



www.infoPLC.net



Universidad de León

Escuela de Ingenierías Industrial e Informática
Titulación de Ingeniero Técnico Industrial (Especialidad en Electrónica,
Regulación y Automatismos)

Configuración Y Programación de Autómata MOELLER XC201 para Maqueta Industrial

Autor: Fernando casado García

Índice

Introducción

Estándar IEC 61131-3

Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

Índice

Introducción

Estándar IEC 61131-3

Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

Introducción

- El estándar IEC 61131-3 ha surgido en respuesta a la necesidad de unificar los lenguajes y el software de programación y comunicación de sistemas de control basados en autómatas programables (PLC).
- El autómata MOELLER XC201 cumple con el estándar IEC 61131-3.
- El autómata MOELLER XC201 dispone de conectividad Ethernet y un servidor OPC para intercambio de datos.
- El software CoDeSys es un entorno de programación para autómatas programables que cumplen el estándar IEC 61131-3.
- El software CoDeSys también incorpora un sistemas HMI/SCADA para la supervisión y control de las variables a través de un interfaz gráfico.

Índice

Introducción

Estándar IEC 61131-3

Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

Índice

Introducción

Estándar IEC 61131-3

Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

Estándar IEC 61131-3

El estándar internacional IEC 61131 es una colección completa de estándares referentes a autómatas programables y sus periféricos asociados. Consta de las siguientes partes:

- **Parte 1: información general**

Establece las definiciones e identifica las principales características relativas a la selección y aplicación de los autómatas programables y sus periféricos asociados.

- **Parte 2: equipo requerimientos y pruebas**

Especifica los requisitos del equipo y pruebas relacionadas para los autómatas programables (PLC) y sus periféricos asociados.

- **Parte 3: lenguajes de programación**

- Diagrama de Contactos
- Diagrama de Bloques de Funciones
- Lista de Instrucciones
- Texto estructurado.

Estándar IEC 61131-3

- **Parte 4: guías de usuario**

Un reporte técnico que proporciona una vista general y guías de aplicación del estándar para los usuarios finales de los controladores programables.

- **Parte 5: especificación del servicio de mensajería**

Define la comunicación de datos entre controladores programables y otros sistemas electrónicos usando el “Manufacturing Message Specification” (MMS, acorde al ISO/IEC 9506).

- **Parte 6: programación en lógica difusa**

Define los elementos básicos de programación de “lógica difusa” para su uso en Controladores programables.

- **Parte 7: guías para aplicación e implementación de lenguajes de programación**

Proporciona una guía para los desarrolladores de software para los lenguajes de programación definidos en la parte 3.

Índice

Introducción

Estándar IEC 61131-3

Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

Índice

Introducción

Estándar IEC 61131-3

Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

Lenguajes de programación

Elementos Comunes

- **Tipos de Datos**

- Estándar: (Booleanos, enteros, reales, byte, palabra...)
- Derivados.

- **Variables**

- Locales
- Globales

- **Unidades de organización del programa**

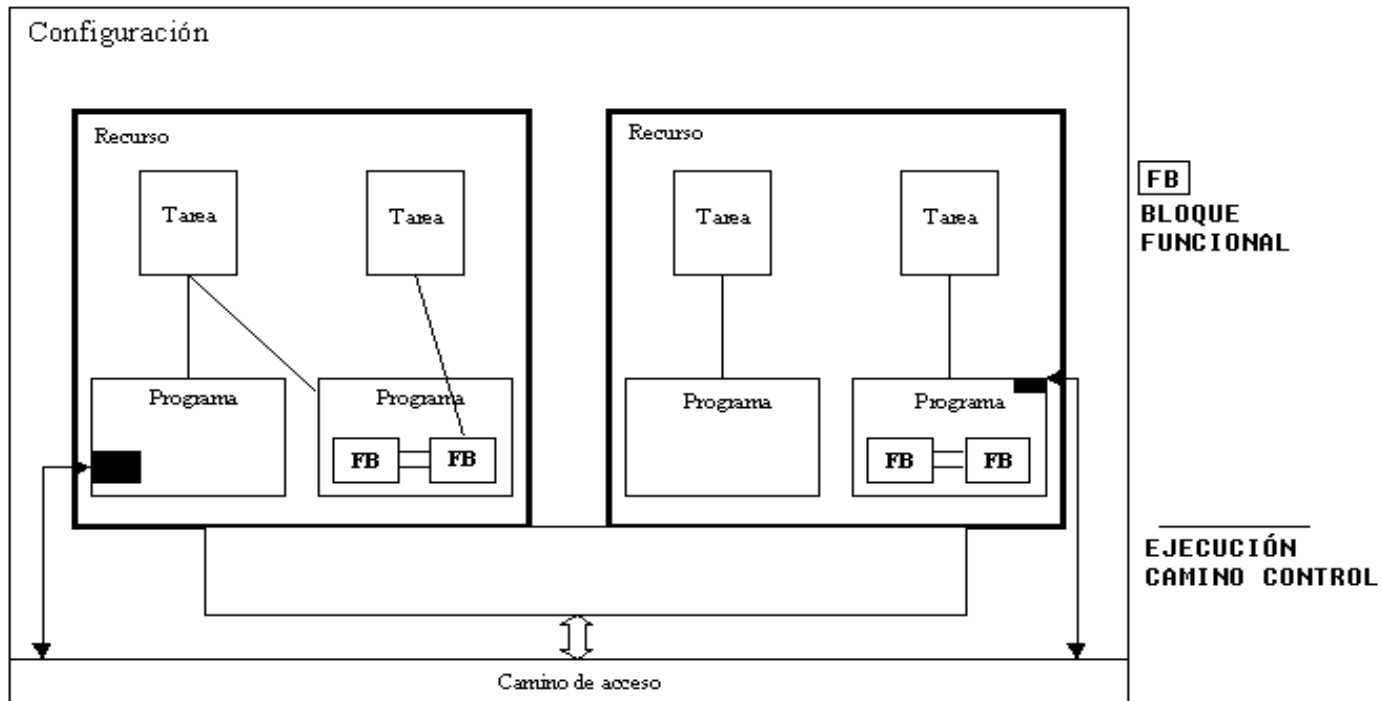
En IEC 61131-3 los Programas, Bloques de Función y Funciones se denominan Unidades de Organización de Programa (program organization units o POU).

- Funciones
 - Funciones estándar: ADD, ABS, SQRT, SIN, COS...
 - Funciones definidas por el usuario
- Bloques Funcionales (Function Blocks FBs): Los bloques funcionales son los equivalentes de los circuitos integrados
- Programas

Lenguajes de programación

Elementos Comunes

- **Configuración, recursos y tareas**
Modelo de software multiprocesamiento:

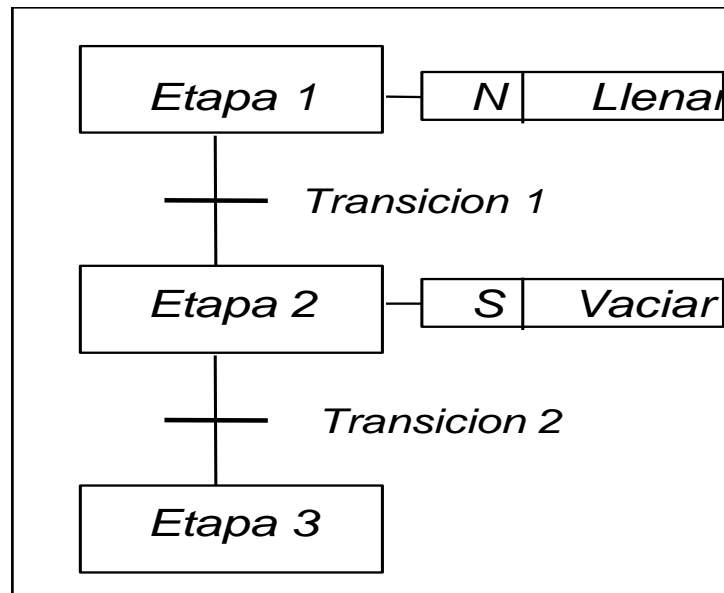


Lenguajes de programación

Elementos Comunes

- **Grafica de Secuencia de Funciones (Sequential Function Chart - SFC)**

SFC describe gráficamente el comportamiento secuencial de un programa de control. Esta definición deriva de las Redes de Petri y Grafcet (IEC 848)



Lenguajes de programación

Se definen cuatro lenguajes de programación normalizados. Esto significa que su sintaxis y semántica ha sido definida, no permitiendo particularidades distintivas (dialectos). Una vez aprendidos se podrá usar una amplia variedad de sistemas basados en esta norma.

- **Lenguajes textuales**

- Lista de Instrucciones (Instruction List – IL)
- Texto estructurado (Structured Text – ST)

- **Lenguajes gráficos**

- Diagrama de Contactos (Ladder Diagram – LD)
- Diagrama de Bloques de Funciones (Function Block Diagram – FBD)

La elección del lenguaje de programación depende de los conocimientos del programador, el problema a tratar, el nivel de descripción del proceso, la estructura del sistema de control o la coordinación con otras personas o departamentos.

Lenguajes de programación

LISTA DE INSTRUCCIONES

Lista de Instrucciones (IL) es el modelo de lenguaje ensamblador basado un acumulador simple; procede del alemán “Anweisungsliste”, AWL.

Instruction List - IL

LD	A
ANDN	B
ST	C

Lenguajes de programación

TEXTO ESTRUCTURADO

El lenguaje *Texto estructurado* (ST) es un lenguaje de alto nivel con orígenes en el Ada, Pascal y 'C'; puede ser utilizado para codificar expresiones complejas e instrucciones anidadas; este lenguaje dispone de estructuras para bucles (REPEAT-UNTIL; WHILE-DO), ejecución condicional (IF-THEN-ELSE; CASE), funciones (SQRT, SIN, etc.).

Structured Text - ST

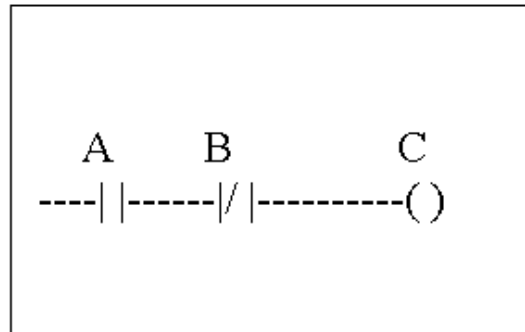
```
C = A AND NOT B
```


Lenguajes de programación

DIAGRAMA DE CONTACTOS

El Diagrama de contactos (LD) tiene sus orígenes en los Estados Unidos. Está basado en la presentación gráfica de la lógica de relés.

Ladder Diagram - LD

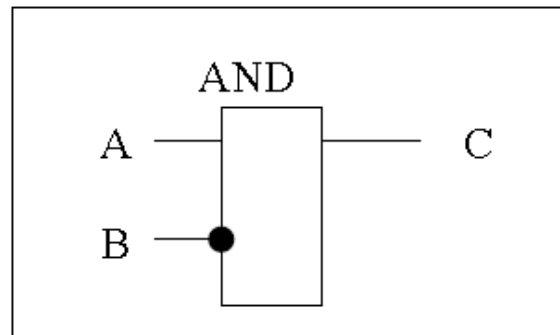


Lenguajes de programación

DIAGRAMA DE BLOQUES FUNCIONALES

El *Diagramas de Bloques Funcionales* (FBD) es muy común en aplicaciones que implican flujo de información o datos entre componentes de control. Las funciones y bloques funcionales aparecen como circuitos integrados y es ampliamente utilizado en Europa.

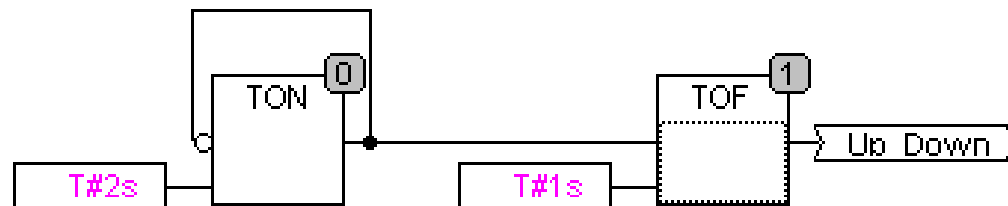
Function Block Diagram - FBD



Lenguajes de programación

GRÁFICO DE FUNCIÓN CONTINUA

CoDeSys soporta todos los lenguajes mencionados por la norma IEC-61131, además hay disponible, basado en el Diagrama de Bloques Funcionales, el Gráfico de Función Continua (CFC). Sin embargo, no funciona con las redes, sino más bien con elementos que se pueden situar libremente. Esto permite la retroalimentación, por ejemplo.



Índice

Introducción

Estándar IEC 61131-3

Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

Índice

Introducción

Estándar IEC 61131-3

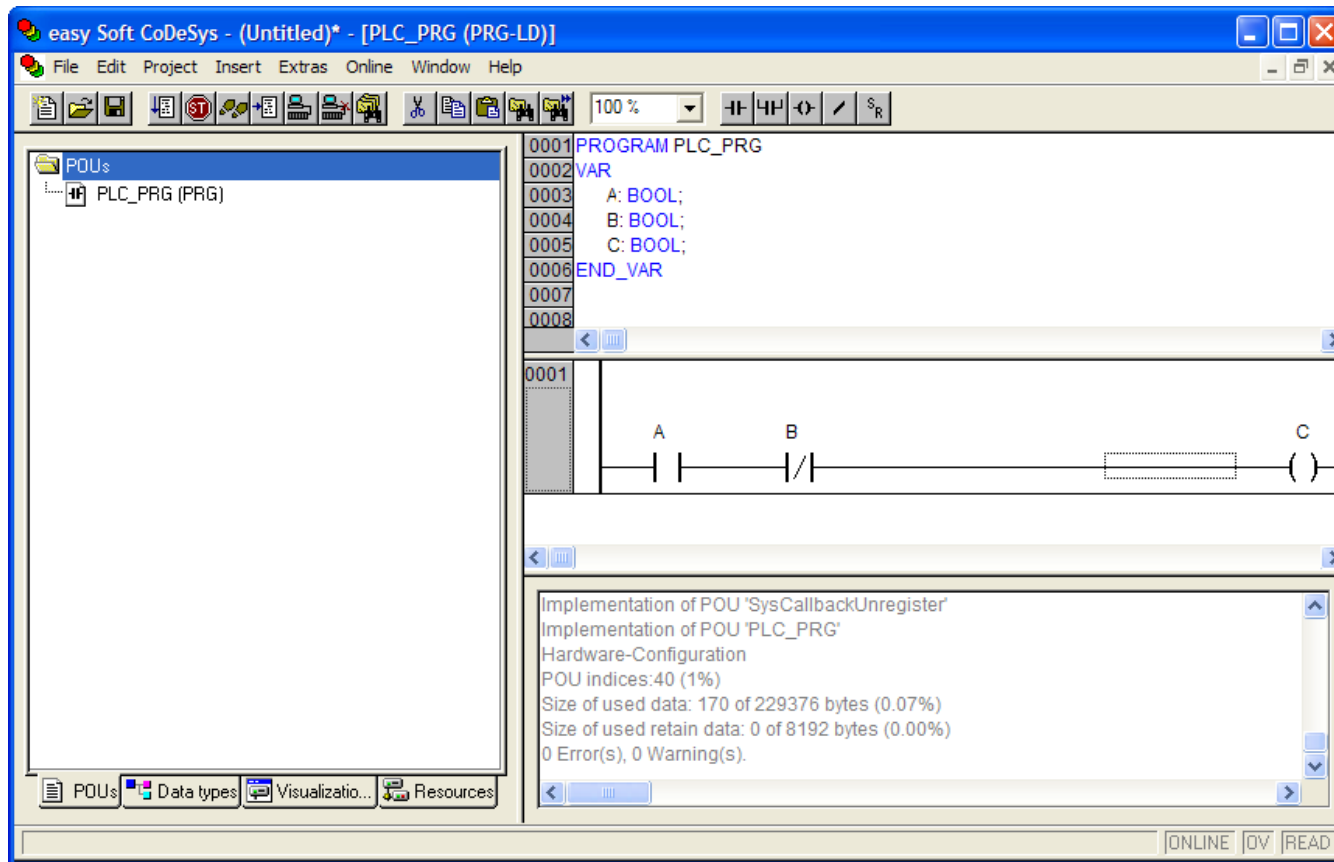
Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

CoDeSys



CoDeSys

Barra de menú

The screenshot displays the 'easy Soft CoDeSys' application window. The title bar reads 'easy Soft CoDeSys - (Untitled)* - [PLC_PRG (PRG-LD)]'. The menu bar includes 'File', 'Edit', 'Project', 'Insert', 'Extras', 'Online', 'Window', and 'Help'. The toolbar contains various icons for file operations, editing, and execution. The main workspace is divided into three sections:

- Left Panel (Project Explorer):** Shows a tree view with 'POUs' and a sub-entry 'PLC_PRG (PRG)'.
- Top Right Panel (Text Editor):** Contains the following code:

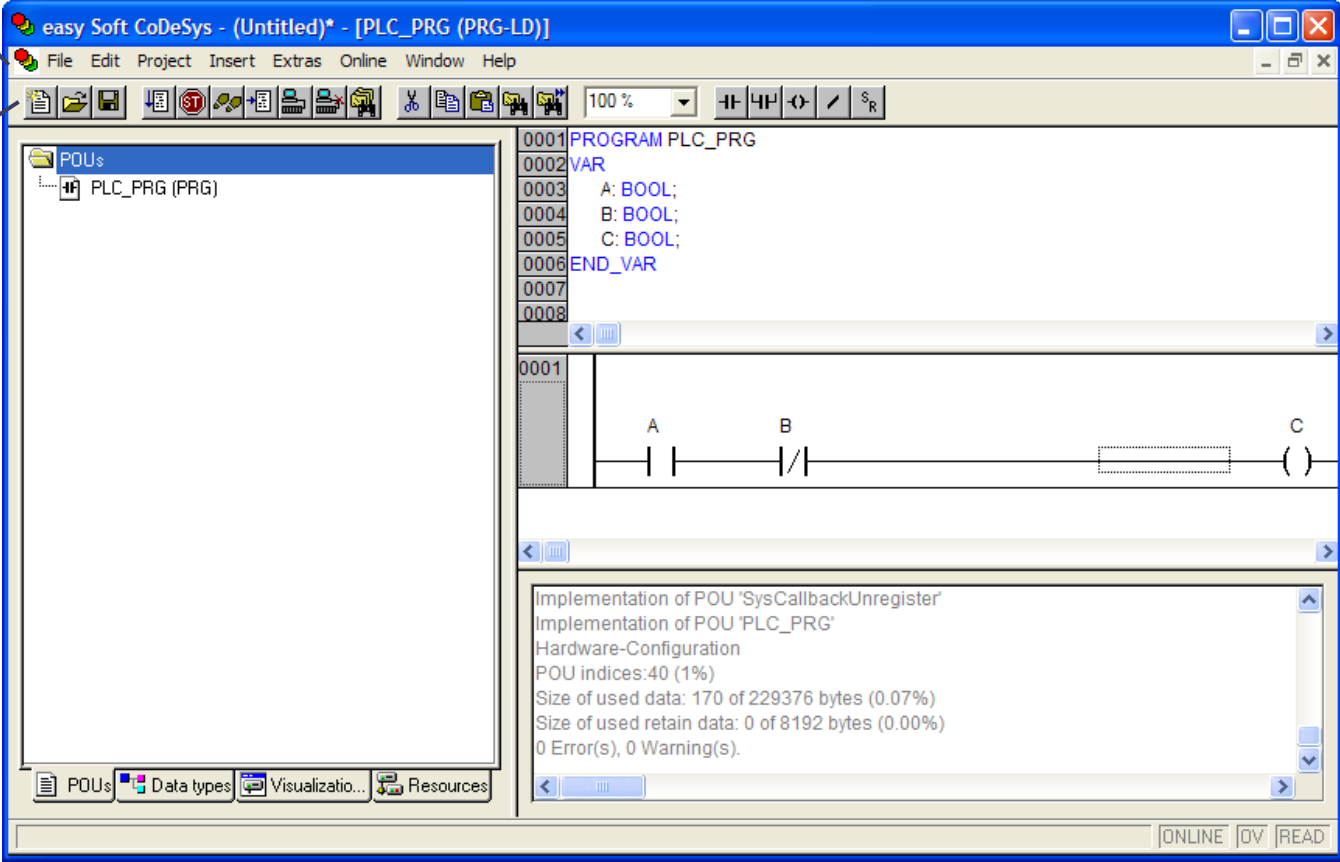
```
0001 PROGRAM PLC_PRG
0002 VAR
0003   A: BOOL;
0004   B: BOOL;
0005   C: BOOL;
0006 END_VAR
0007
0008
```
- Bottom Right Panel (Ladder Logic):** Shows a single rungs ladder logic diagram for step 0001. It features three normally open contacts labeled 'A' and 'B' in series, followed by a normally closed contact labeled 'C', and a coil symbol '()' at the end of the rung.

At the bottom of the window, there is a status bar with 'ONLINE', 'OV', and 'READ' indicators. A bottom toolbar includes icons for 'POUs', 'Data types', 'Visualizatio...', and 'Resources'.

CoDeSys

Barra de menú

Barra de iconos

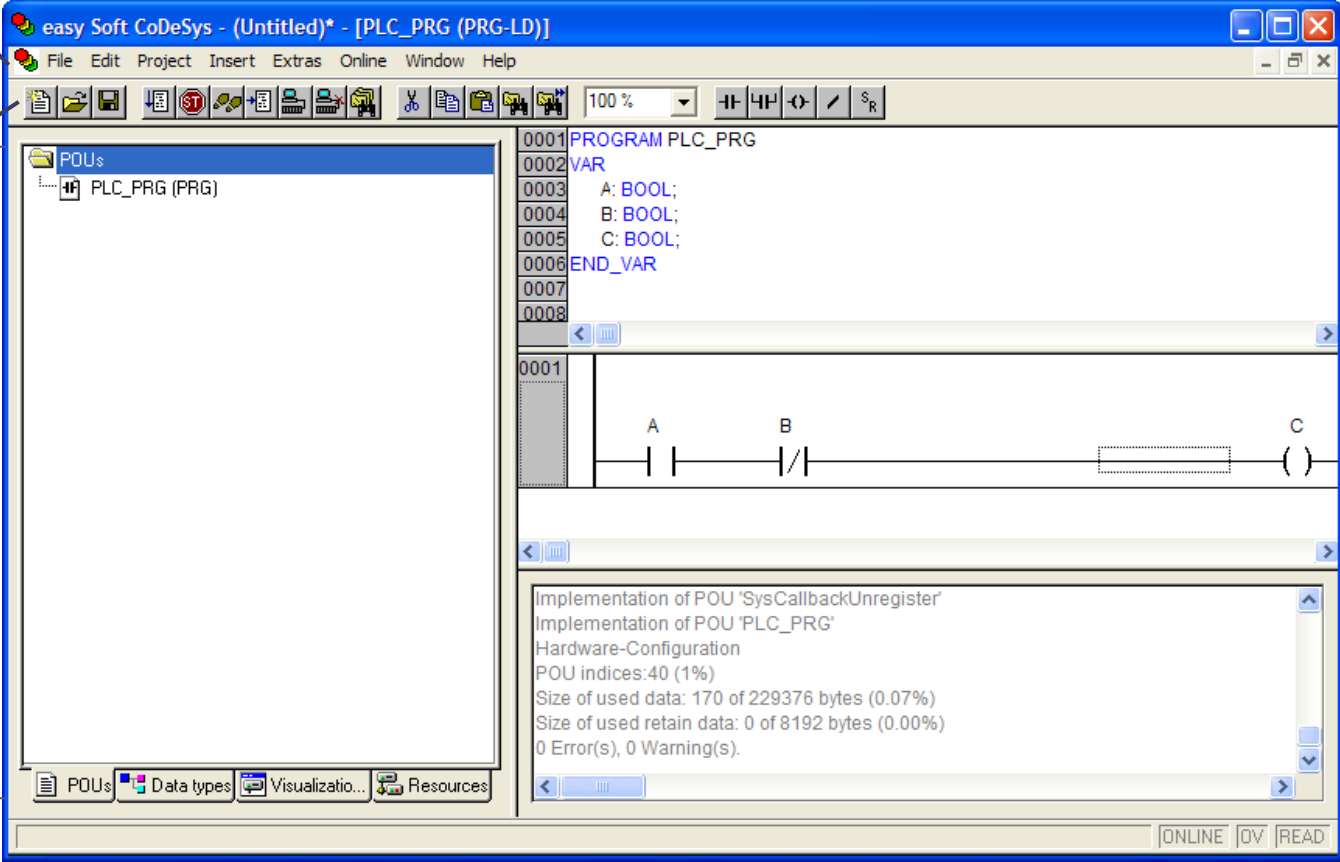


CoDeSys

Barra de menú

Barra de iconos

Árbol del proyecto



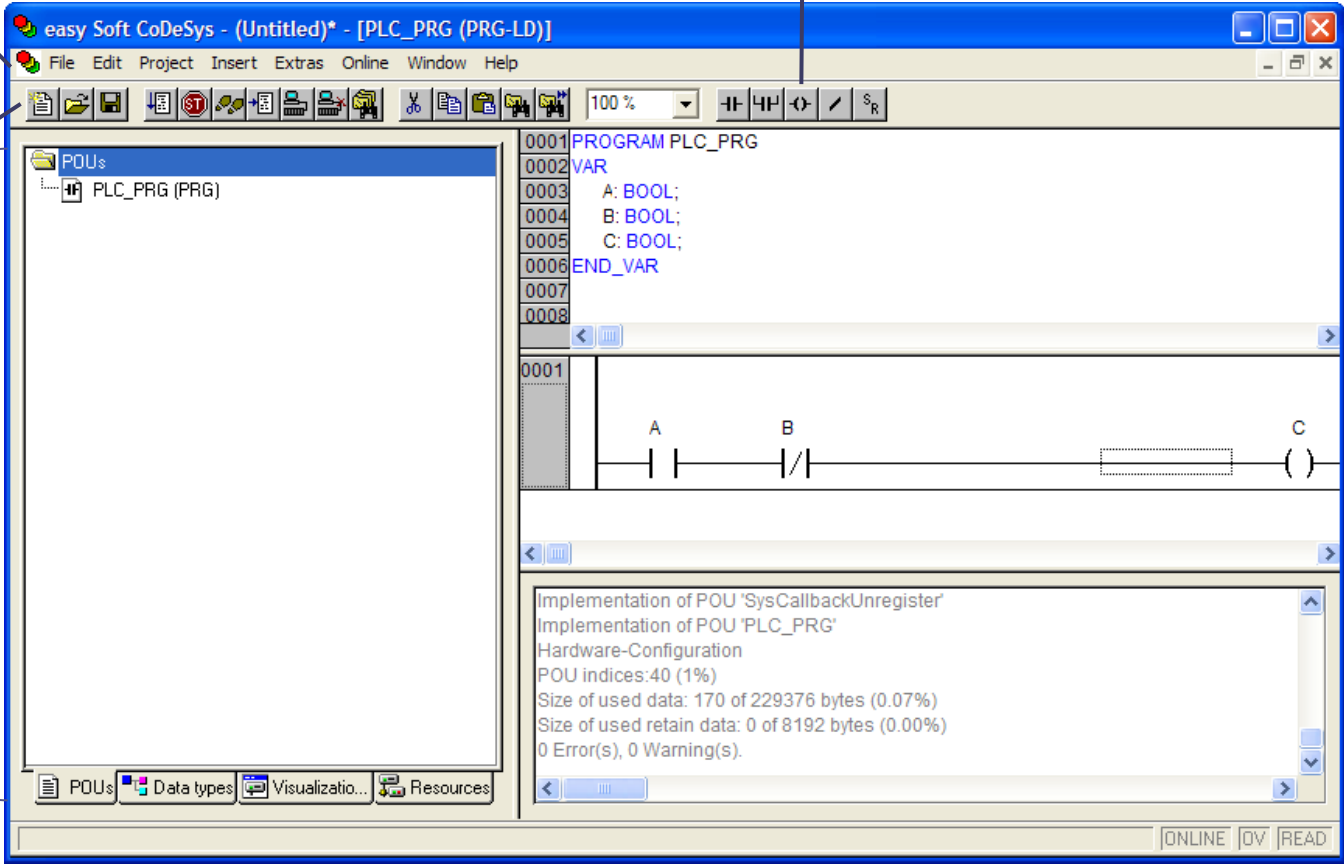
CoDeSys

Barra de menú

Barra de iconos

Árbol del proyecto

Barra de instrucciones



CoDeSys

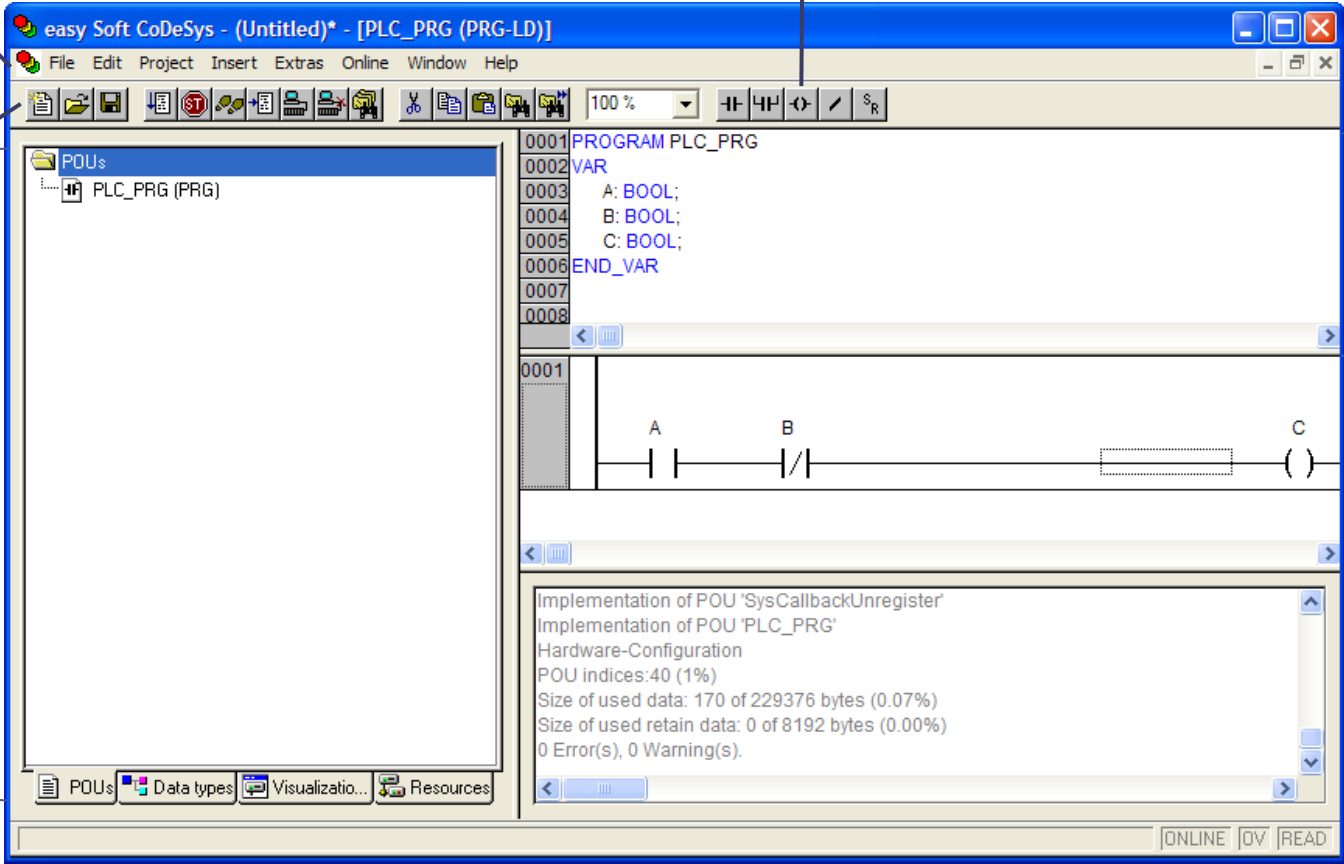
Barra de menú

Barra de iconos

Árbol del proyecto

Barra de instrucciones

Declaración de variables



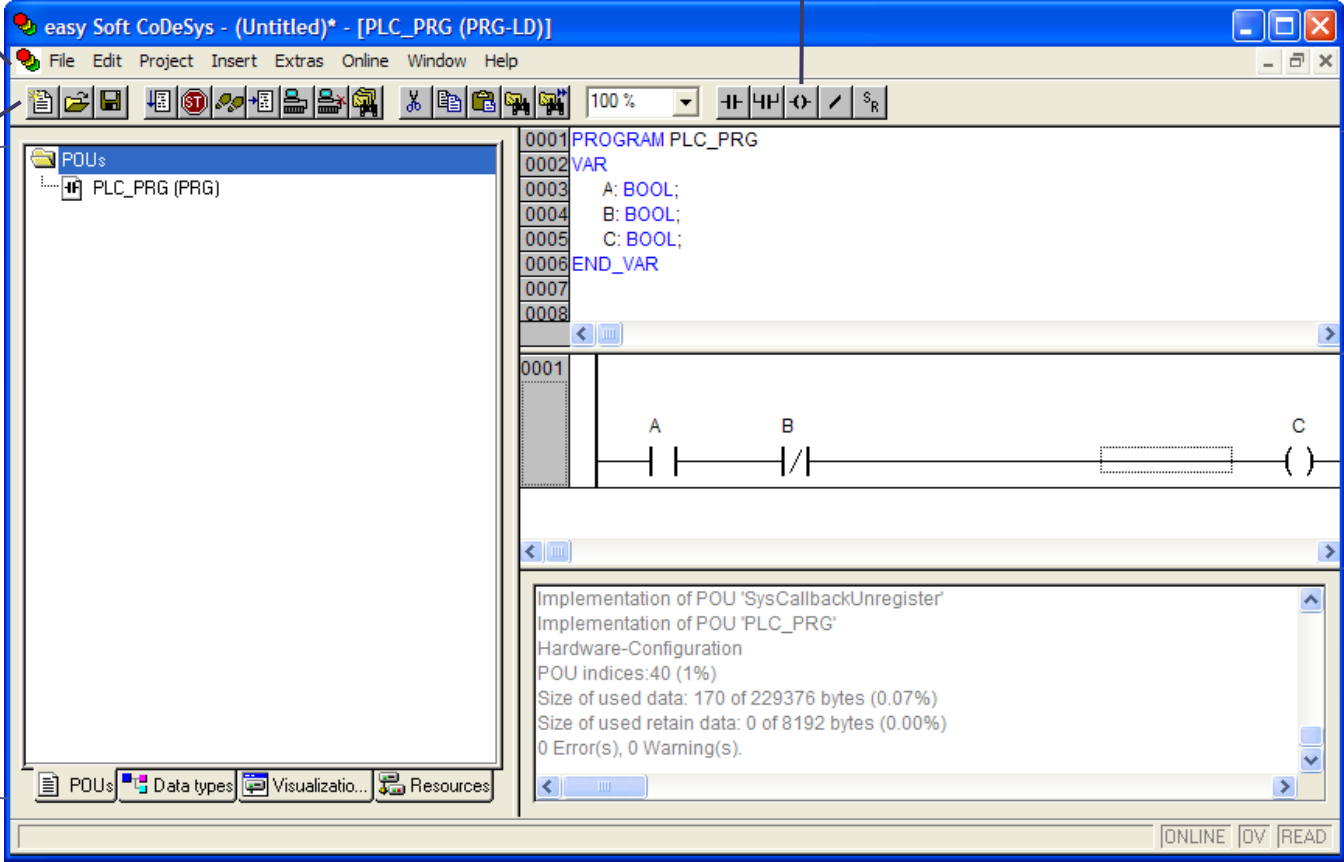
CoDeSys

Barra de menú

Barra de iconos

Árbol del proyecto

Barra de instrucciones



Declaración de variables

Editor del programa

CoDeSys

Barra de menú

Barra de iconos

Árbol del proyecto

Barra de instrucciones

The screenshot shows the CoDeSys software interface with the following components:

- Project Tree (Left):** Shows a tree structure with 'POUs' and a sub-entry 'PLC_PRG (PRG)'.
- Code Editor (Top Right):** Contains the following text:

```
0001 PROGRAM PLC_PRG
0002 VAR
0003   A: BOOL;
0004   B: BOOL;
0005   C: BOOL;
0006 END_VAR
0007
0008
```
- Ladder Logic Editor (Middle Right):** Shows a single rung with three normally open contacts labeled 'A', 'B', and 'C' in series, followed by a coil symbol '()'.
- Results Panel (Bottom Right):** Displays the following information:

```
Implementation of POU 'SysCallbackUnregister'
Implementation of POU 'PLC_PRG'
Hardware-Configuration
POU indices:40 (1%)
Size of used data: 170 of 229376 bytes (0.07%)
Size of used retain data: 0 of 8192 bytes (0.00%)
0 Error(s), 0 Warning(s).
```
- Bottom Bar:** Includes icons for 'POUs', 'Data types', 'Visualizatio...', and 'Resources', along with status indicators for 'ONLINE', 'OV', and 'READ'.

Declaración de variables

Editor del programa

Panel de resultados

CoDeSys

Barra de menú

Barra de iconos

Árbol del proyecto

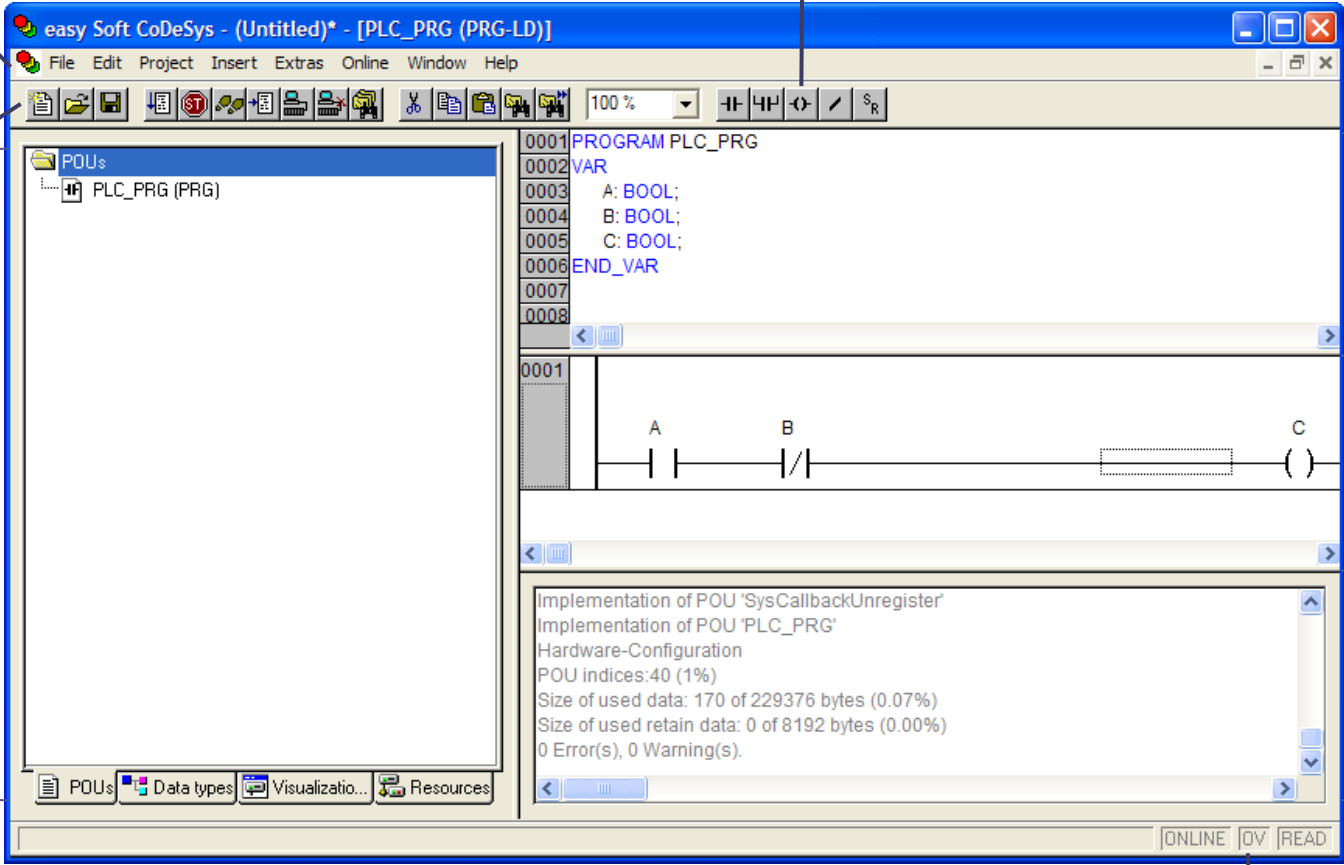
Barra de instrucciones

Declaración de variables

Editor del programa

Panel de resultados

Barra de estado

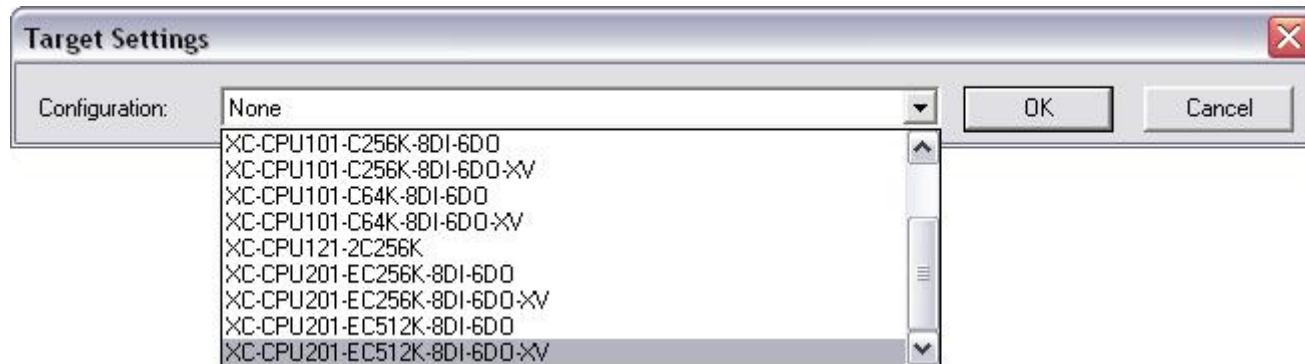


CoDeSys

La versión suministrada con el autómata programable MOELLER XC201 es easy Soft CoDeSys versión 2.3.5.8.

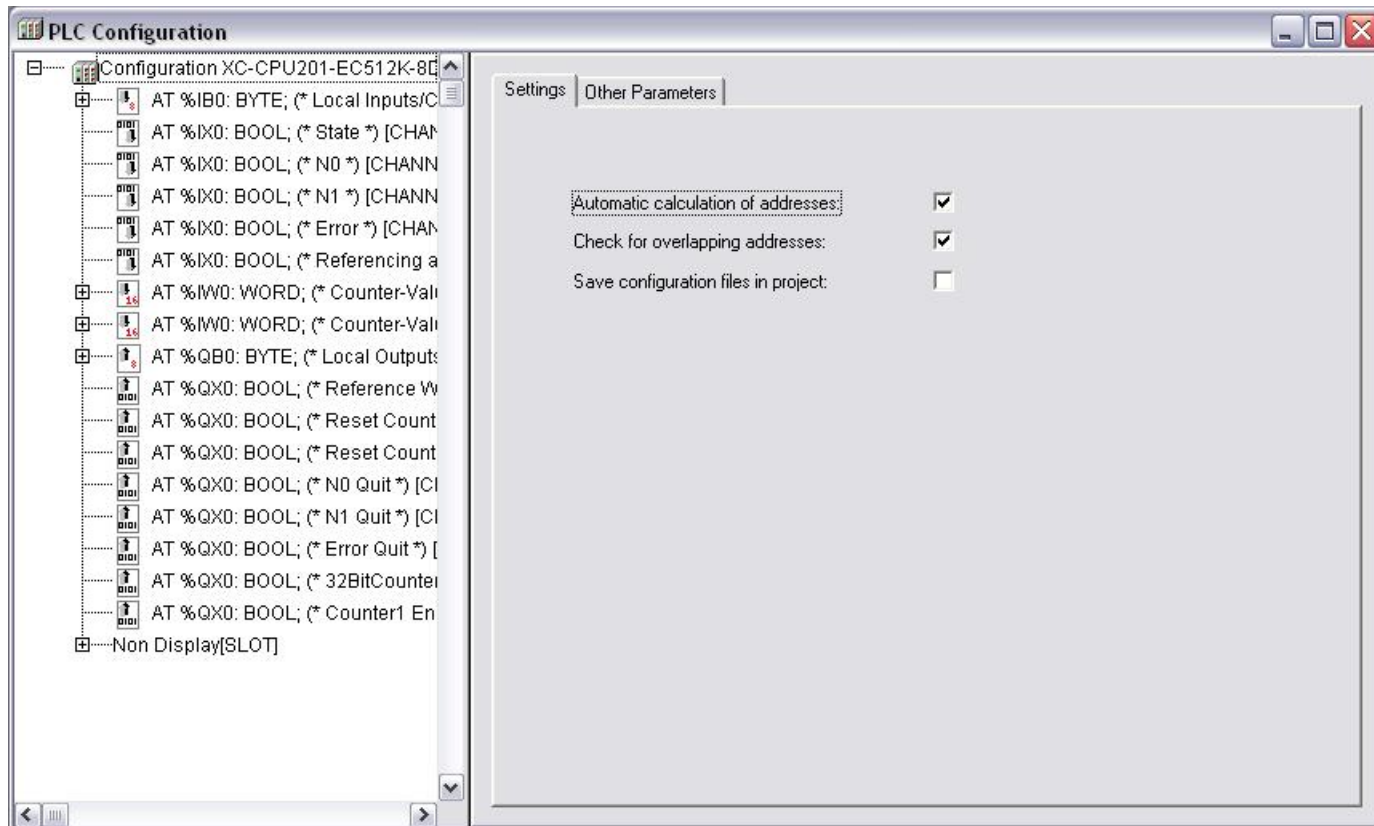
Pasos para la configuración del autómata y las comunicaciones:

- Selección del modelo de autómata (al inicio de un nuevo proyecto).

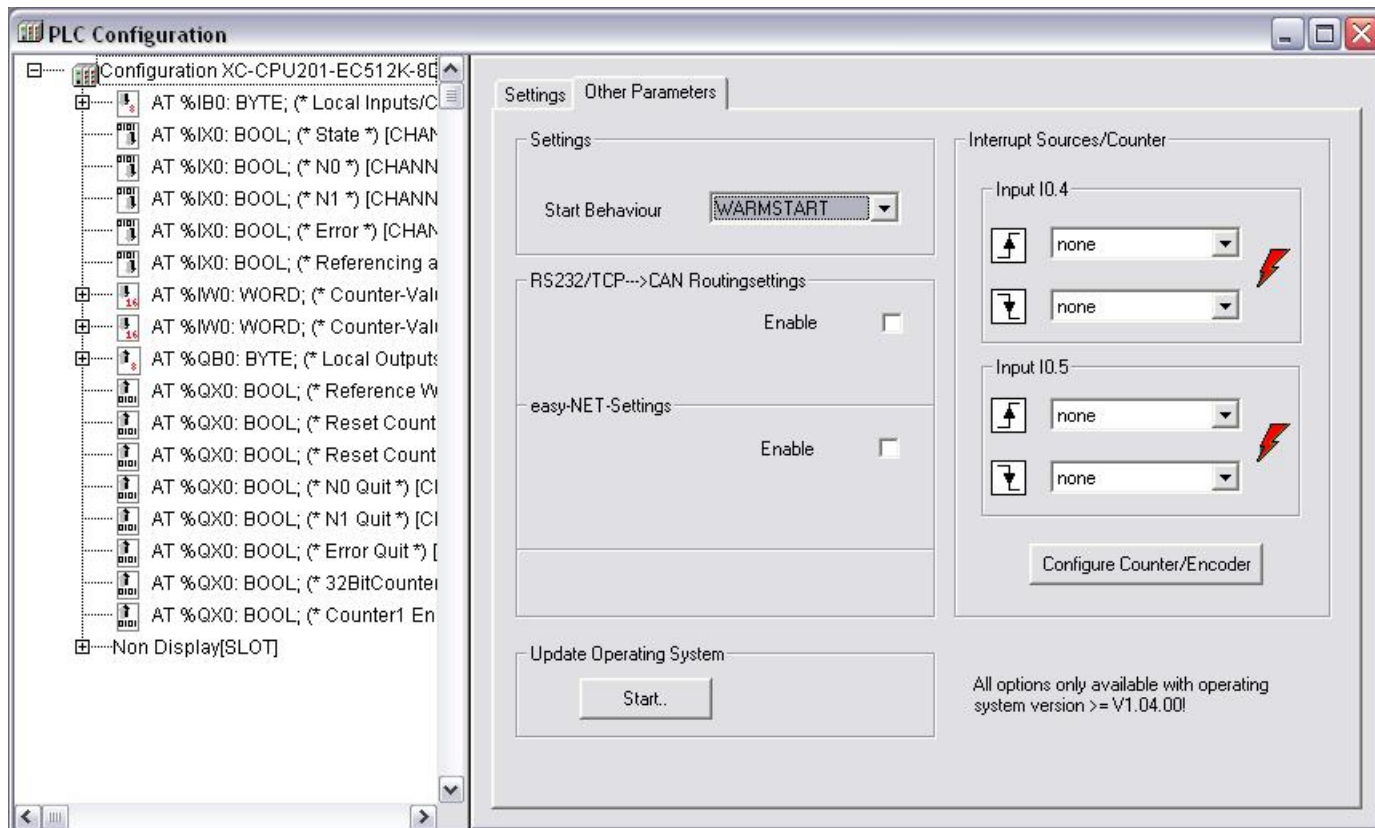


CoDeSys

- Configuración del autómata (Resources > PLC Configuration).

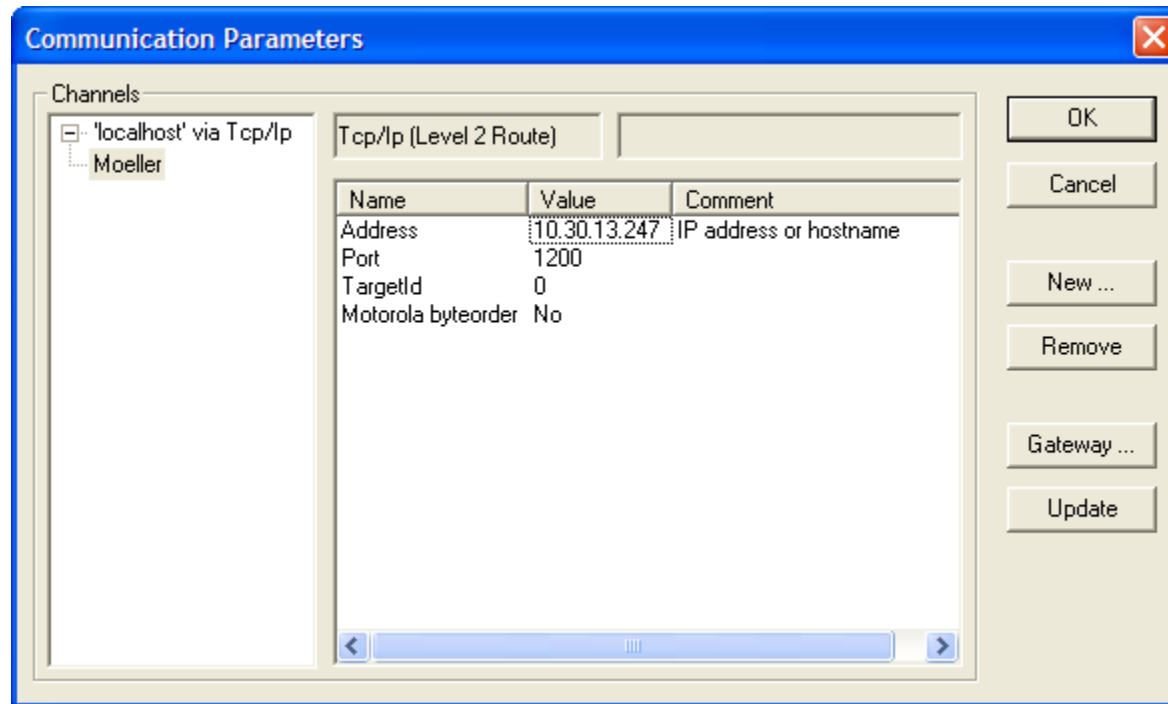


CoDeSys



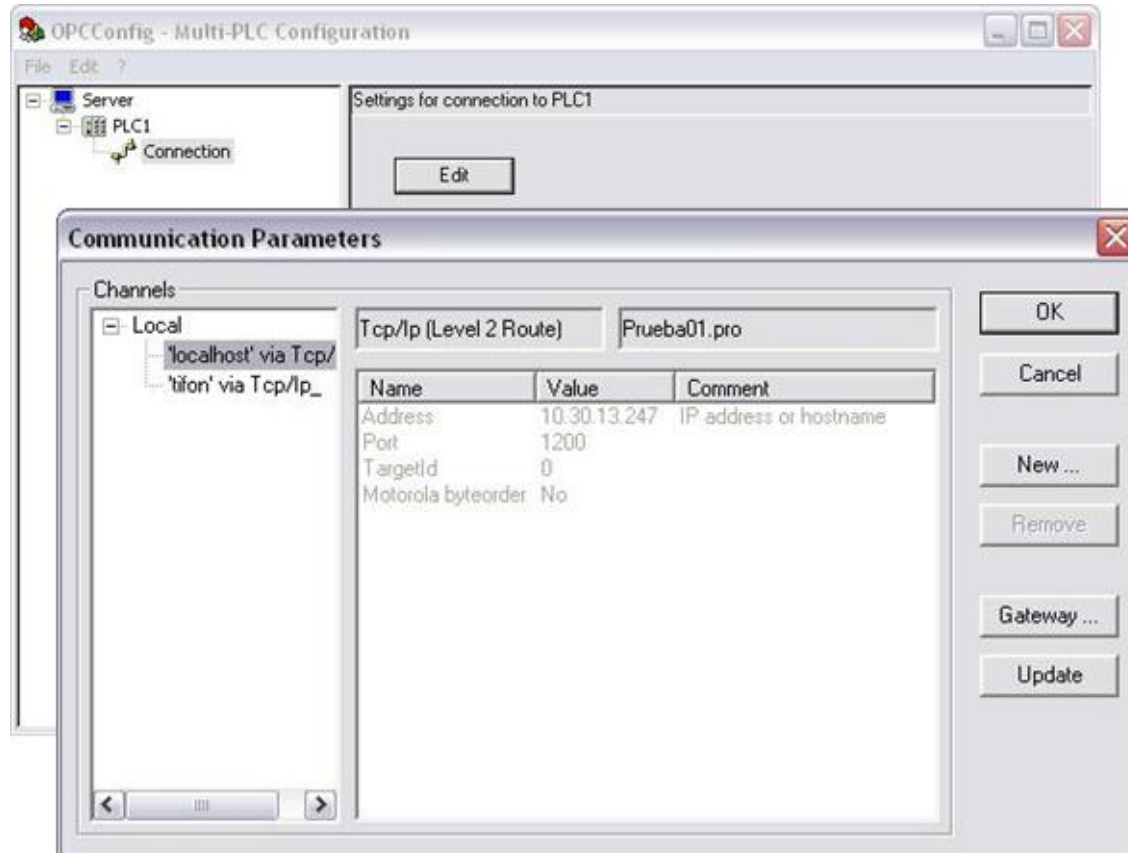
CoDeSys

- Configuración de las comunicaciones (Online > Communication Parameters).



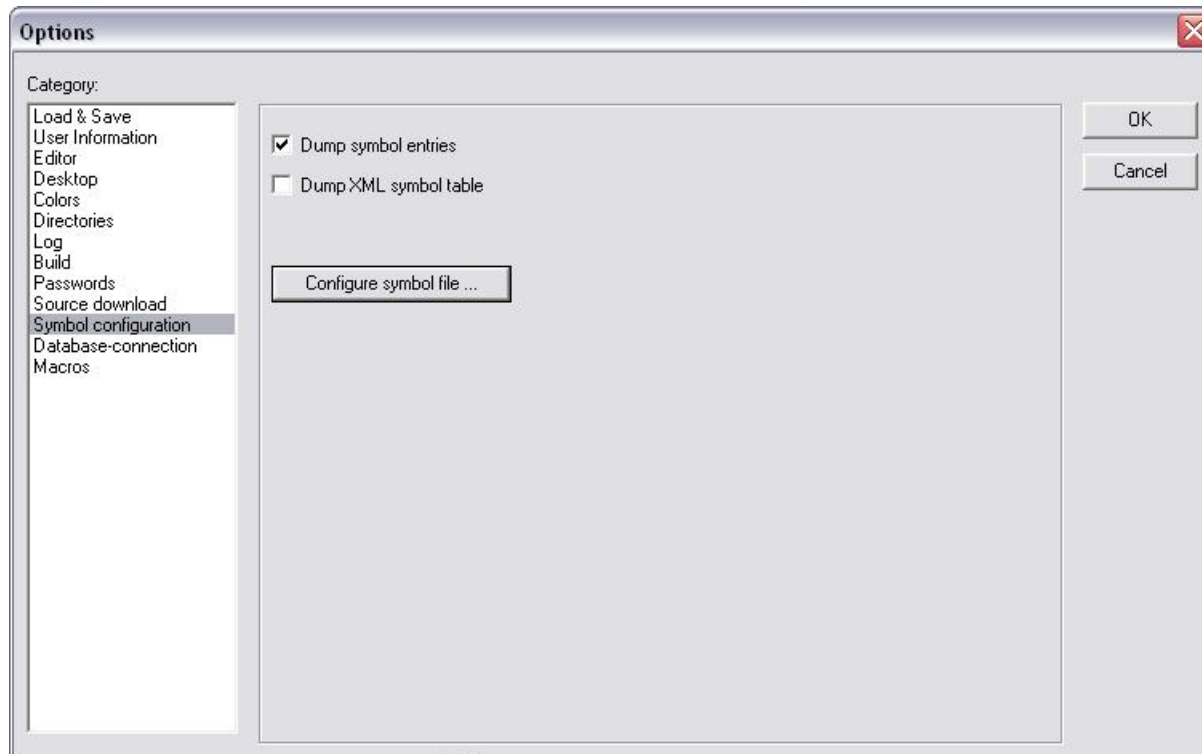
CoDeSys

- Configuración del servidor OPC (OPC Configurator).

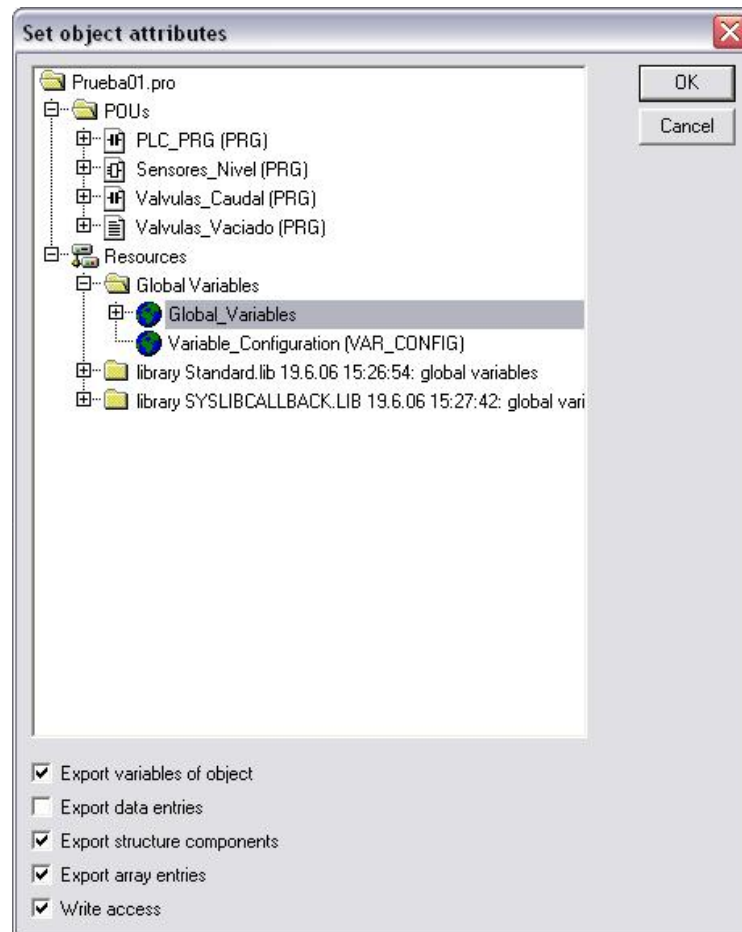


CoDeSys

- Selección de variables OPC (Project > Options > Symbol configuration).

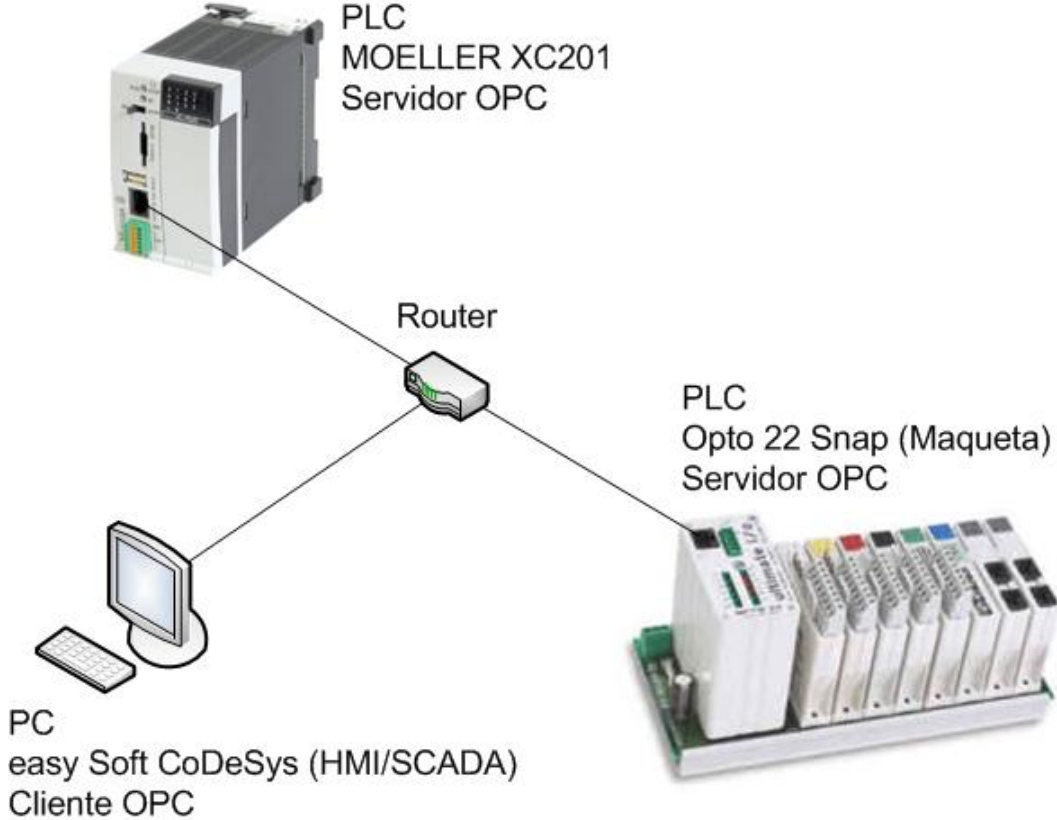


CoDeSys



CoDeSys

Comunicaciones



Índice

Introducción

Estándar IEC 61131-3

Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

Índice

Introducción

Estándar IEC 61131-3

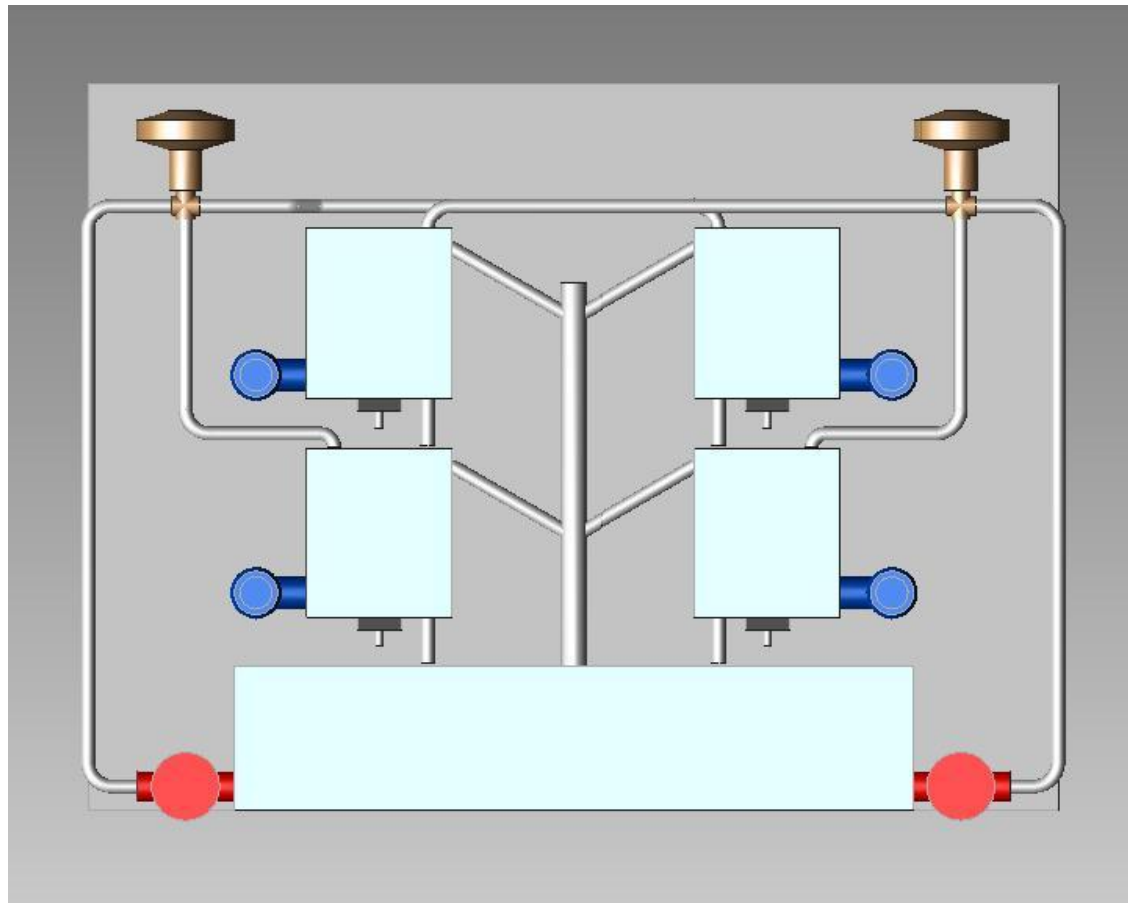
Lenguajes de programación

CoDeSys

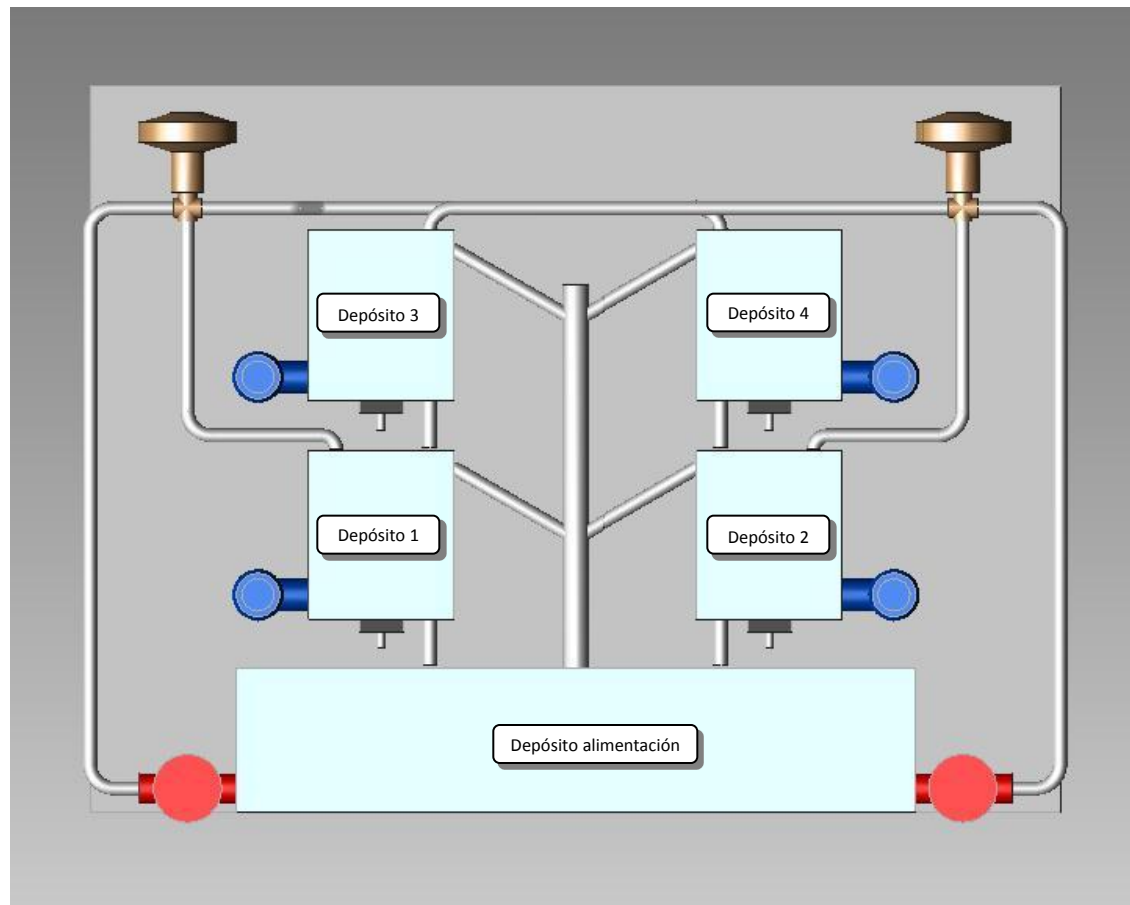
Prácticas

Conclusiones

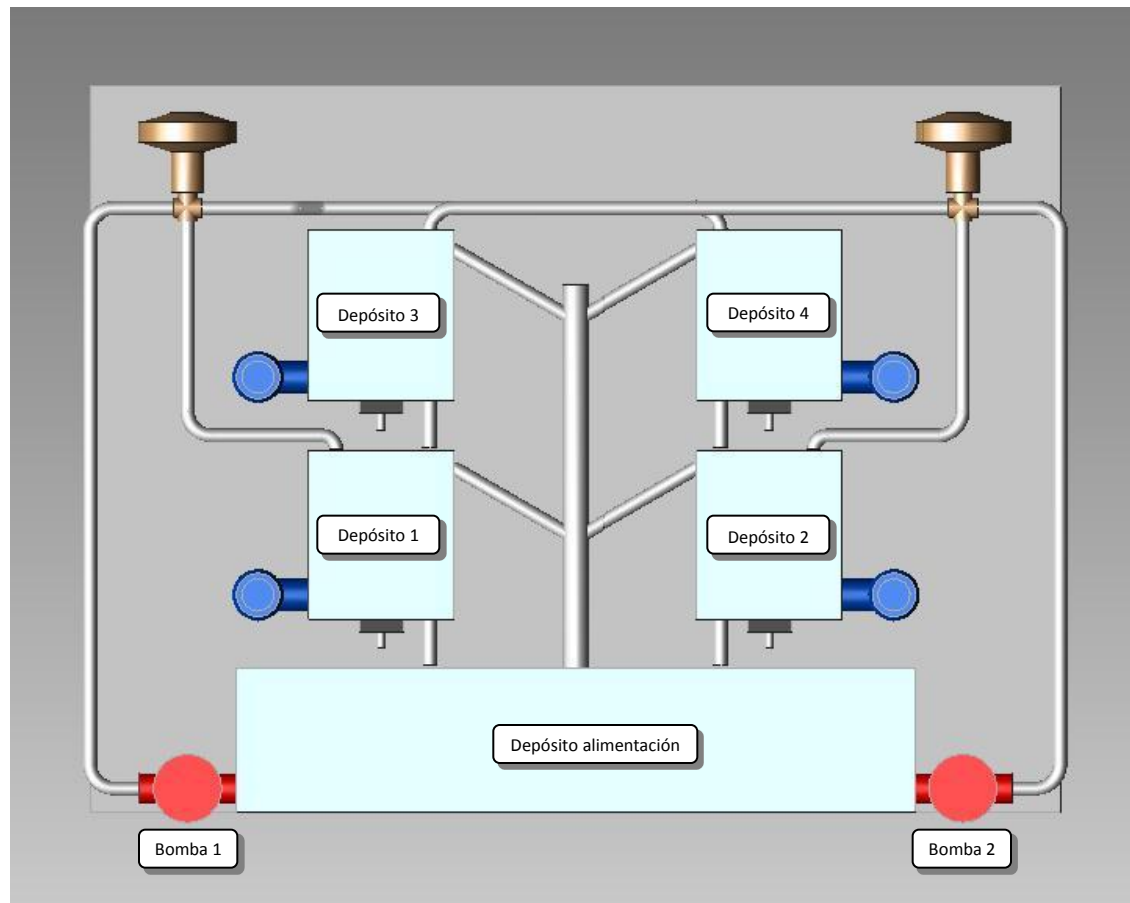
Prácticas Maqueta



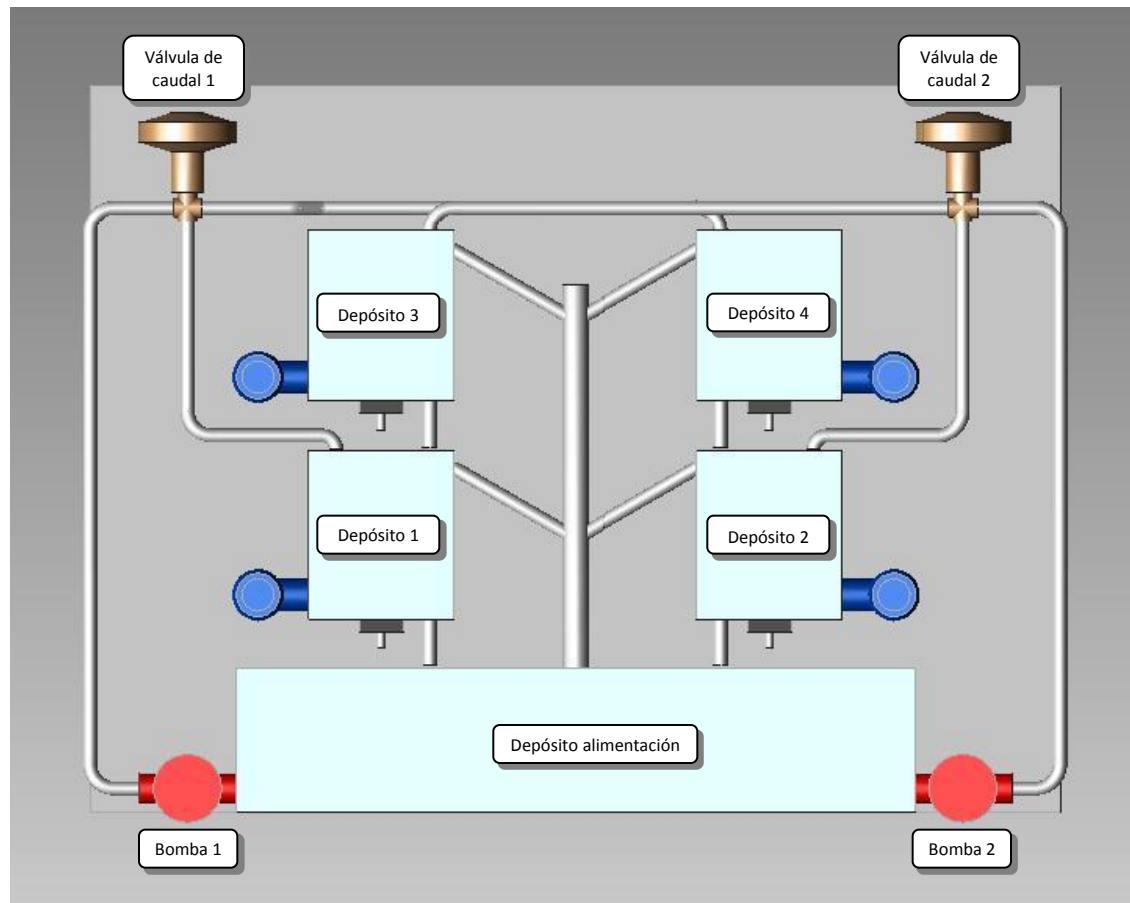
Prácticas Maqueta



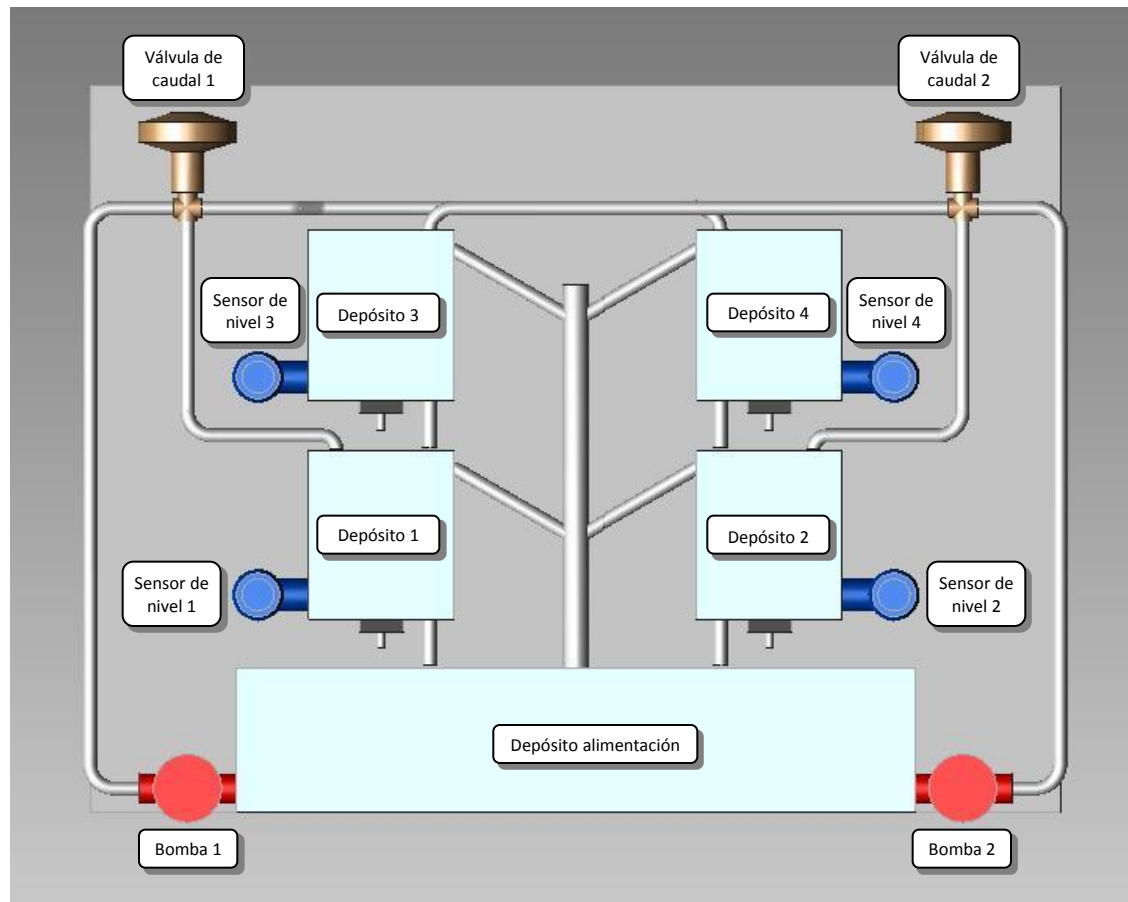
Prácticas Maqueta



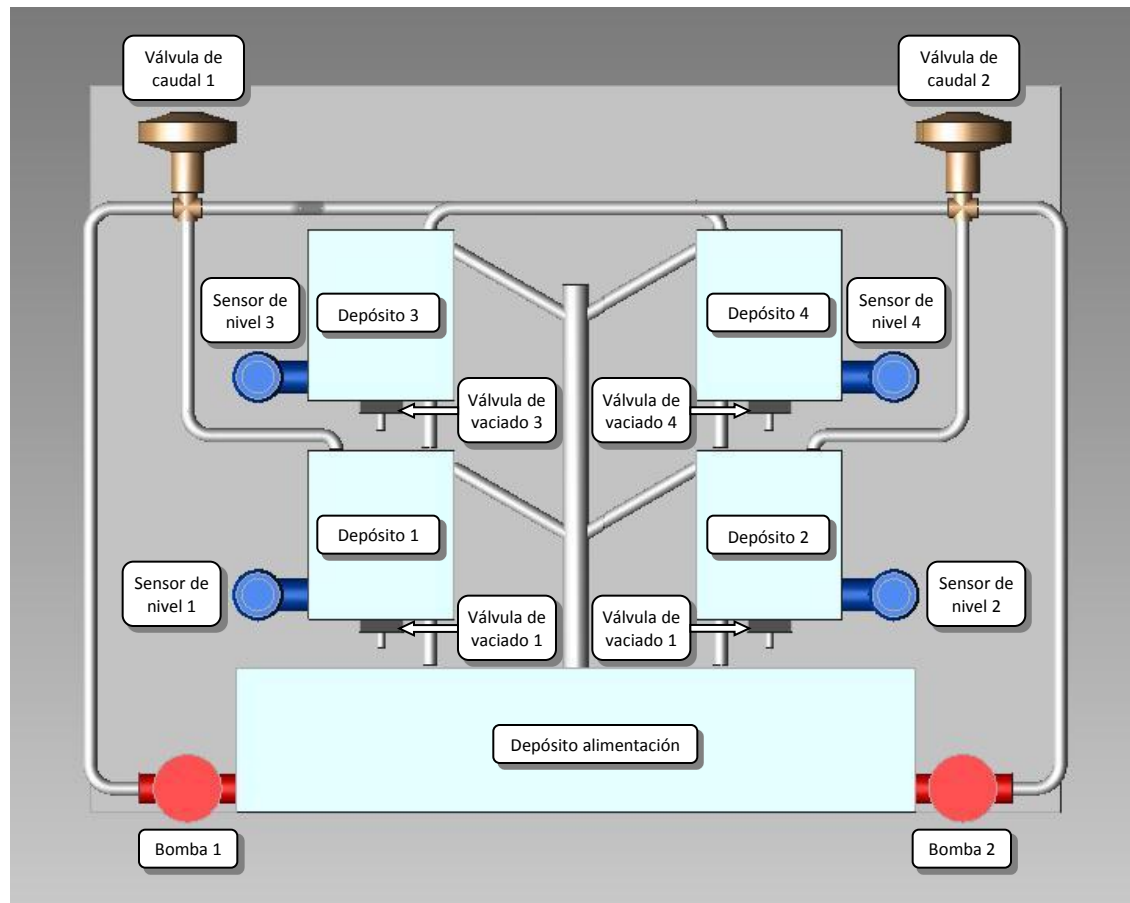
Prácticas Maqueta



Prácticas Maqueta



Prácticas Maqueta



Prácticas

Las prácticas se dividen en dos bloques dependiendo del tipo de control que se efectúa sobre los elementos de la maqueta.

- **Control en lazo abierto**

Se trata de una serie de programas, dentro de un mismo proyecto, que controlan el accionamiento de cada uno de los elementos de la maqueta por separado, sin existir interacción entre elementos.

- **Control en lazo cerrado**

Se efectúa un control automático de nivel de cada depósito, pudiendo escoger que depósito o que grupo de depósitos controlar. Todos los elementos interactúan para lograr mantener un nivel.

Prácticas

Variables Globales (Variables OPC)

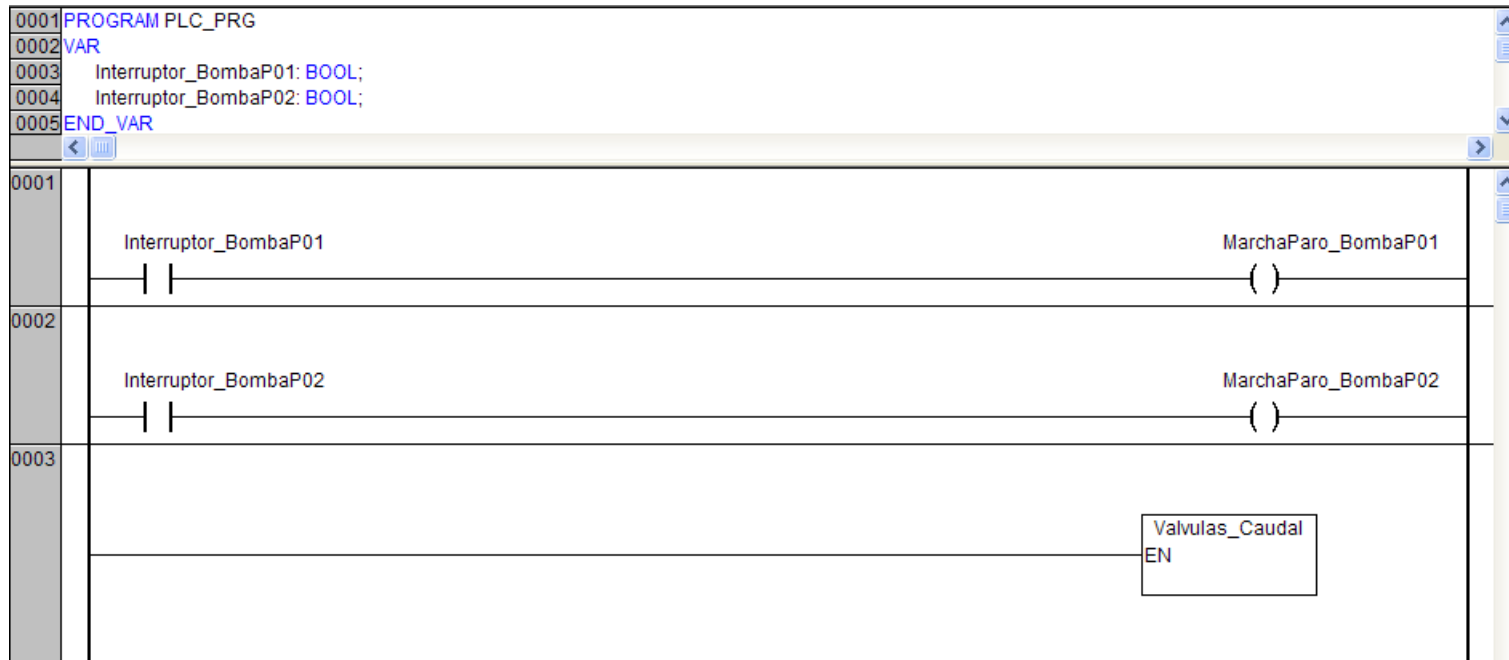
```
0001 VAR_GLOBAL
0002   MarchaParo_BombaP01:BOOL;
0003   MarchaParo_BombaP02:BOOL;
0004   Valvula_LV03:BOOL;
0005   Valvula_LV04:BOOL;
0006   Valvula_LV05:BOOL;
0007   Valvula_LV06:BOOL;
0008   Valvula_Nivel_D01D04:REAL;
0009   Valvula_Nivel_D02D03:REAL;
0010   Convertidor_BombaP01:REAL;
0011   Convertidor_BombaP02:REAL;
0012
0013   Conf_Marcha_BombaP01:BOOL;
0014   Conf_Marcha_BombaP02:BOOL;
0015   Fallo_BombaP01:BOOL;
0016   Fallo_BombaP02:BOOL;
0017   Presencia_Tension_24Vdc:BOOL;
0018   Presencia_Tension_220Vac:BOOL;
0019   Nivel_D01:REAL;
0020   Nivel_D02:REAL;
0021   Nivel_D03:REAL;
0022   Nivel_D04:REAL;
0023
0024 END_VAR
```


Prácticas

Control en lazo abierto

Programas de control

- **Bombas (LD)**



Prácticas

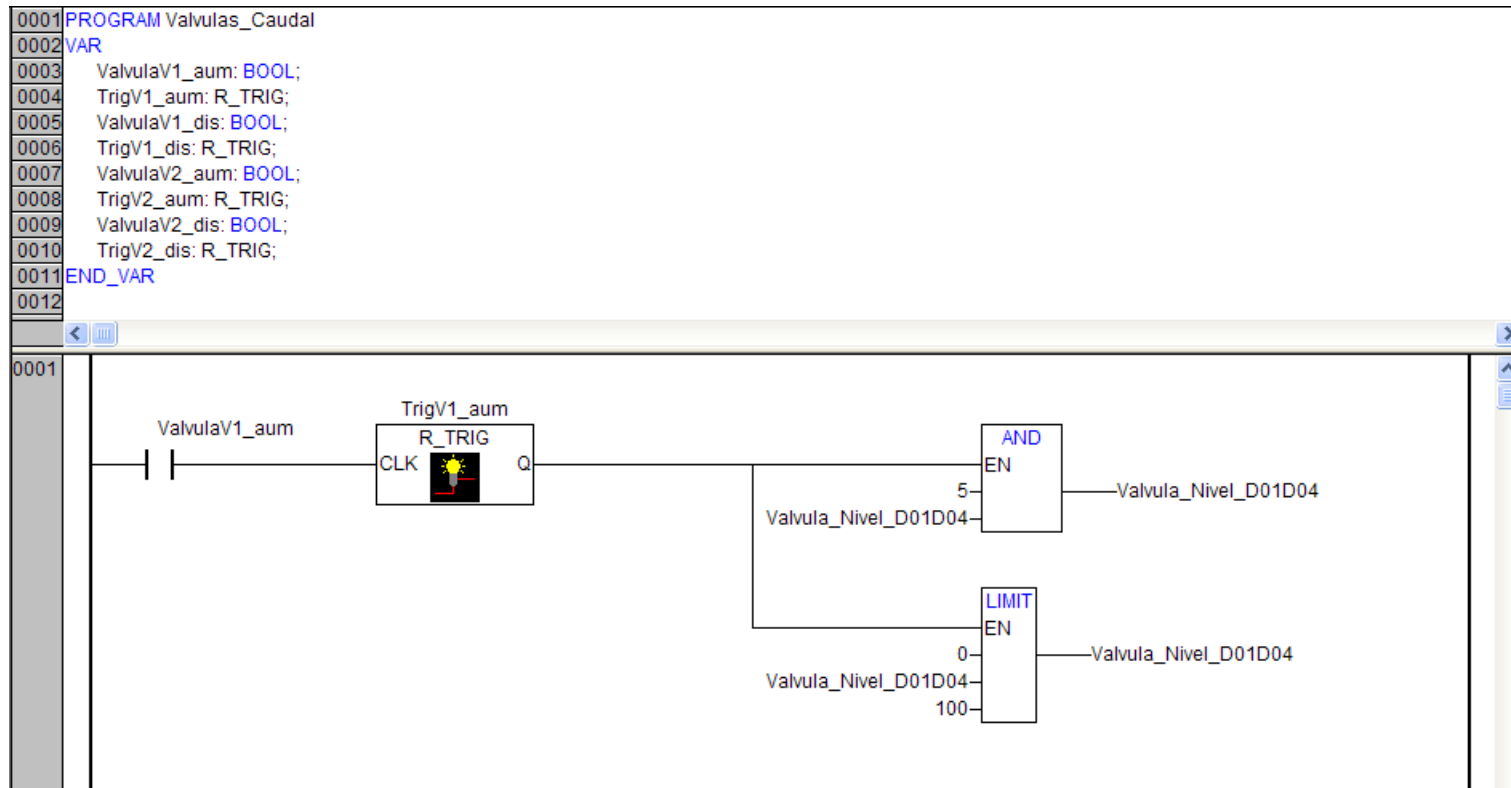
Control en lazo abierto



Prácticas

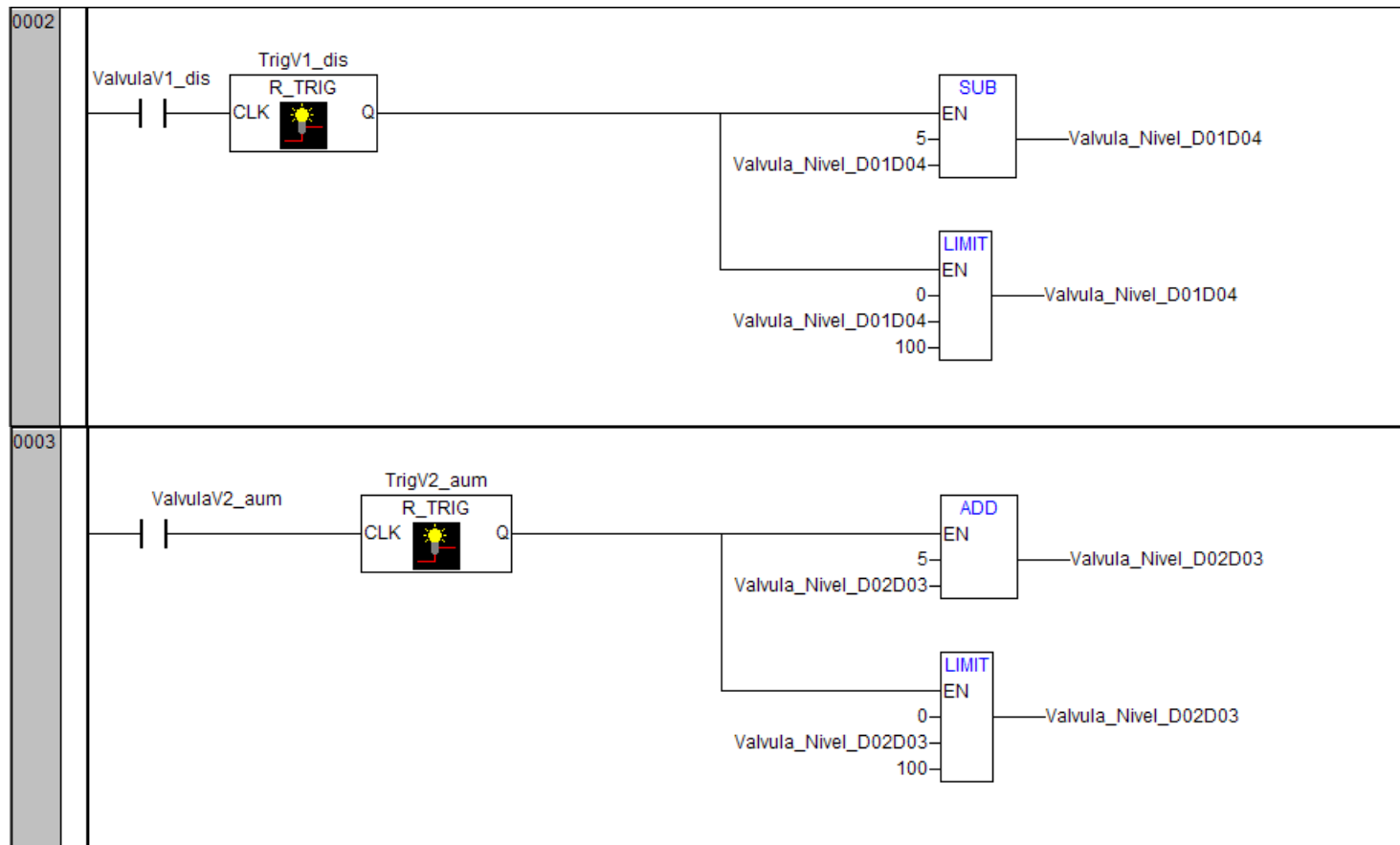
Control en lazo abierto

- Válvulas de caudal (LD)



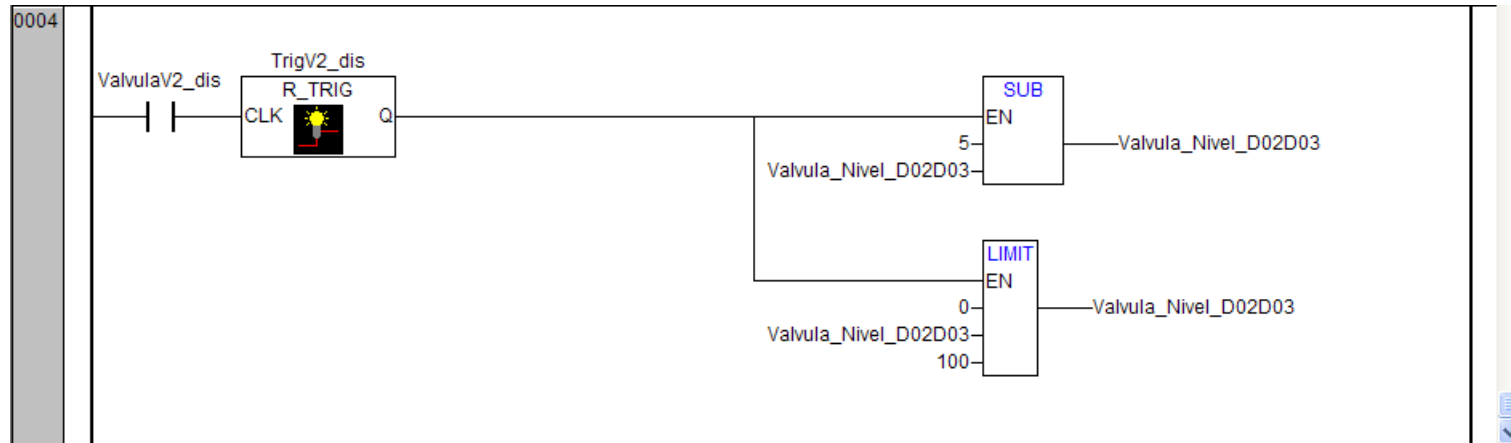
Prácticas

Control en lazo abierto



Prácticas

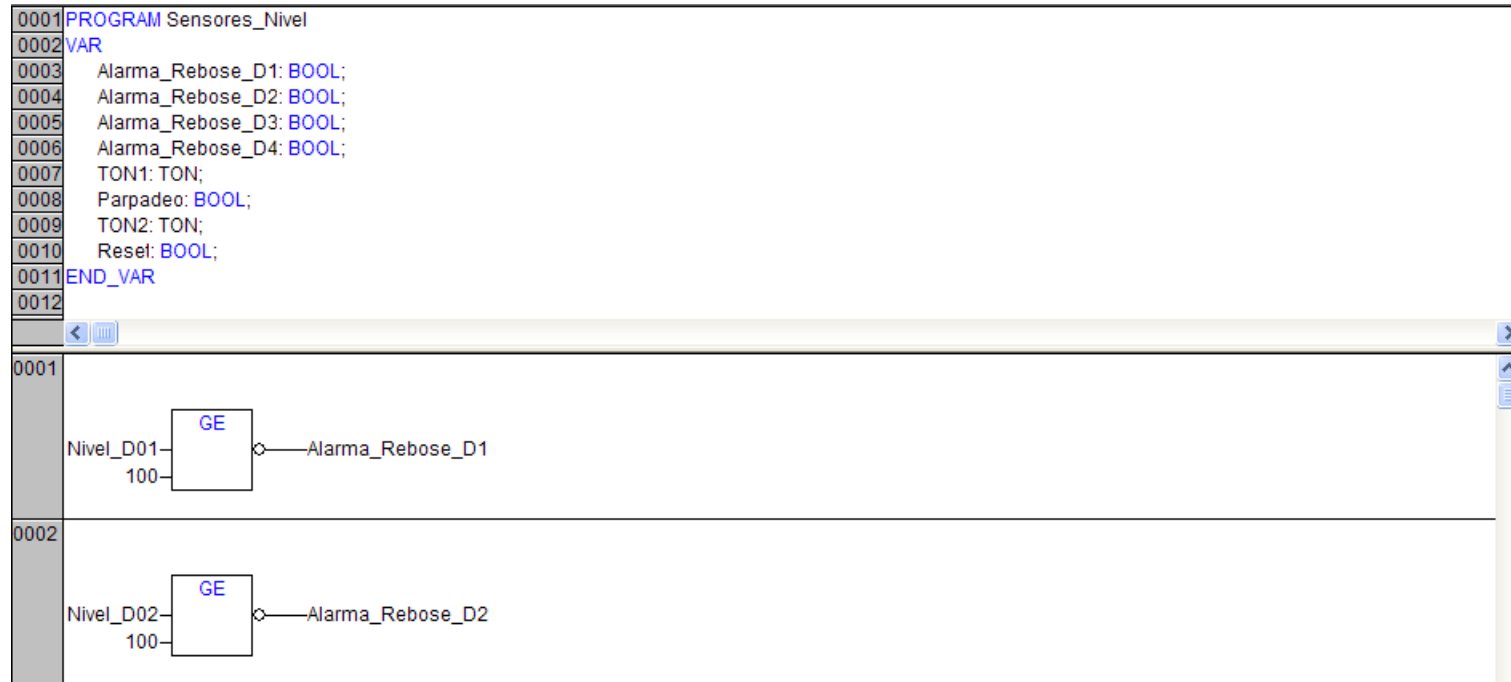
Control en lazo abierto



Prácticas

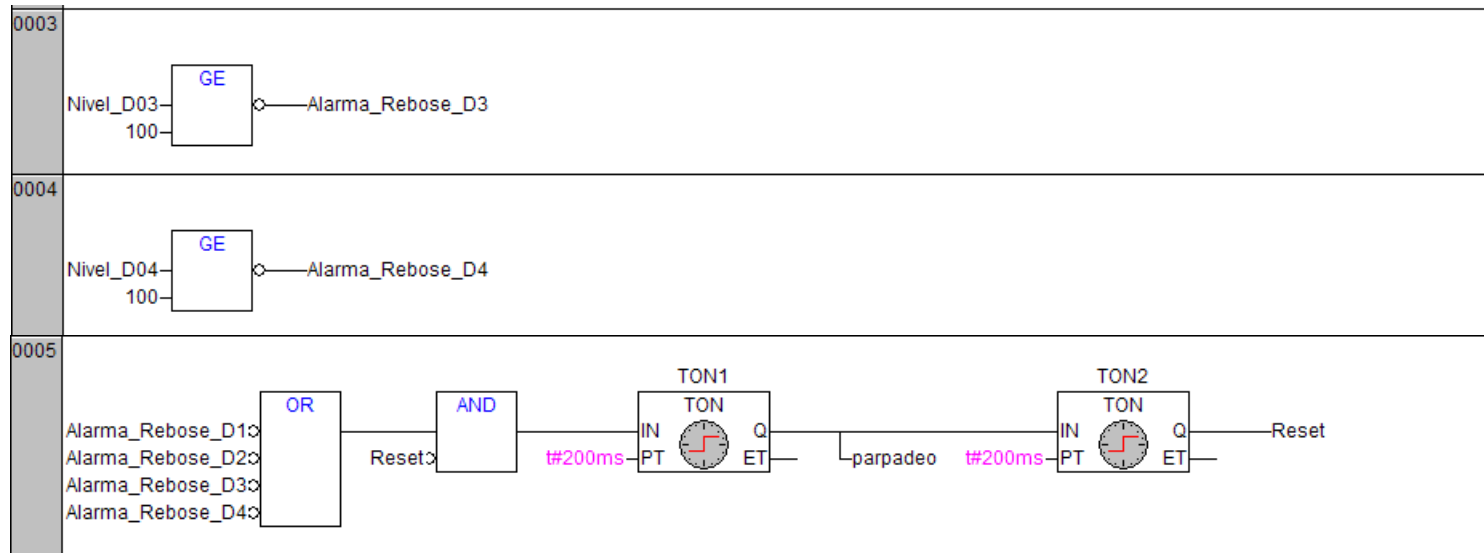
Control en lazo abierto

- **Sensores de nivel (FBD)**



Prácticas

Control en lazo abierto



Prácticas

Control en lazo abierto

- **Válvulas de vaciado (ST)**

```
0001 PROGRAM Valvulas_Vaciado
0002 VAR
0003     Pulsador_LV03: BOOL;
0004     Contador_LV03: INT:=0;
0005     A: BOOL;
0006     Trig1: R_TRIG;
0007     Pulsador_LV04: BOOL;
0008     Contador_LV04: INT:=0;
0009     B: BOOL;
0010     Trig2: R_TRIG;
0011     Pulsador_LV05: BOOL;
0012     Contador_LV05: INT:=0;
0013     C: BOOL;
0014     Trig3: R_TRIG;
0015     Pulsador_LV06: BOOL;
0016     Contador_LV06: INT;
0017     D: BOOL;
0018     Trig4: R_TRIG;
0019 END_VAR
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
```



Prácticas

Control en lazo abierto

```
0001 Trig1(CLK:= Pulsador_LV03, Q=> A);
0002 IF A=TRUE THEN
0003 Contador_LV03:=Contador_LV03+1;
0004 ELSIF Contador_LV03=1 THEN
0005 Valvula_LV03:= TRUE;
0006 ELSE
0007 Valvula_LV03:=FALSE;
0008 Contador_LV03:=0;
0009 END_IF;
0010
0011 Trig2(CLK:= Pulsador_LV04, Q=> B);
0012 IF B=TRUE THEN
0013 Contador_LV04:=Contador_LV04+1;
0014 ELSIF Contador_LV04=1 THEN
0015 Valvula_LV04:= TRUE;
0016 ELSE
0017 Valvula_LV04:=FALSE;
0018 Contador_LV04:=0;
0019 END_IF;
0020
0021 Trig3(CLK:= Pulsador_LV05, Q=> C);
0022 IF C=TRUE THEN
0023 Contador_LV05:=Contador_LV05+1;
0024 ELSIF Contador_LV05=1 THEN
0025 Valvula_LV05:= TRUE;
0026 ELSE
0027 Valvula_LV05:=FALSE;
0028 Contador_LV05:=0;
0029 END_IF;
```

Prácticas

Control en lazo abierto

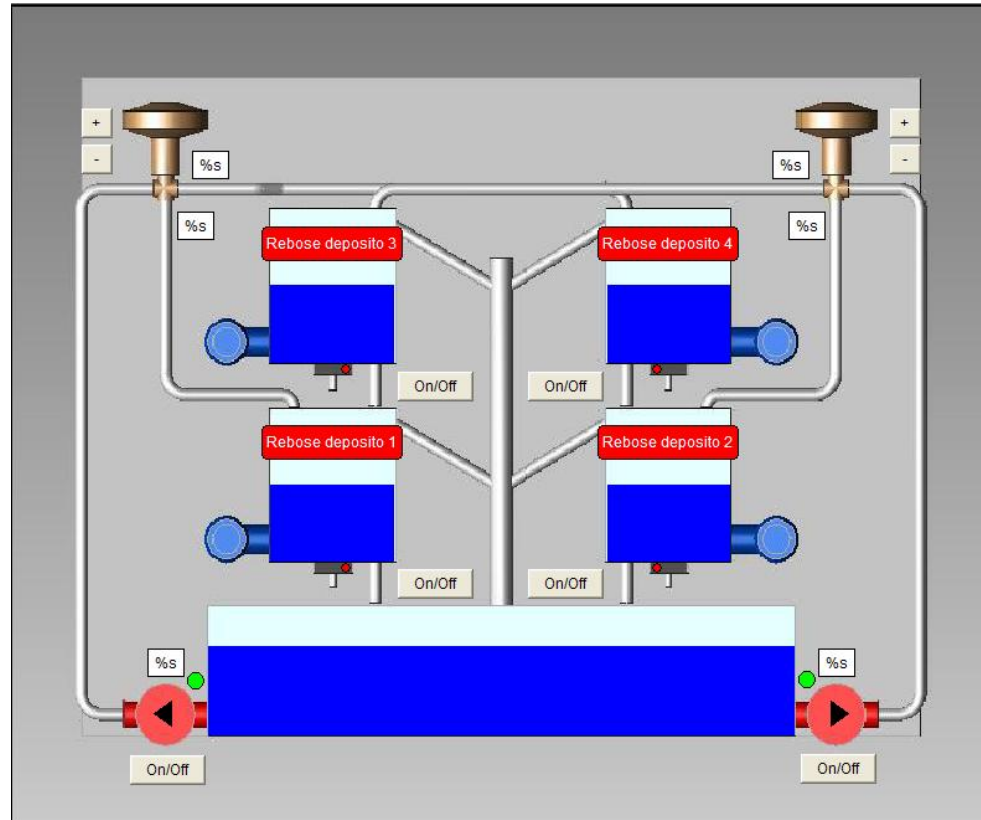
```
0021 Trig3(CLK:= Pulsador_LV05, Q=> C);
0022 IF C=TRUE THEN
0023 Contador_LV05:=Contador_LV05+1;
0024 ELSIF Contador_LV05=1 THEN
0025 Valvula_LV05:= TRUE;
0026 ELSE
0027 Valvula_LV05:=FALSE;
0028 Contador_LV05:=0;
0029 END_IF;
0030
0031 Trig4(CLK:= Pulsador_LV06, Q=> D);
0032 IF D=TRUE THEN
0033 Contador_LV06:=Contador_LV06+1;
0034 ELSIF Contador_LV06=1 THEN
0035 Valvula_LV06:= TRUE;
0036 ELSE
0037 Valvula_LV06:=FALSE;
0038 Contador_LV06:=0;
0039 END_IF;
```



Prácticas

Control en lazo abierto

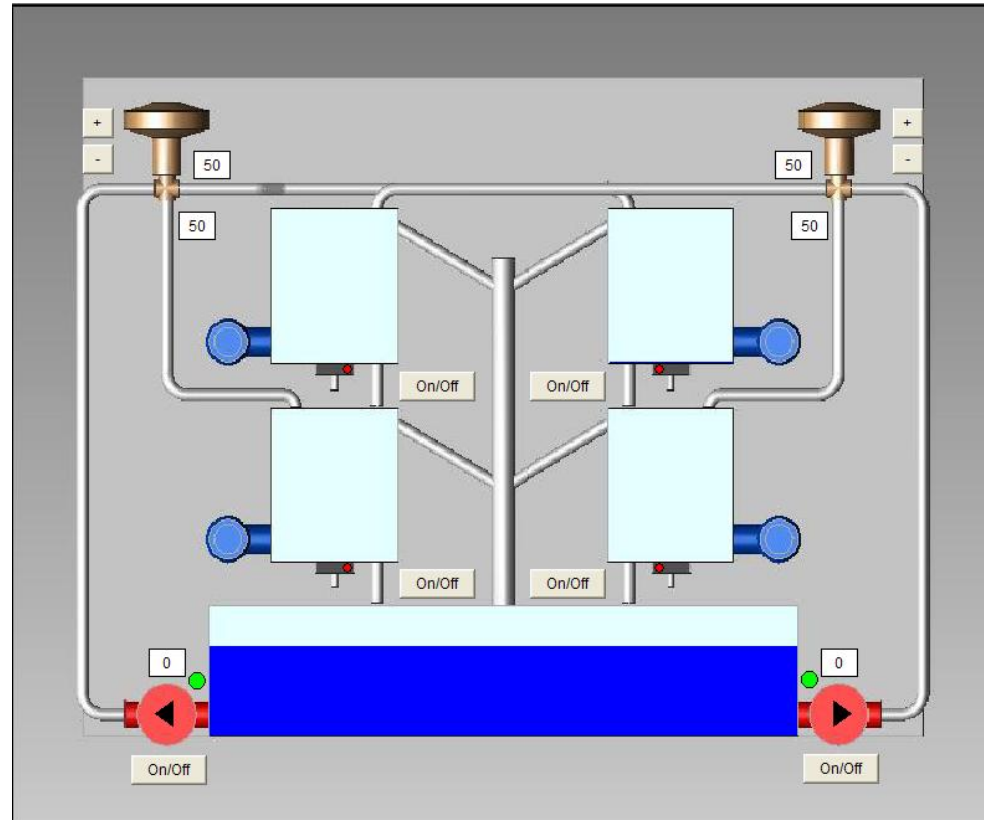
Visualización (HMI) offline



Prácticas

Control en lazo abierto

Visualización (HMI) online



Prácticas

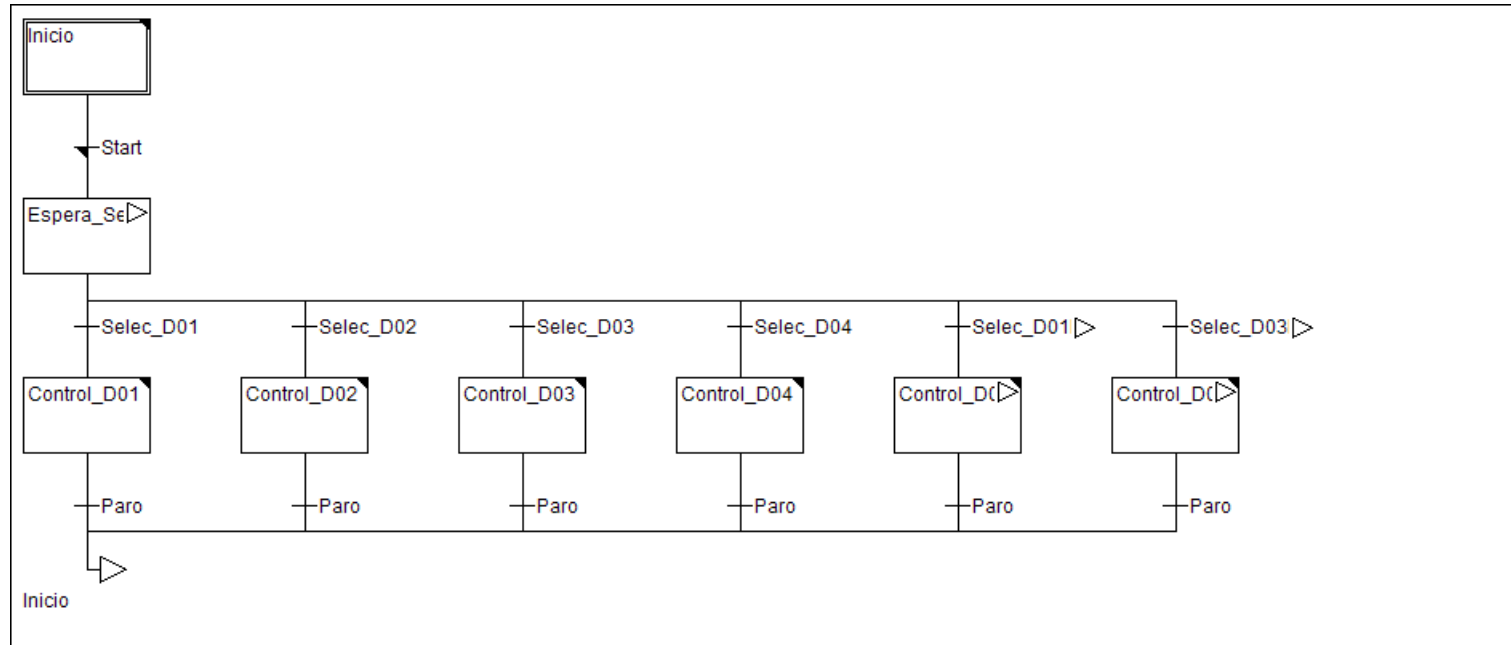
Control en lazo cerrado

Programa de control (SFC)

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     Marcha: BOOL;
0004     Selec_D01: BOOL;
0005     Selec_D02: BOOL;
0006     Selec_D03: BOOL;
0007     Selec_D04: BOOL;
0008     Selec_D01D02: BOOL;
0009     Selec_D03D04: BOOL;
0010     Paro: BOOL;
0011     Control_D01_Activa: BOOL;
0012     Control_D02_Activa: BOOL;
0013     Control_D03_Activa: BOOL;
0014     Control_D04_Activa: BOOL;
0015 END_VAR
```

Prácticas

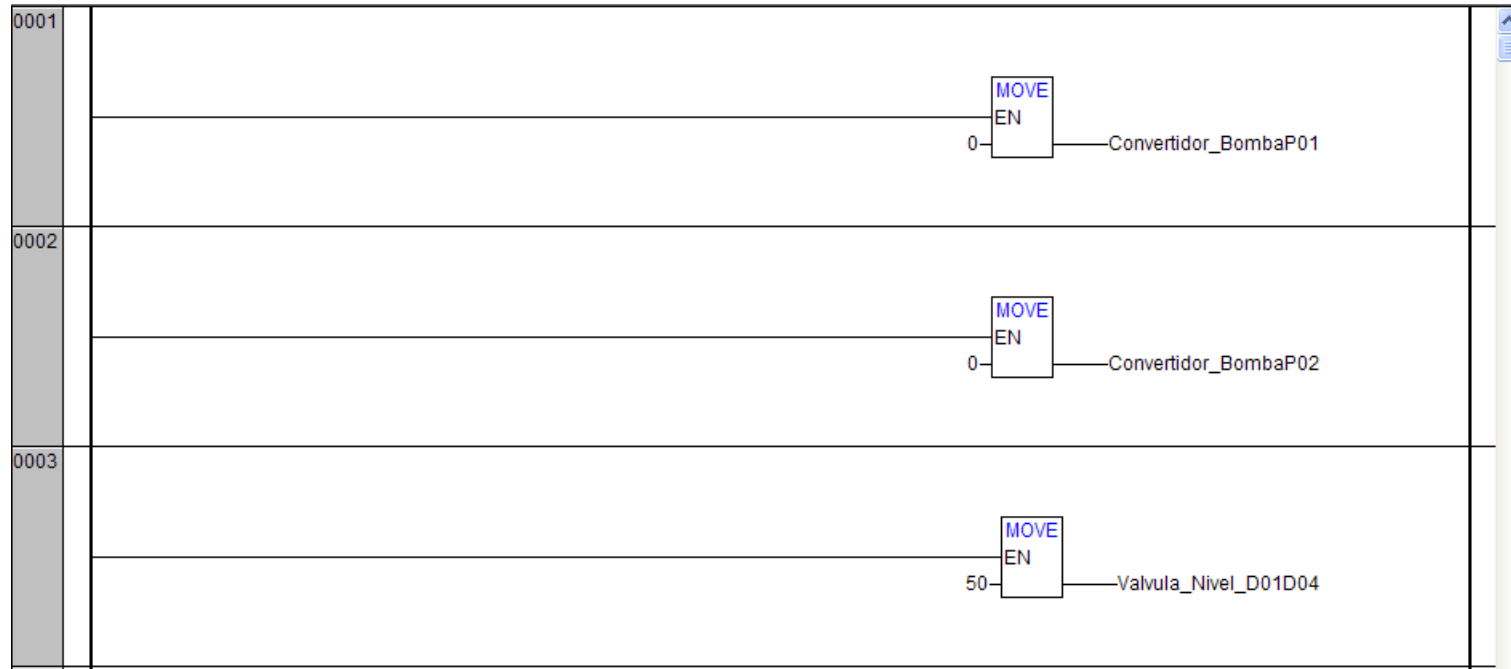
Control en lazo cerrado



Prácticas

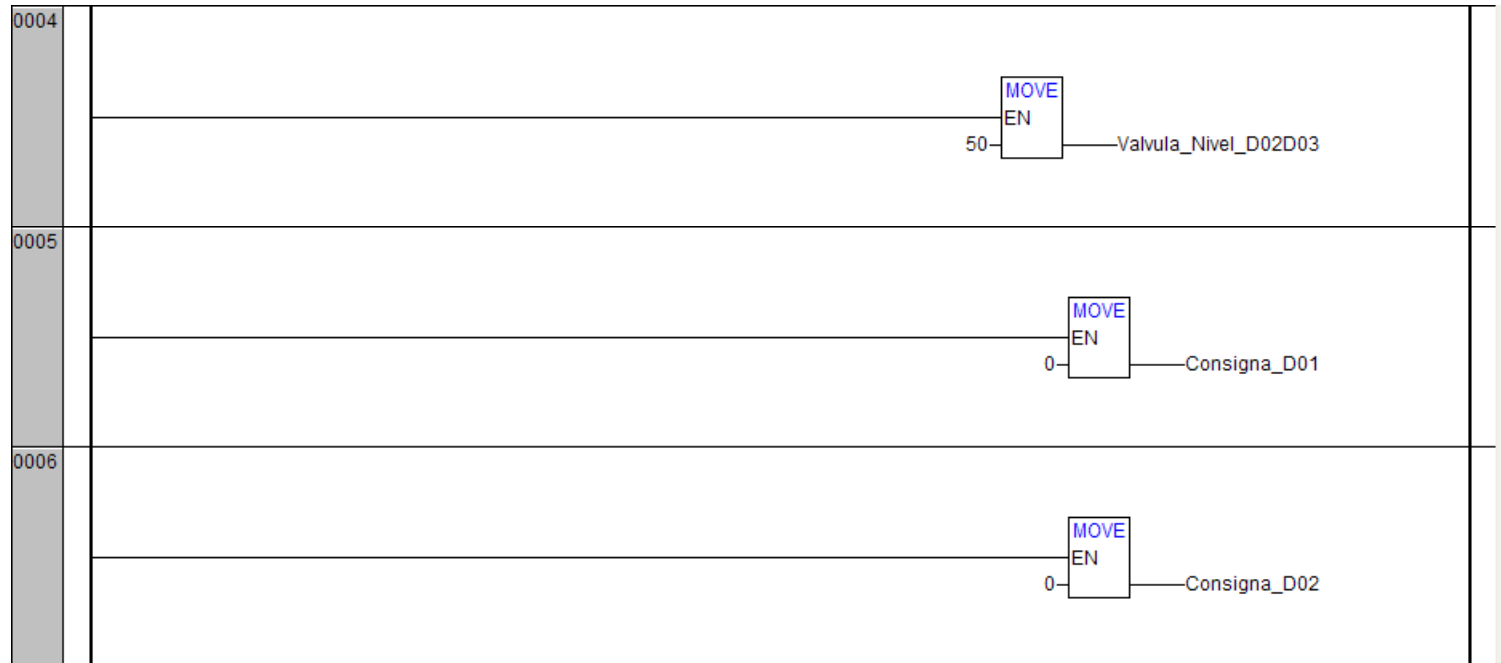
Control en lazo cerrado

- **Acción asociada a la etapa “Inicio” (LD)**



Prácticas

Control en lazo cerrado



Prácticas

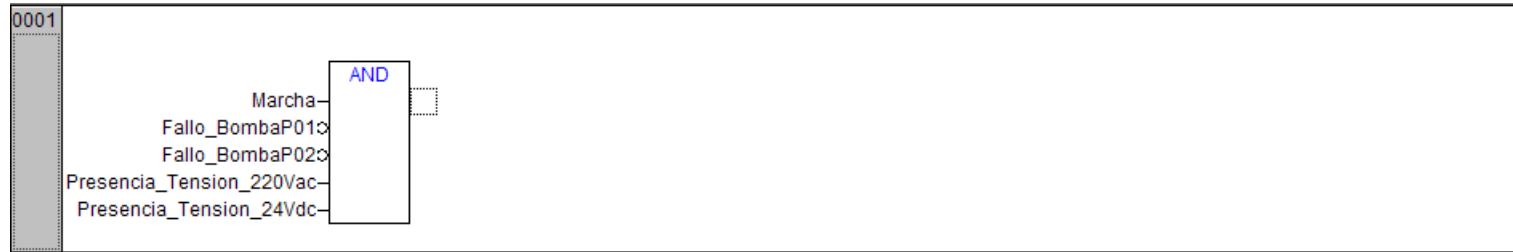
Control en lazo cerrado



Prácticas

Control en lazo cerrado

- **Condición asociada a la transición “Start” (FBD)**



Prácticas

Control en lazo cerrado

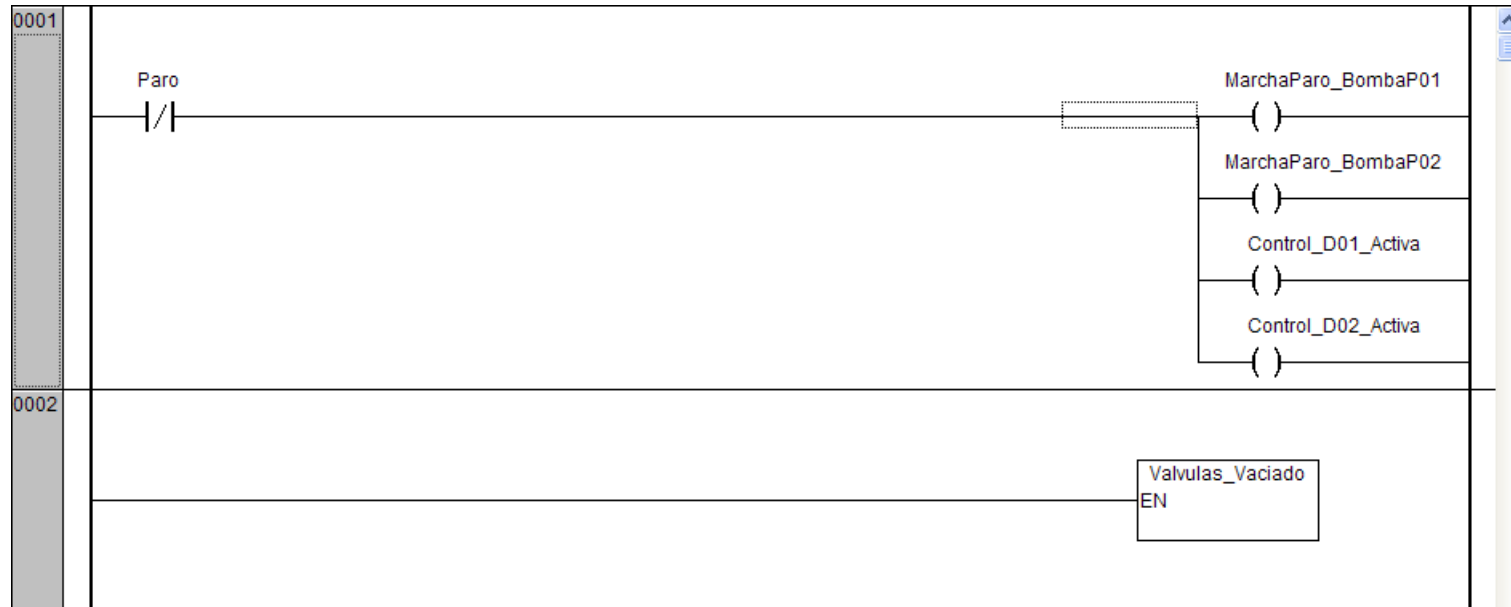
- **Acción asociada a la etapa “Control_Do1” (LD)**



Prácticas

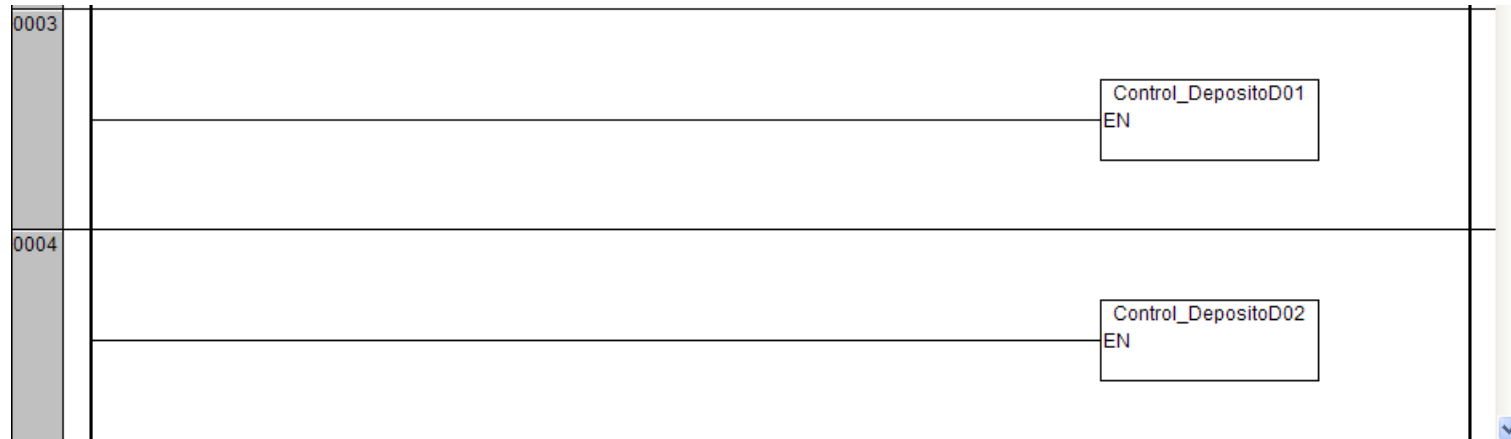
Control en lazo cerrado

- **Acción asociada a la etapa “Control_Do1Do2” (LD)**



Prácticas

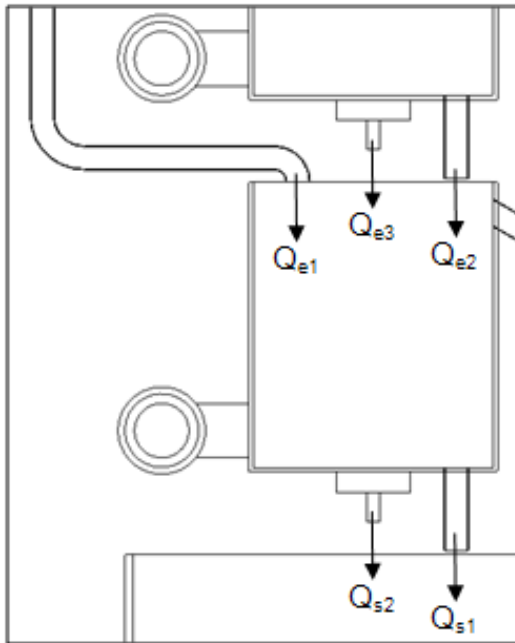
Control en lazo cerrado



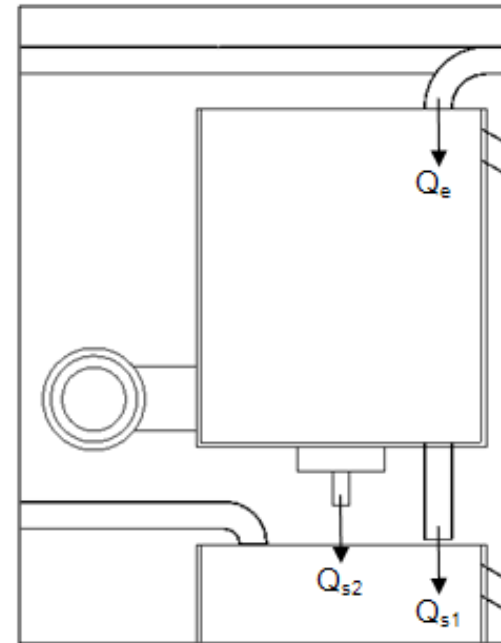
Prácticas

Control en lazo cerrado

Caudales depósitos 1 y 2



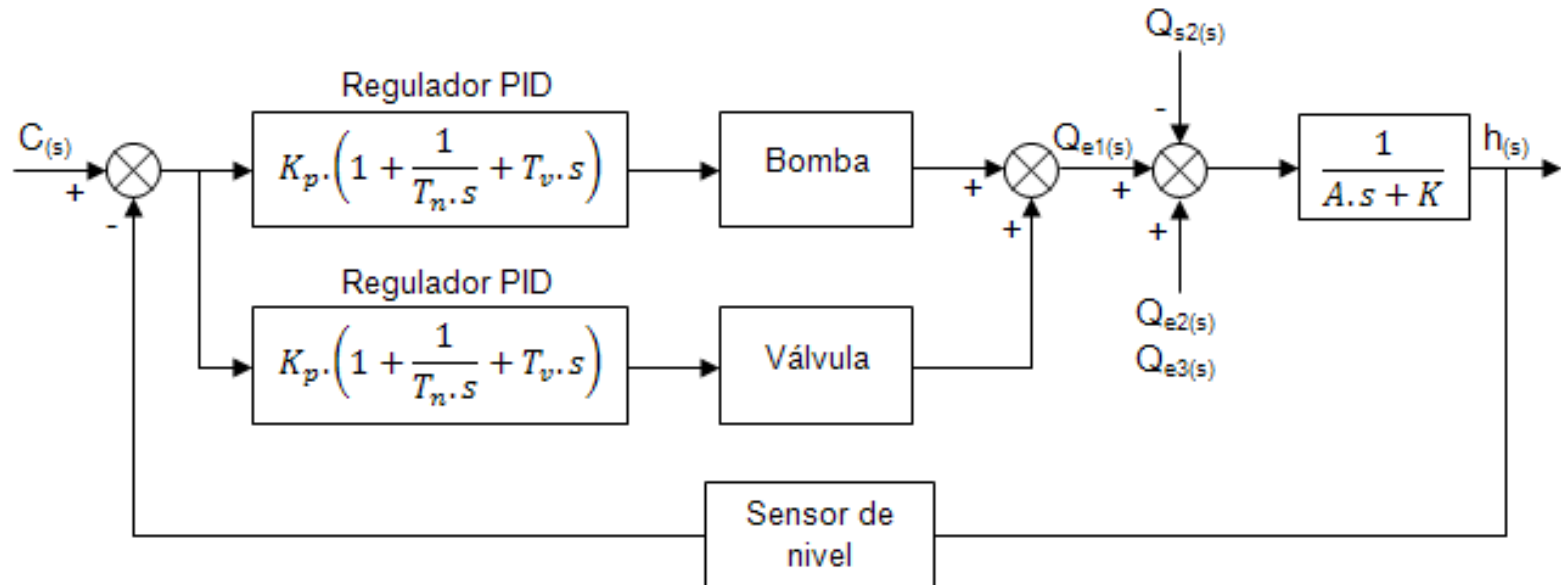
Caudales depósitos 3 y 4



Prácticas

Control en lazo cerrado

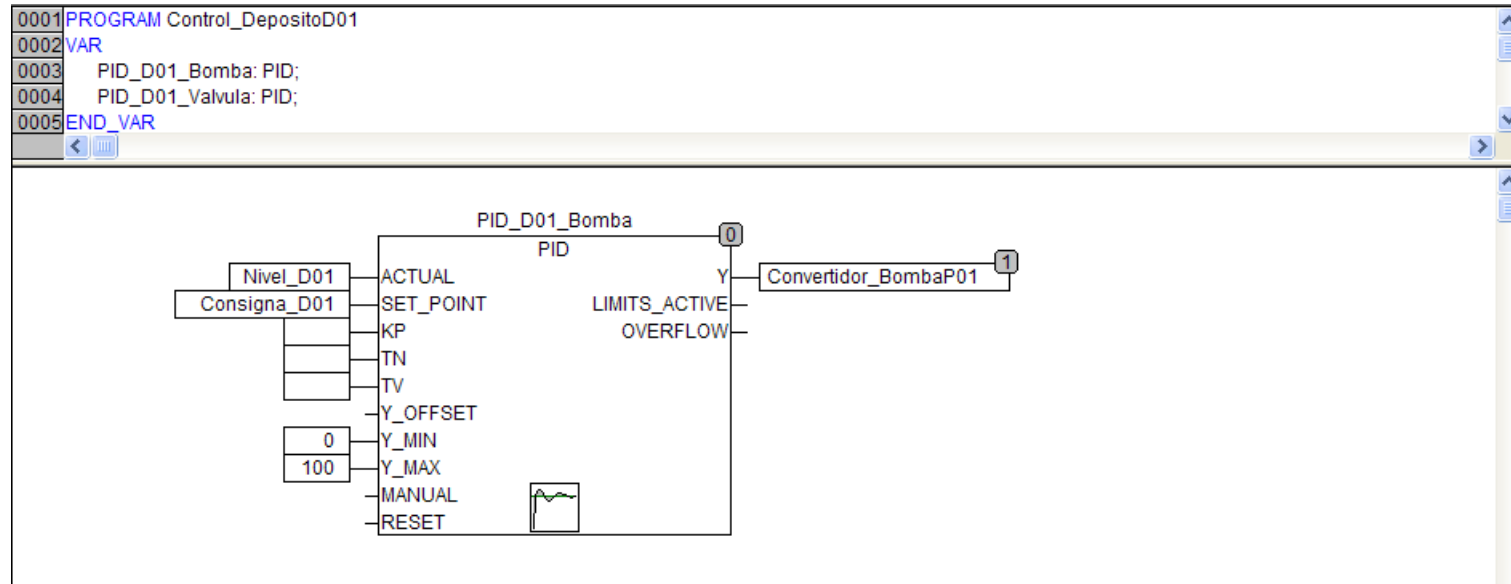
Lazo de control



Prácticas

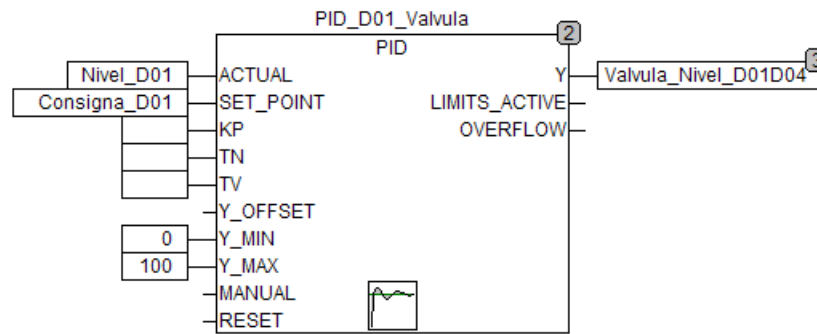
Control en lazo cerrado

Programa control depósito 1



Prácticas

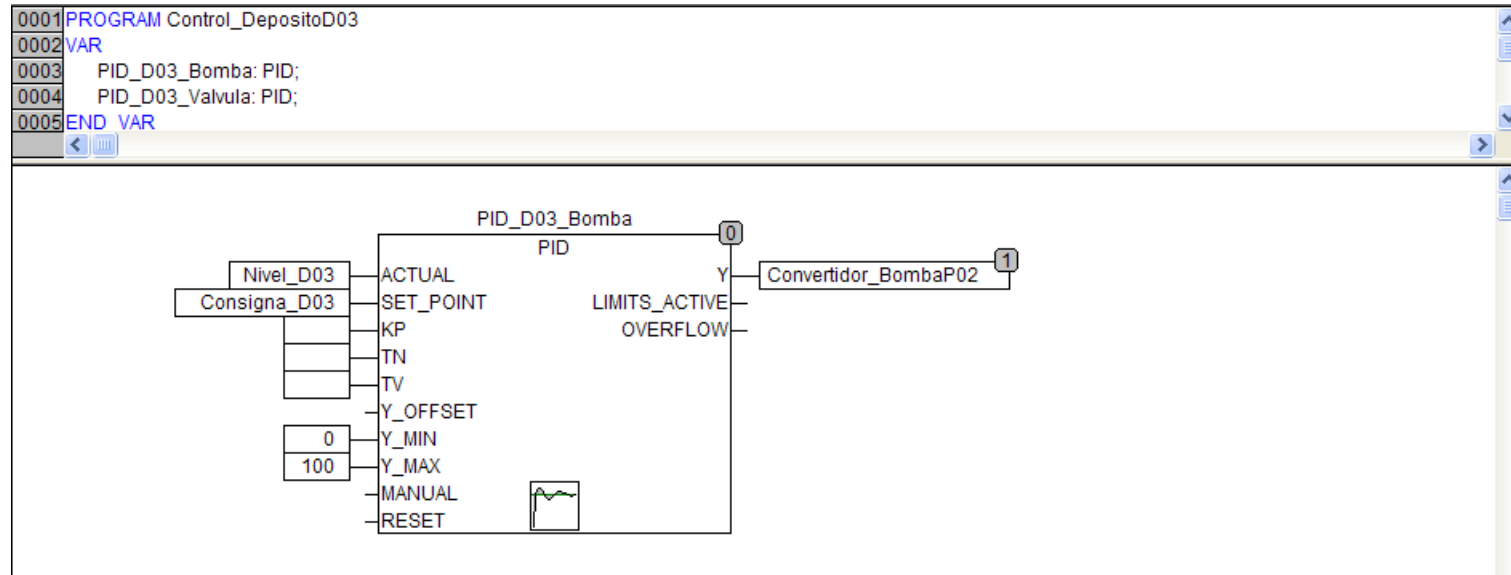
Control en lazo cerrado



Prácticas

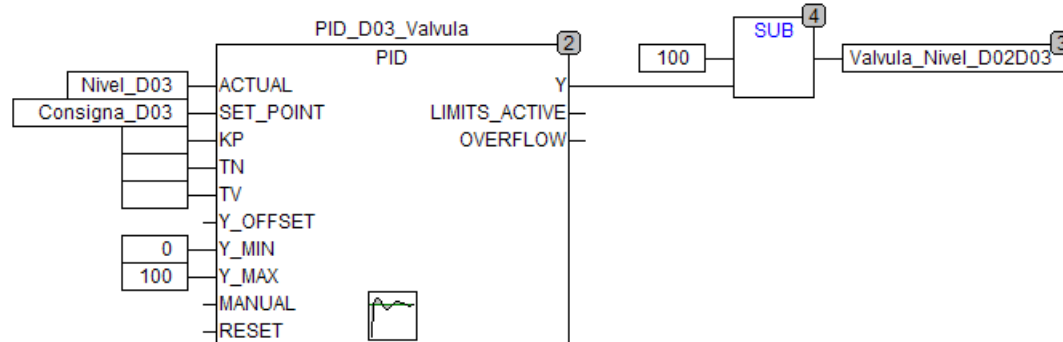
Control en lazo cerrado

Programa control depósito 3



Prácticas

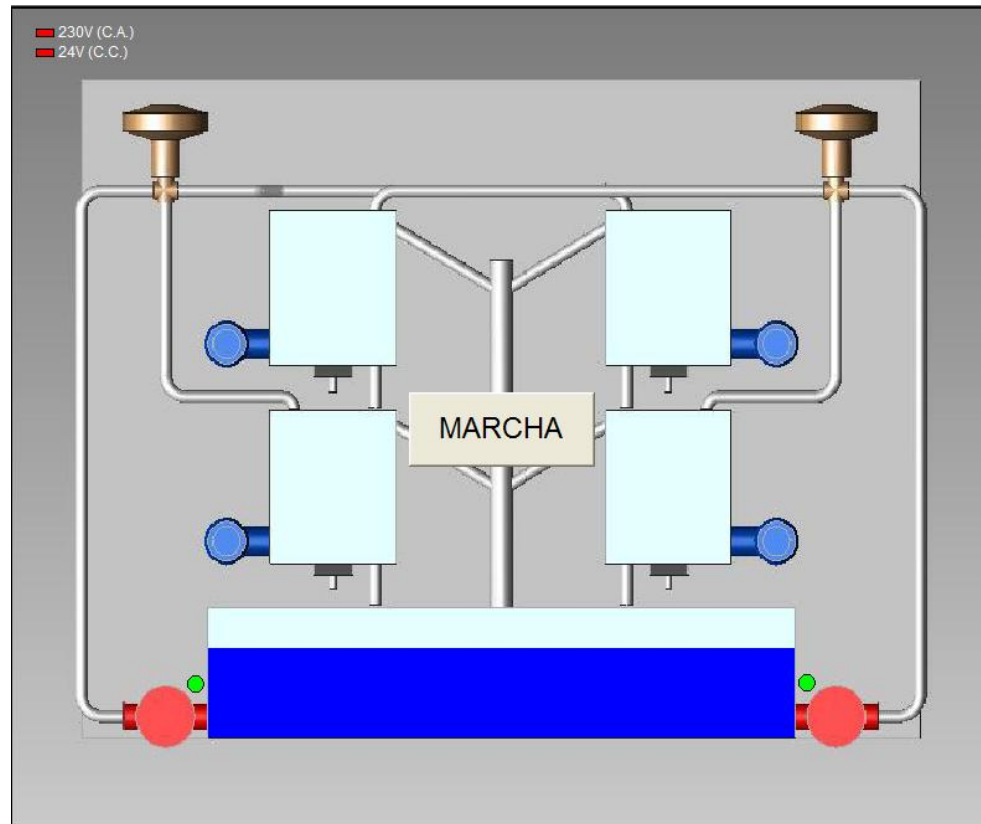
Control en lazo cerrado



Prácticas

Control en lazo cerrado

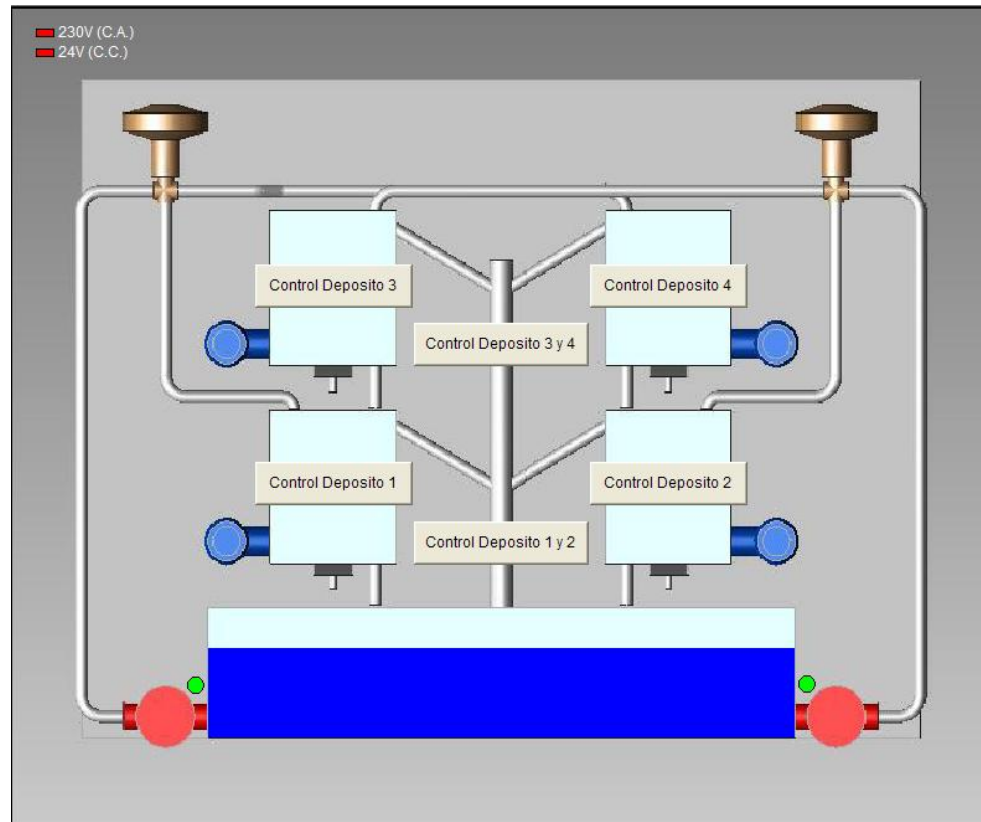
Visualización (HMI) Pantalla inicial



Prácticas

Control en lazo cerrado

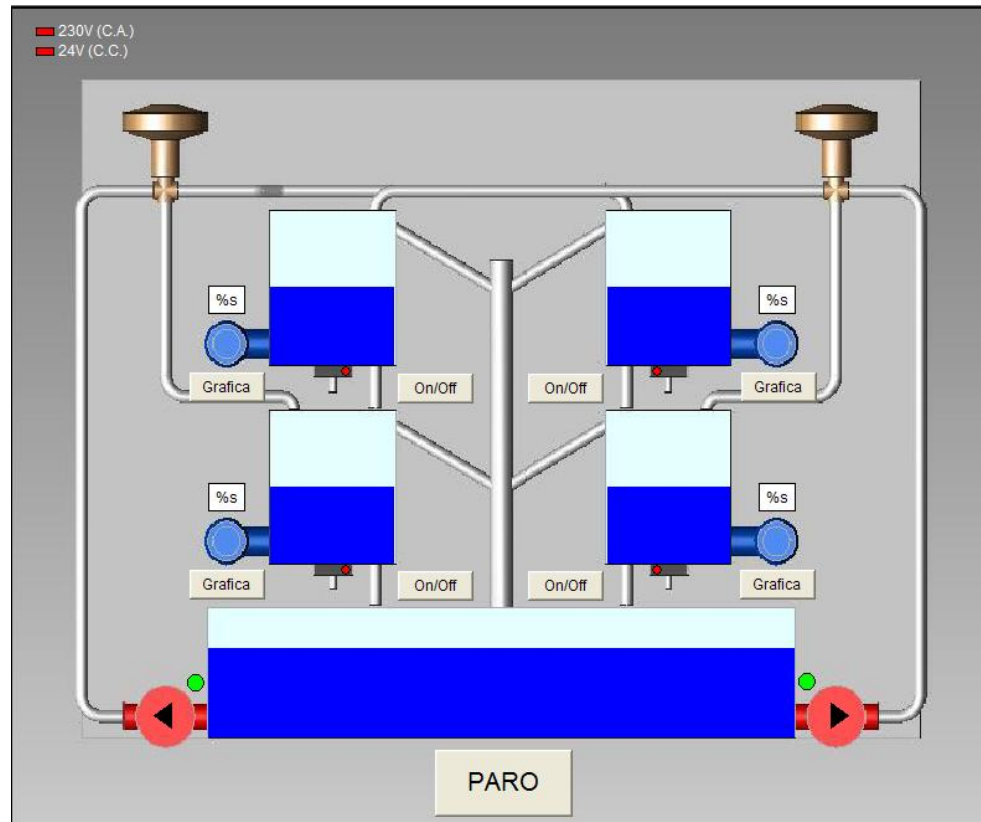
Visualización (HMI) Pantalla de selección



Prácticas

Control en lazo cerrado

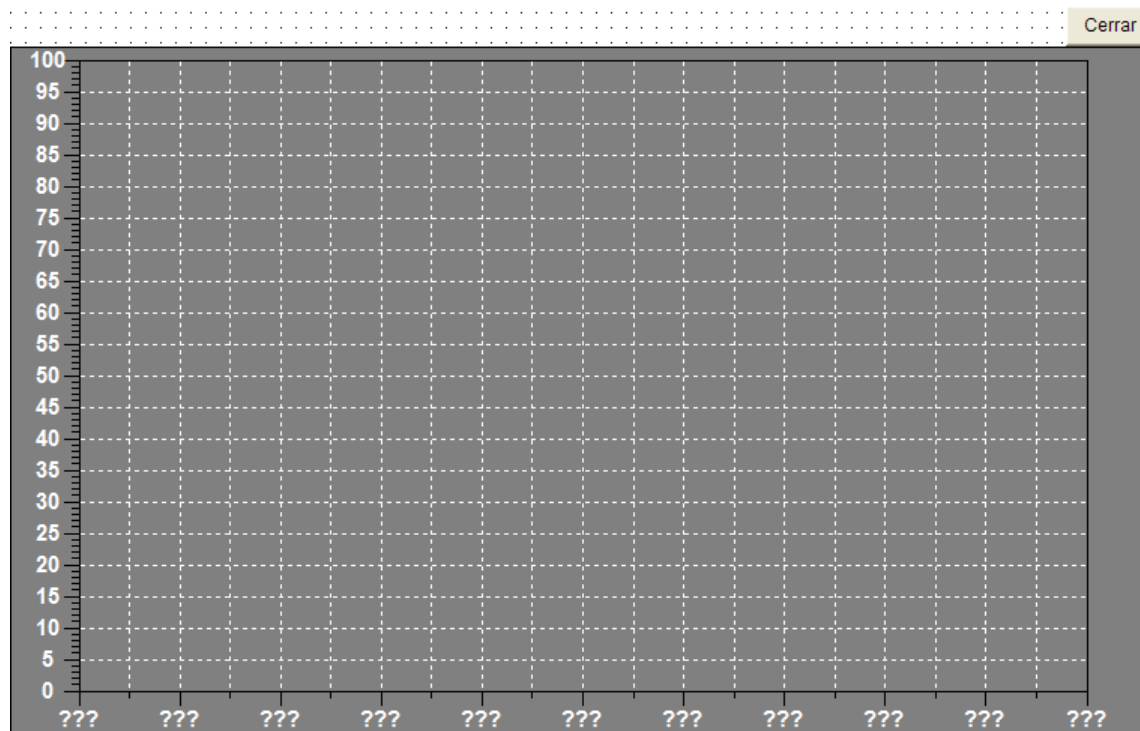
Visualización (HMI) Pantalla de ejecución



Prácticas

Control en lazo cerrado

Visualización (HMI) Gráfica nivel depósitos



Índice

Introducción

Estándar IEC 61131-3

Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

Índice

Introducción

Estándar IEC 61131-3

Lenguajes de programación

CoDeSys

Prácticas

Conclusiones

Conclusiones

- El estándar IEC 61131-3 es una buena base para el desarrollo para autómatas programables, ya que el número de lenguajes es suficientemente amplio y adaptados a todos los niveles formativos.
- El software CoDeSys basado en el estándar IEC 61131-3 es de lo mas completo, además de disponer de herramientas adicionales como el configurador OPC.
- El autómata MOELLER XC201 basado en el estándar IEC 61131-3 tiene unas buenas prestaciones, además de conectividad Ethernet y un servidor OPC muy útil para intercambio de datos en sistemas distribuidos.