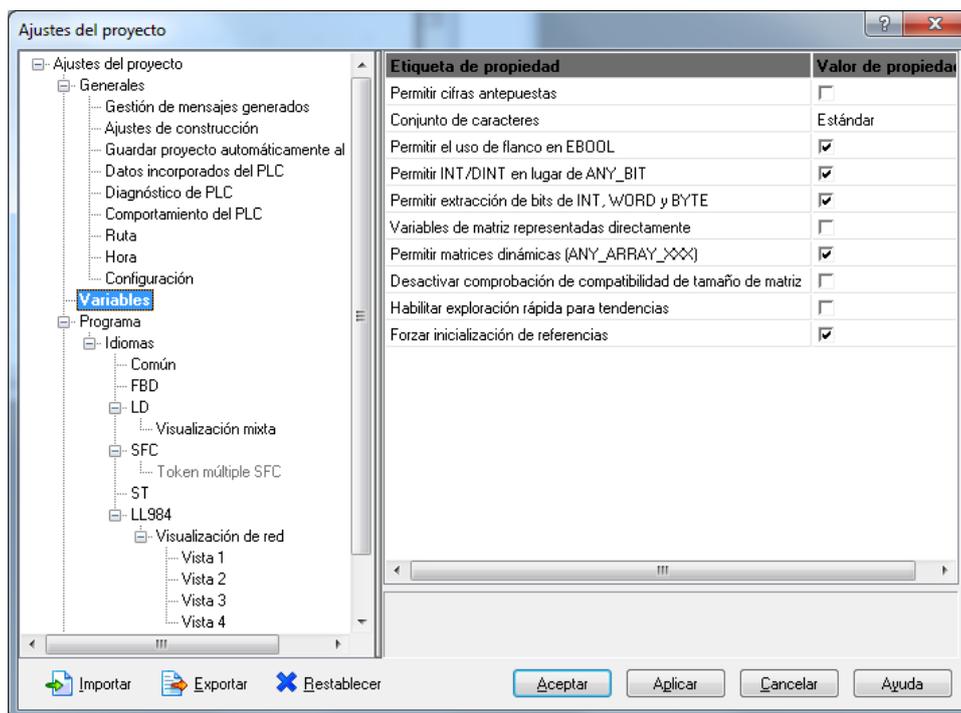


Figura 3 – Ajustes del proyecto

5.3.2. Estructura de la aplicación

Las aplicaciones estarán estructuradas de la de la siguiente forma:

La tarea MAST estará dividida en Secciones. Una primera Sección Principal en la cual se realizarán, por ej. : todas las asignaciones de entradas físicas a Palabras (%MWxx) como así también asignación de bits o variables generadas a partir de procesos lógicos internos y quedando ordenados para ser comunicados a algún SCADA o a otro autómatas.

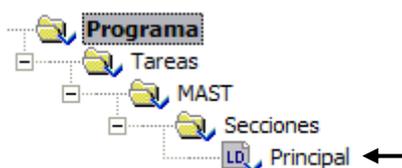


Figura 4 – Estructura de la aplicación

Dependiendo del tipo de instalación a automatizar, se irán incorporando nuevas secciones; por ej. : para una planta que cuenta con varias bombas (más de una) se generarán secciones por cada grupo (conjunto bombas/válvulas, etc.). En cada sección se gestionará la marcha, parada, proceso de fallas, etc. de cada uno de los grupos.



Figura 5 – Estructura de la aplicación

Todas las mediciones ingresadas al autómata, cualquiera sea la forma de ingreso al mismo (4 a 20 mA, ethernet, etc.) deberán ser agrupadas en una sección, Por ej. : caudalímetros, medidores de presión, instrumentos de calidad, etc.. En esta sección se realizarán las comparaciones y se generarán alarmas o bits de habilitación que sean necesarios.

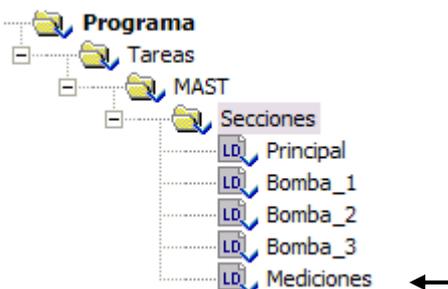


Figura 6 – Estructura de la aplicación

Si fuera necesario realizar gestiones de comunicación, con sus respectivas instrucciones (WRITE_VAR o READ_VAR) deberán ser agrupadas en una nueva sección.



Figura 7 – Estructura de la aplicación

Todas las fallas provenientes de señales externas, como así también las generadas por procesos internos, serán seteadas y bloquearán el funcionamiento del equipo correspondiente. Los reset de estos bits seteados serán agrupados en una sección dedicada.

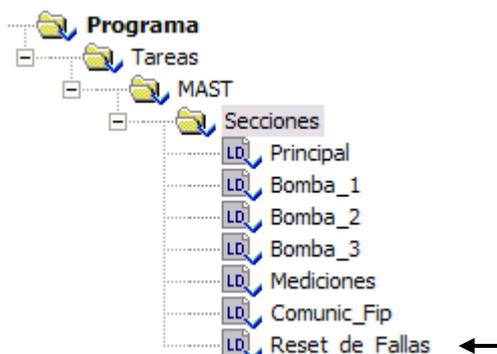


Figura 8 – Estructura de la aplicación

Teniendo en cuenta este concepto, cada proceso de consideración deberá ser ordenado en una sección a tal efecto.

Cada una de las secciones deberá llevar un breve comentario en el cual se informan los procesos que se realicen dentro de la misma.

Ej.:

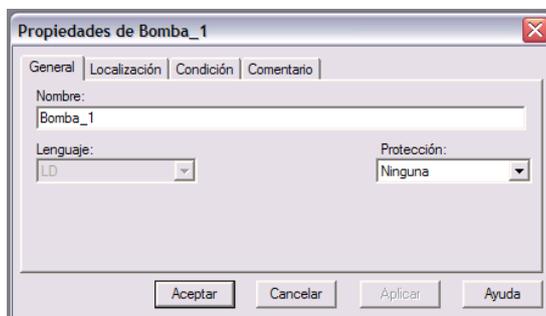


Figura 9 – Nombre de la tarea y comentarios

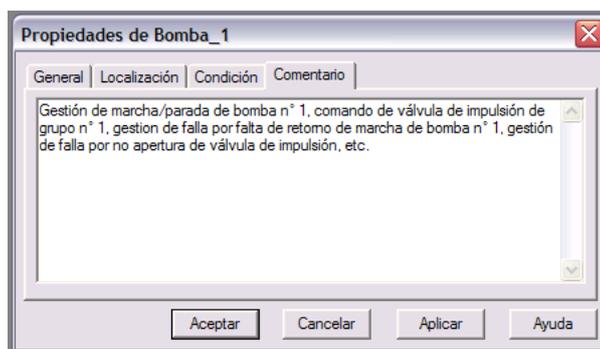


Figura 10 – Nombre de la tarea y comentarios

5.3.3. Subrutinas

De ser necesario se podrán utilizar Secciones SR (Subrutinas), éstas estarán configuradas de la misma forma que las secciones antes mencionadas y programadas con el mismo lenguaje (LD).

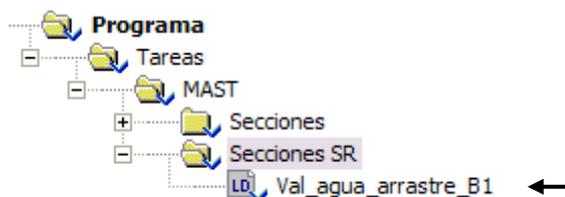


Figura 11 – Estructura de la aplicación (Subrutinas)

Las subrutinas podrán ser llamadas de cualquiera de las secciones.

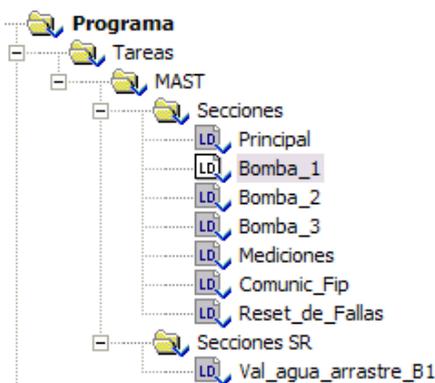


Figura 12 – Nombre de la tarea y subrutina

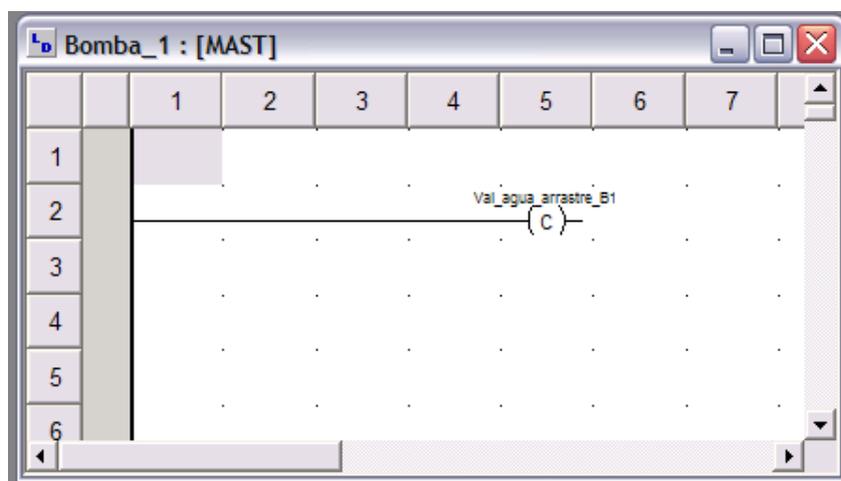


Figura 13 – Llamado a subrutina

5.3.4. Organización de tablas de palabras para comunicación.

Todas las entradas físicas, digitales y analógicas, deberán ser transferidas a tablas de palabras (%MWxx).

Para los TELECOMANDOS recibidos desde el Topkapi, deberán utilizarse variables de punto flotante (%MFxxx).

Las salidas digitales no serán transferidas, salvo que se trate de algún equipamiento a comandar que no cuente con señalización (Ej.: vál. Solenoide).

Los Bits internos cuyo estado merezca comunicarse al SCADA también deberán ser transferidos a tablas de palabras.

Ordenamiento de las tablas de memoria:

Tipo	Desde	Hasta
Entradas Digitales:	%MW100	%MW199
Entradas Analógicas	%MW200	%MW249
Entradas Analógicas (punto flotante)	%MF250	%MF299
Horas funcionamiento De máquinas	%MD300	%MD348
Bits o palabras a comunicar (READ/WRITE)	%MW350	%MW399
Palabras para uso general interno	%MW400	%MW599
Telecomandos digitales	%MW600	%MW624
Telecomandos analógicos (enteros)	%MW625	%MW649
Telecomandos analógicos (flotante)	%MF650	%MF698

Figura 14 – Tabla de organización de memoria interna

5.3.5. Variables

Toda variable utilizada en el programa, por más que sea de uso auxiliar, deberá tener un TAG y comentario asociado. El tag deberá ser significativo, deberá proveer información sobre la variable. No se permitirá el uso de variables empleando la dirección de memoria directa de las mismas. Esto se muestra a continuación:

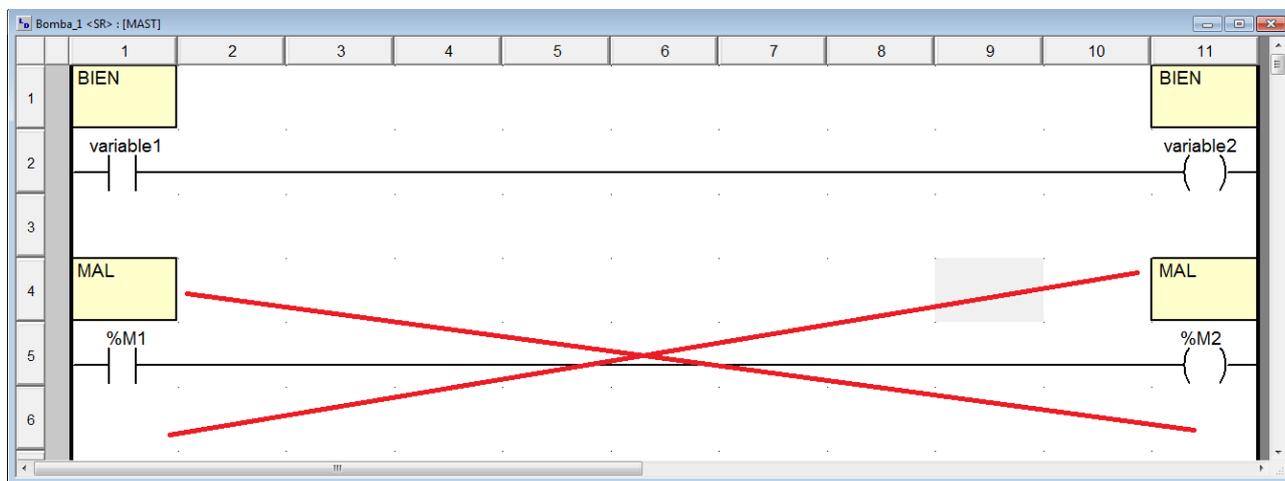


Figura 15 – Uso de variables

No se aceptará el uso de las variables propias de un bloque. Se deberán definir variables auxiliares cuando sea necesario. Estas variables auxiliares deberán tener un TAG significativo. Esto se muestra a continuación:

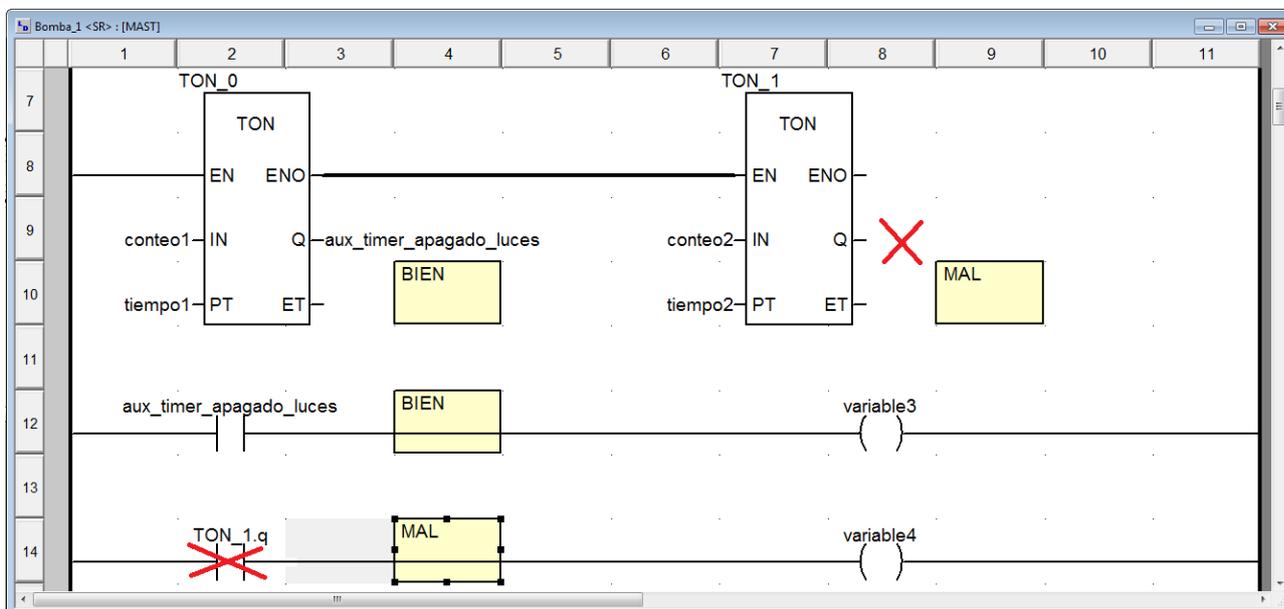


Figura 16 – Uso de variables

5.3.6. Nomenclatura de las entradas/salidas físicas.

Todas las entradas y salidas físicas deberán estar identificadas con un símbolo o mnemónico (TAG). El formato del símbolo estará compuesto por iniciales o abreviaturas, separadas por guión bajo, sin espacio.

En primer lugar, deberá ir el equipo (Ej.: **Bomba 1**) y luego la acción o información que se extrae del mismo (Ej.: **Marcha**).

Ejemplo:

Equipo_Acción
 %I0.2.0 = Bomba 1 en automático
 TAG: **B1_Auto**

Nombre	Tipo	Dirección	Valor	Comentario
B1_Auto	BOOL	%I0.2.0		
B1_Caudal	INT	%IW0.4.0		
B1_Marcha	BOOL	%Q0.3.0		
B1_var_veloc	INT	%QW0.5.0		

Figura 17 – Nomenclatura de entradas físicas

Algunos ejemplos de formas para resumir los nombres de los equipos o elementos de proceso podrán ser los siguientes:

Letra	Equipo
B	Bomba
V	Válvula
F	Filtro
BAT	Batería de filtros
TAM	Tamiz
TOR	Tornillo de Arquímedes
CL	Clarificador de arenas
CPT	Compactador
COMP	Compuerta

Figura 18 – Posible identificación de equipos

Los tag de las variables deberán ser significativos y proveer información sobre la variable. La longitud de los mismos no podrá ser mayor a 30 caracteres. Deben poseer una estructura como la que se indica en los siguientes ejemplos:

TAG ejemplo (máx. 30 caracteres)	comentario ejemplo	dirección ejemplo	Uso
B1_Marcha	Bomba 1, orden de marcha	%Q0.3.0	entrada/salida
B1_En_Marcha	Bomba 1, confirmación de marcha	%I0.1.0	entrada/salida
B1_Ref_Vel	Bomba 1, referencia de velocidad	%QW0.7.0	entrada/salida
V1_Abrir	Válvula 1, orden de abrir	%Q0.3.1	entrada/salida
V2_Cerrar	Válvula 2, orden de cerrar	%Q0.3.2	entrada/salida
V1_Abierta	Válvula 1, abierta	%I0.1.1	entrada/salida
V2_cerrada	Válvula 2, cerrada	%I0.1.2	entrada/salida
BAT9_F1_V_Dren_Abrir	Batería ix, filtro 1, válvula de drenaje, abrir	%Q0.3.3	entrada/salida
BAT9_F1_V_Dren_Abierta	Batería ix, filtro 1, válvula de drenaje abierta	%I0.1.3	entrada/salida
F1_Nivel	Filtro 1, nivel	%IW0.5.0	entrada/salida
B1_Caudal	bomba 1, caudal	%IW0.5.1	entrada/salida
TAM1_Motor_Cepillo_Marcha	Tamiz, motor de cepillo, orden de marcha	%Q0.3.4	entrada/salida

Figura 19 – Tabla ejemplo identificación de variables

Como se puede observar, algunas veces hay varios equipos (por ejemplo válvula de drenaje) de distintos sectores (por ejemplo filtro 1 de batería 9), por lo que se deberá identificar el sector antes que el equipo.

En casos excepcionales se permitirá omitir el guión bajo para separar palabras, escribiéndolas juntas, pero diferenciándolas con la inicial mayúscula de cada una.

5.3.7. Nomenclatura de variables

Cuando se deban incluir varias palabras en un tag, las mismas deberán estar separadas por un guión bajo y cada una deberá comenzar en mayúscula. Se pueden utilizar abreviaciones en caso que sea necesario para no excederse en la longitud máxima del tag (30 caracteres).

5.3.8. Nomenclatura de variables de uso general.

Se deberá utilizar un prefijo que indique qué tipo de variable es. Se muestra a continuación una tabla con los prefijos a utilizar:

Prefijo	Uso
SP_	Set Point
AL_	Alarma
AUX_	Variable auxiliar.
FAL_	Falla

Figura 20 – Prefijos para señales de uso general

En la tabla siguiente se muestran algunos ejemplos:

TAG ejemplo (máx. 30 caracteres)	comentario ejemplo	dirección ejemplo	Uso
SP_Nivel_F1	Set point de nivel de filtro 1	%MW1	Set point usado en programa
SP_Caudal_B1	Set point de caudal de bomba 1	%MW2	Set point usado en programa
AL_B1_No_Marcha	Alarma, bomba 1 no marcha	%M600	Alarma
AL_V1_no_abre	Alarma, válvula 1 no abre	%M601	Alarma
AUX_variable_1	Variable auxiliar 1	%M350	variable auxiliar

Figura 21 – Tabla ejemplo identificación de variables de uso general

5.3.9. Uso de las variables a comunicar al SCADA

Deben estar disponibles la mayor cantidad de variables para comunicar al TOPKAPI/HMI. Entre ellas se encuentran:

- a) Entradas físicas
- b) Salidas físicas
- c) Estado de selectoras
- d) Estado de térmicas
- e) Alarmas de equipos
- f) Alarmas de procesos
- g) Fallas de equipos
- h) Fallas de procesos
- i) Horas de marcha de equipos
- j) Condiciones de arranque de equipos
- k) Estados de procesos cuando hay algún proceso secuencial

Las variables que se utilicen para comunicar al TOPKAPI/HMI **NO deben usarse en el proceso del PLC**. Esto encapsula las variables del proceso en el PLC y crea una interfaz con el TOPKAPI/HMI, de modo que para cualquier cambio en el PLC que deba reflejarse en el TOPKAPI/HMI solo haya que cambiar la interfaz en el mapeo de variables.

Se muestra a continuación un esquema de ejemplificación:

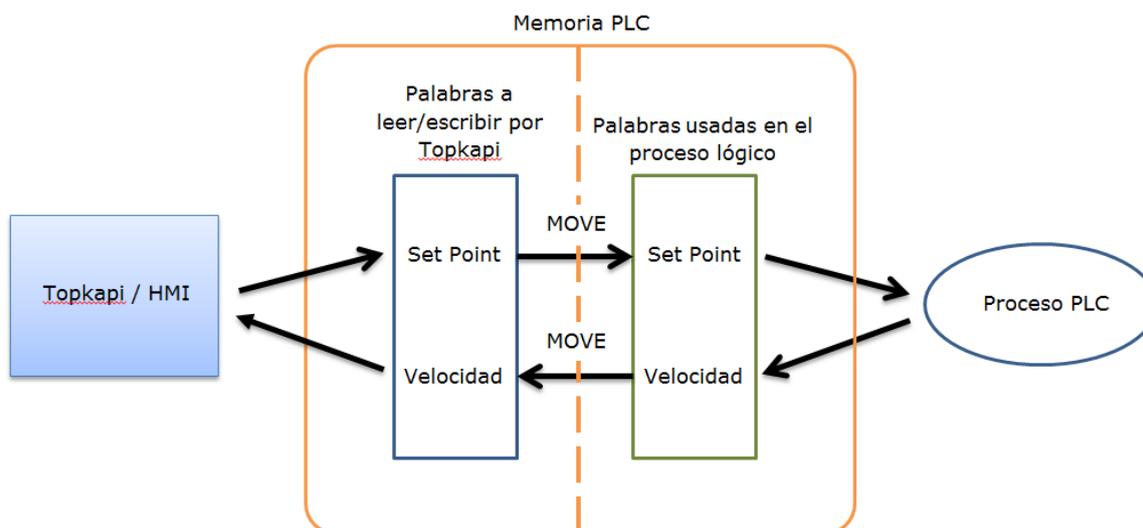


Figura 22 – Separación de variables usadas en el PLC y variables leídas por el SCADA

5.3.10. Identificación de las palabras a comunicar al SCADA

Se deben identificar las palabras que se transmitirán al SCADA Topkapi mediante el Símbolo y el comentario o descripción de la acción a realizar por vínculo de esta palabra. Es importante la correcta identificación para poder realizar la importación de estos datos, que se reportaran en el Topkapi mediante la herramienta SOFTLINK

incluida. Esta herramienta reconocerá el Tipo de Dato (Digital, Entero, Real, etc.), el Mnemónico o TAG (símbolo) y el Título para la celda (Comentario)

Símbolos / TAGs / mnemónicos de las variables.

El formato estandarizado de identificar el símbolo: debe ser con las iniciales y comenzando desde el sector del proceso afectado, luego el equipo al que se hace referencia y por último la acción o información que se extrae del mismo. Estos datos deben estar separados por "guiones bajos", sin espacios.

Ejemplo:

Sector_Equipo_Acción

%IO.2.0 => %MW100.0 => Bomba 1 en automático

Entonces:

Si el sector es la planta de PAC.

TAG: **PAC_B1_Auto** => %MW100.0

Si el sector es Planta de Cloro.

TAG: **CL_B1_Auto** => %MW100.0



Nombre	Tipo	Dirección	Valor
PAC_B1_Auto	BOOL	%MW100.0	

Figura 23 – Tabla de organización de memoria interna

Esto se debe a que, el TAG de las variables debe ser único en el SCADA, y el sistema no permitirá la importación de 2 variables que se llamen por ej. **B1_Auto** pertenecientes a 2 sectores distintos, si no se las diferencia de esta manera.

Comentarios de las variables.

El formato estandarizado para los comentarios debe ser la misma información usada en el símbolo, pero de manera más detallada para mayor compresión de la acción.

Se debe respetar el orden según Sector del proceso, detalle.

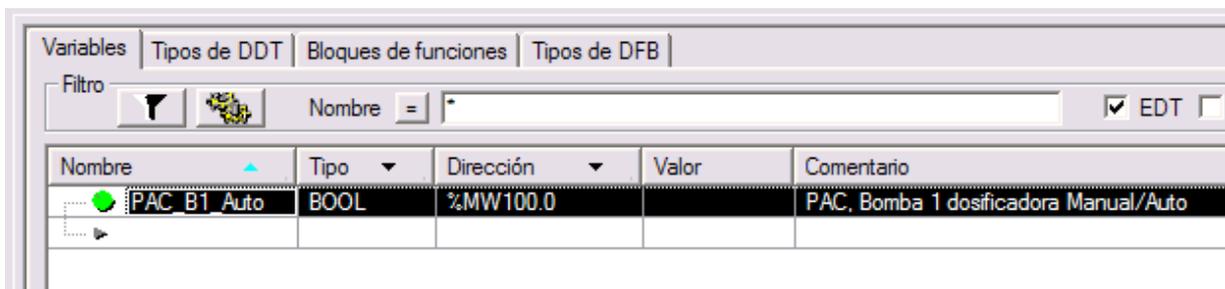
Ejemplo

Sector, Detalle

PAC, Bomba 1 dosificadora Manual/Auto

El comentario debe contener:

- Información detallada de la acción.
- El texto debe tener un máximo de 0 a 40 caracteres de largo para ser compatible con Topkapi.



Nombre	Tipo	Dirección	Valor	Comentario
PAC B1 Auto	BOOL	%MW100.0		PAC, Bomba 1 dosificadora Manual/Auto

Figura 24 – Tabla de organización de memoria interna

5.3.11. Configuración de las comunicaciones.

Redes de comunicación.

Cada PLC deberá estar integrado dentro de una red Ethernet, por lo tanto deberá estar identificado con una dirección IP definida.



Figura 25 – Red de comunicación

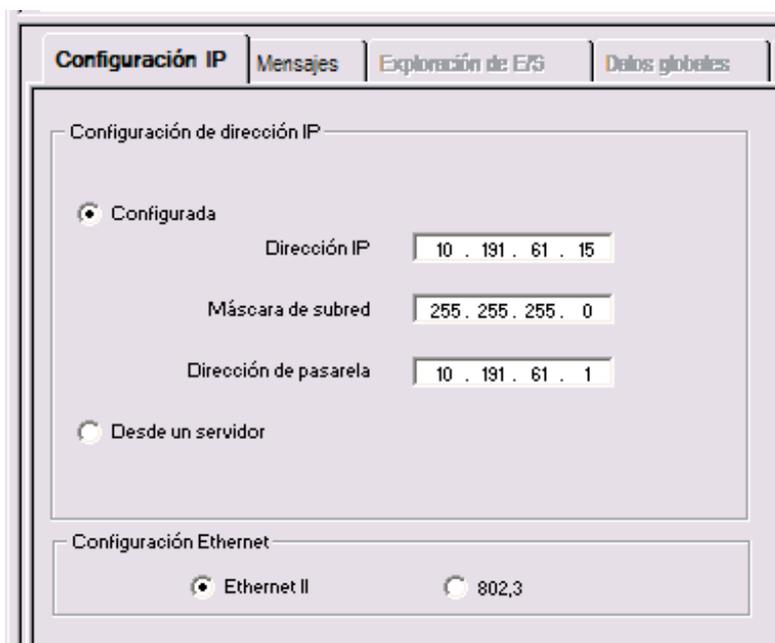


Figura 26 – Configuración de la red de comunicación

Las direcciones IP serán suministradas por el departamento de Netowrking, a través de una solicitud interna de AySC.

Es muy probable que se utilicen otros tipos de redes bajo otros protocolos de comunicación además de Ethernet (Modbus-IP), como por ejemplo Fipway o Modbus.

En todos los casos los números de estaciones para el caso de Fipway y los números Maestro/Esclavo para el caso de Modbus, serán definidos y suministrados por el Dto. Automatismo y Sistemas de Control.

IO Scanning

Para el caso de equipos que permitan esta funcionalidad, las comunicaciones deberán realizarse mediante este mecanismo, en lugar de bloques READ_VAR - WRITE_VAR.

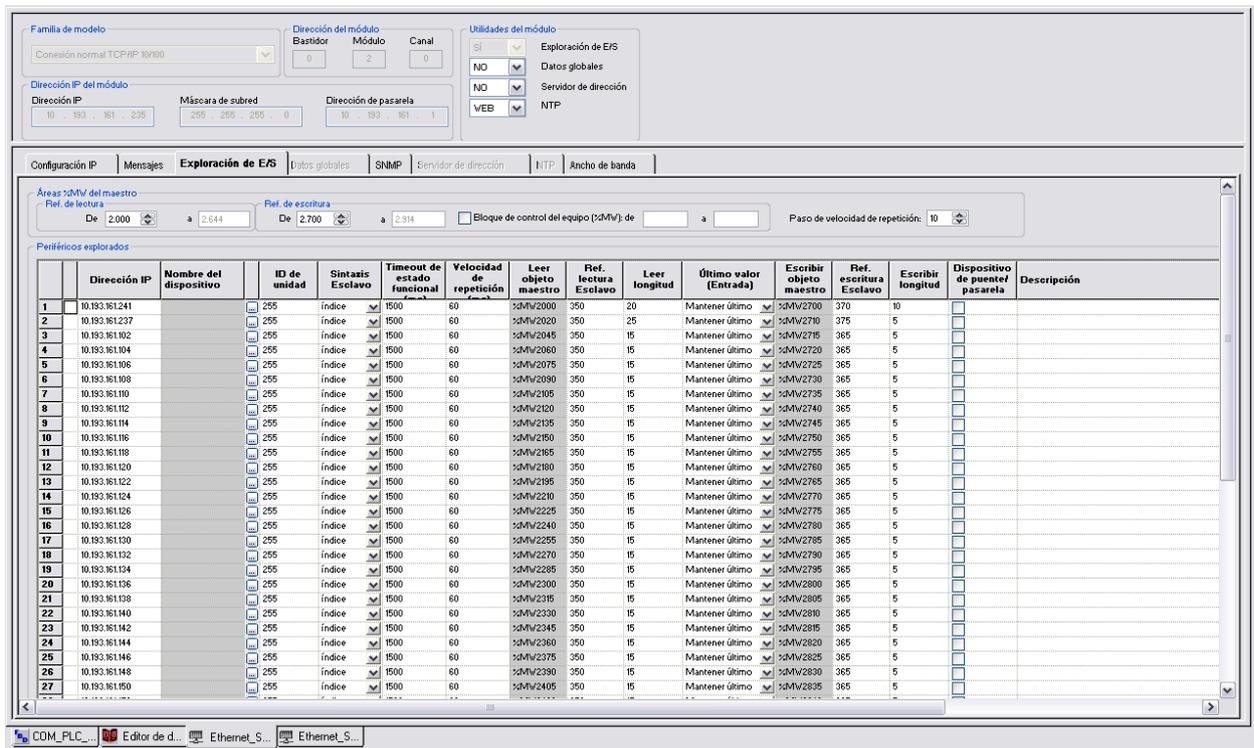


Figura 27 – Configuración de la exploración de entradas/salidas (IO Scanning)

Watch Dog

Para todos los PLC que tengan una comunicación con otro PLC, se deberá establecer un mecanismo de validación del estado del equipo y/o comunicación con el mismo a través de un mecanismo comúnmente conocido como Watch Dog.

El mecanismo a utilizar es el descrito a continuación:

- 1) En cada PLC, deberá existir un bit que alterne su estado (entre 0 y 1) cada 1 segundo.
- 2) El PLC que se comunique con este, deberá estar leyendo por comunicaciones este BIT.
- 3) En caso de no detectar el cambio de estado del bit en un lapso configurable (dependiendo la necesidad del proyecto) se deberá generar la respectiva señal de alarma, la cual deberá ser reportada al Topkapi y Panel HMI (En caso de existir) que reporten otras variables de ese equipo.

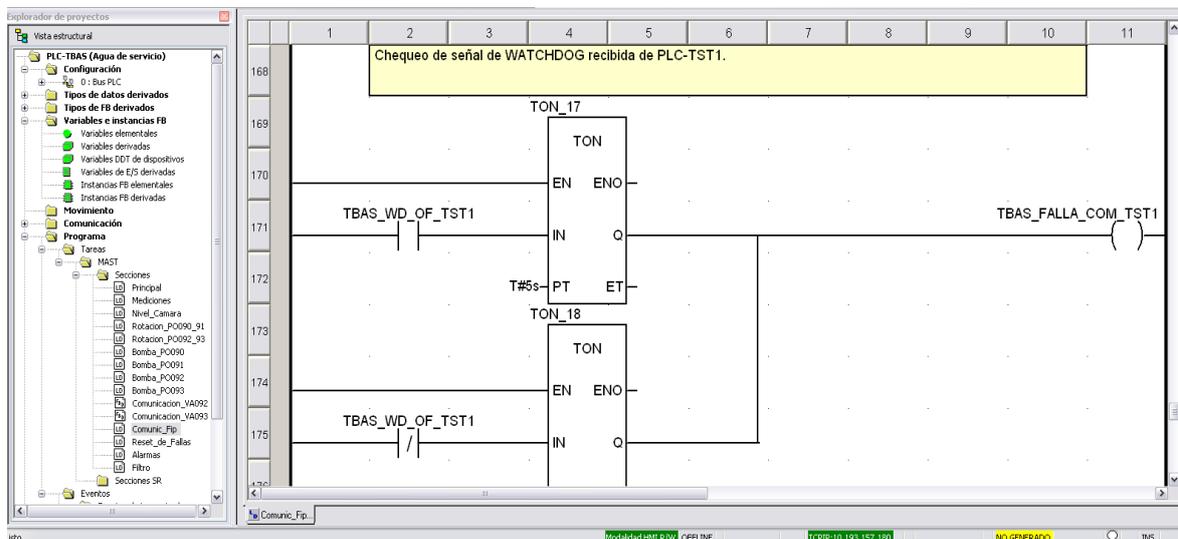


Figura 28 – Ejemplo de implementación de falla de comunicación

5.3.12. Configuración de Entradas/Salidas analógicas.

Entradas:

Todas las entradas analógicas utilizadas deberán estar escaladas en el cuadro de configuración del módulo, no deberán escalarse por programa.

Todas las entradas no utilizadas deberán estar desprovistas del tilde de habilitación del canal, para que de esta forma no se registren indicaciones de falla de módulo.

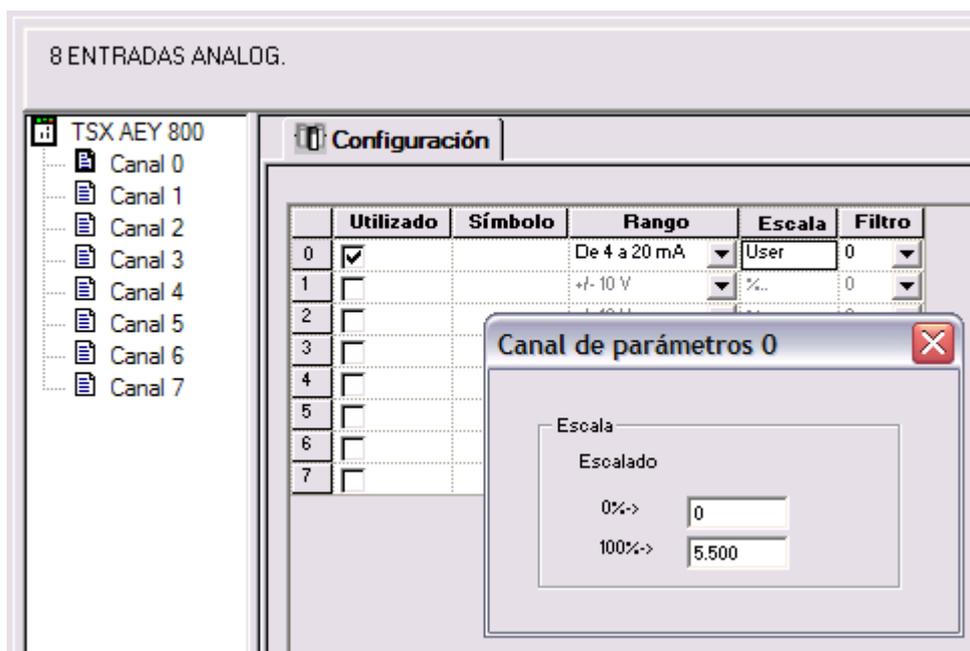


Figura 29 – Escalado de entradas analógicas

Salidas:

A diferencia de las entradas analógicas, las salidas analógicas no serán escaladas en el módulo, salvo solicitud expresa, las conversiones a valores de PLC (0 a 10.000) deberán estar hechos por programa.

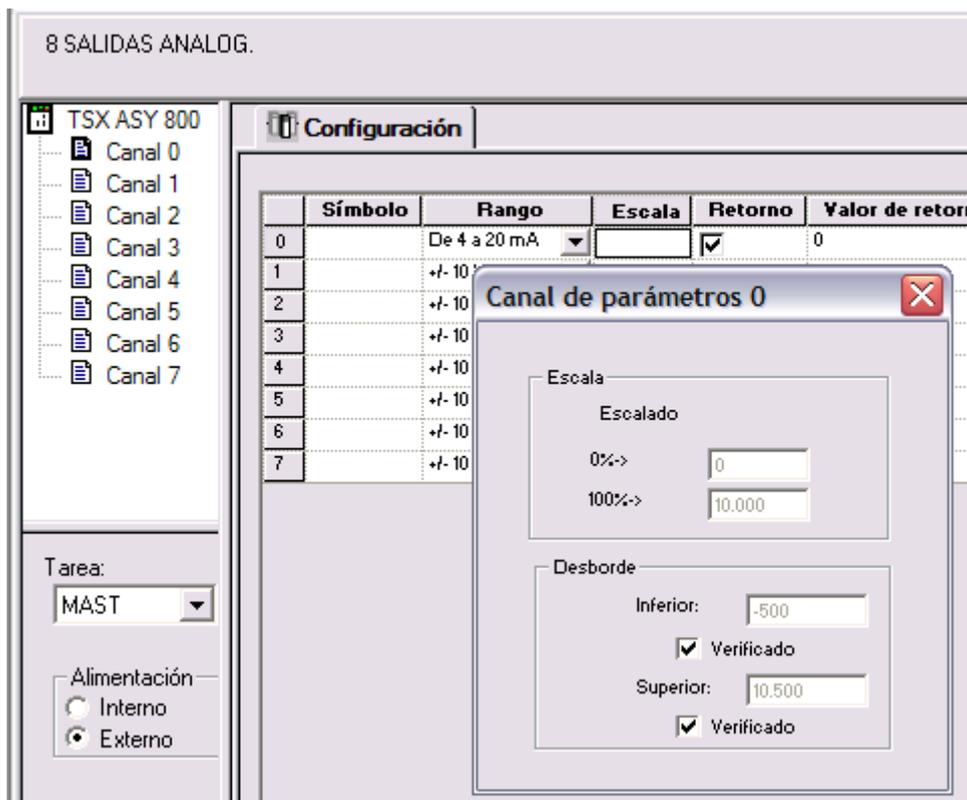


Figura 30 – Salidas analógicas

5.3.13. Variables mínimas a programar

Equipo	Variable a programar	Tipo de variable	Tipo de dato
Motor	Orden de marcha	Salida	EBOOL
	Realimentación de marcha	Entrada	EBOOL
	Estado guardamotor	Entrada	EBOOL
	Selectora en AUTO	Entrada	EBOOL
	Selectora en Manual	Entrada	EBOOL
	Horas de marcha de motor	Interna	DINT
	Referencia de velocidad variador (si existe variador)	Salida	INT
	Variador en falla (si existe variador)	Entrada	EBOOL
	Variador listo (si existe variador)	Entrada	EBOOL
Bomba	Horas de marcha de bomba (cuando pueden intercambiarse motores y bombas)	Interna	DINT
Válvula	Orden de abrir	Salida	EBOOL
	Orden de cerrar (si existe)	Salida	EBOOL
	Válvula abierta	Entrada	EBOOL
	Válvula cerrada	Entrada	EBOOL
Sensores analógicos	Señal del medición del sensor	Entrada	INT
	Señal de falla del sensor (si existe)	Entrada	EBOOL
Sensores digitales	Señal del sensor	Entrada	EBOOL
Pulsadores de emergencia	Estado de los pulsadores	Entrada	EBOOL
Térmica de alimentación de switch	Estado de térmica	Entrada	EBOOL
Térmica de alimentación de HMI	Estado de térmica	Entrada	EBOOL
Térmica de alimentación de PLC (220 V)	Estado de térmica	Entrada	EBOOL
Señales de UPS de PLC	Estado de UPS y alimentaciones	Entrada	EBOOL

Figura 31 – Tabla de variables que deberán ser programadas (en caso de existir el elemento físico)

Nota aclaratoria:

Las horas de marcha (tanto de bombas como de motores) deberán poder ser seteables, para poder reiniciar y/o configurar al momento del reemplazo del equipo.

5.3.14. Alarmas y fallas mínimas a programar

A continuación se muestra un listado de las alarmas / fallas que como mínimo se deben programar en el PLC y mapearlas para que las pueda mostrar el Topkapi/HMI.

Equipo	Tipo	Condición de alarma/falla
Motor	Falla	Si existe orden de marcha pero no hay confirmación de marcha durante de un tiempo definido (motor no marcha)
	Falla	No hay orden de marcha, pero sí hay confirmación de marcha durante un determinado tiempo (motor marcha sin señal)
	Falla	Variador de velocidad en falla (cuando hay variador)
	Falla	Protecciones motor (Humedad, Temperatura, Vibración, etc)
	Alarma	Seccionador de alimentación de motor abierto
	Alarma	Variador de velocidad no está listo (cuando hay variador)
	Alarma	Guardamotor abierto
Válvula	Falla	Existe orden de abrir pero no hay confirmación de apertura dentro de un tiempo definido (válvula no abre)
	Falla	Existe orden de cerrar pero no hay confirmación de cierre dentro de un tiempo definido (válvula no cierra)
	Falla	Incongruencia de sensores. Están activas (o desactivadas) las confirmaciones de apertura y cierre al mismo tiempo
	Falla	Existe orden de cerrar pero falta realimentación de señal "cerrando" (cuando existe esta señal)
	Falla	Existe orden de abrir pero falta realimentación de señal "abriendo" (cuando existe esta señal)
	Alarma	Falta alimentación a la válvula (por ejemplo, protección termomagnética abierta)
Sensor analógico	Falla	Está activa la señal de falla del sensor analógico
	Alarma	Incongruencia de sensores: hay dos indicaciones de sensores que son incompatibles entre sí.
	Alarma	Falta alimentación al sensor (por ejemplo, protección termomagnética abierta)
	Alarma	Sensor fuera de rango
	Alarma	Cable cortado
sensor digital	Alarma	Incongruencia de sensores: hay dos indicaciones de sensores que son incompatibles entre sí.
	Alarma	Falta alimentación al sensor (por ejemplo, protección termomagnética abierta)

Alimentación PLC	Alarma	PLC alimentándose de batería (tensión de 24 VCC no está presente por falla de fuente, de red de 220 o de termomagnética)
	Alarma	Tensión de 220 V no está presente (termomagnética abierta)
	Alarma	Batería de UPS descargada
PLC	Falla	Falla en canal
	Falla	Falla en módulo
	Alarma	Batería interna descargada
Switch	Falla	Termomagnética de switch abierta
HMI	Falla	Termomagnética de HMI abierta
General	Alarma	Tablero abierto
	Alarma	Cámara inundada

Figura 32 – Tabla de alarmas y fallas que se deberán programar (en caso de existir el elemento físico)

Nota aclaratoria:

Todas las fallas deberán quedar enclavadas con el reset de falla existente tanto como botón físico, como así también botones software (HMI y SCADA)