

# Guía Rápida CodeSys: IO-Link Call Modbus TCP.



www.elion.es

Comunicación EtherNet IP: NX1P2 & TBEN

Versión: 1.0

19/12/2017

**Servicio Asistencia Técnica**  
Farell, 5  
08014 Barcelona  
Tel. 932 982 040  
soporte.tecnico@elion.es

 **elion**<sup>®</sup>

## 1. Alcance

En esta guía se pretende mostrar como utilizar, desde el inicio de un proyecto, el bloque de función IO-Link Call en Codesys.

Este FB nos permite leer y escribir valores sobre un sensor IO-Link, pudiendo así conocer su estado en todo momento o asignarle nuevos valores. También podemos conocer el estado de diagnósticos del sensor, en caso de que disponga.

Utilizarmeos:

- Codesys V3.5 SP11 Patch2
- TBEN-S2-4IOL

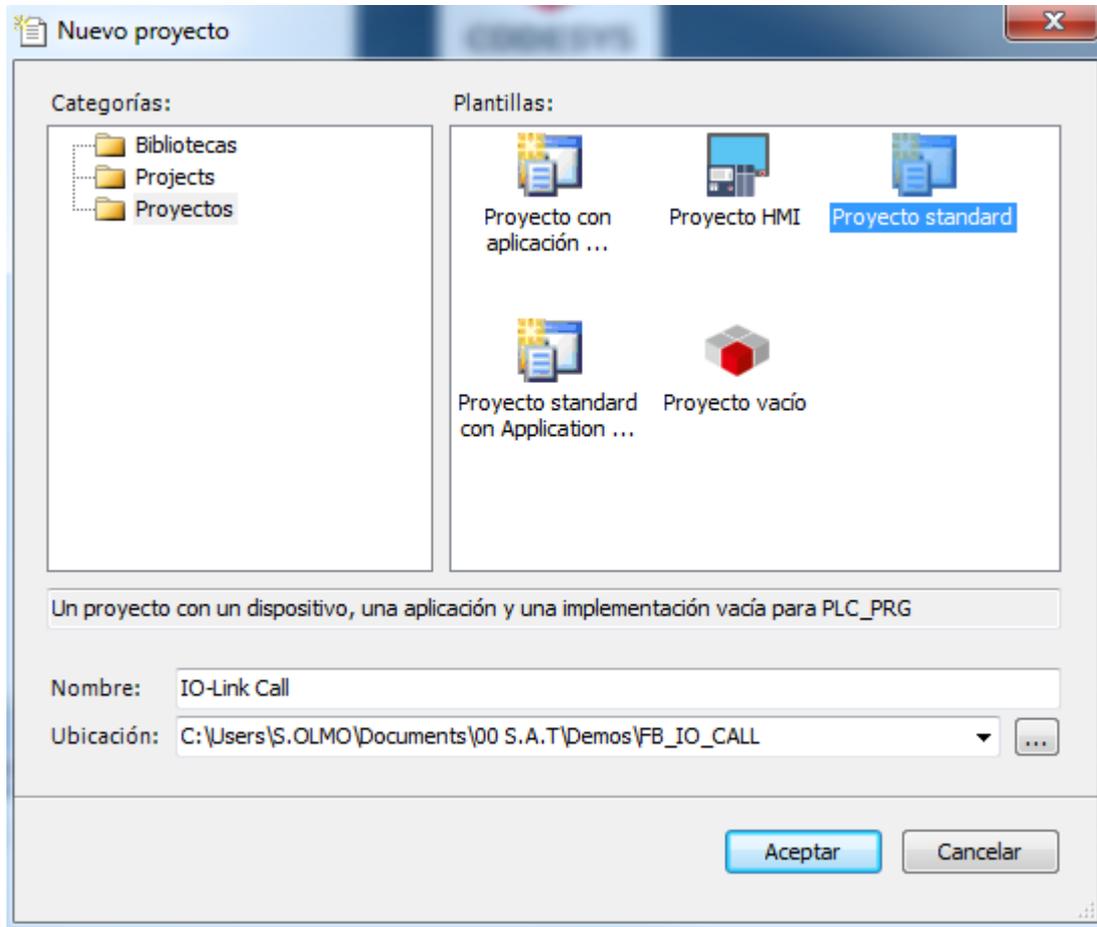
Enlace descarga: [Librería IO-Link FB](#)



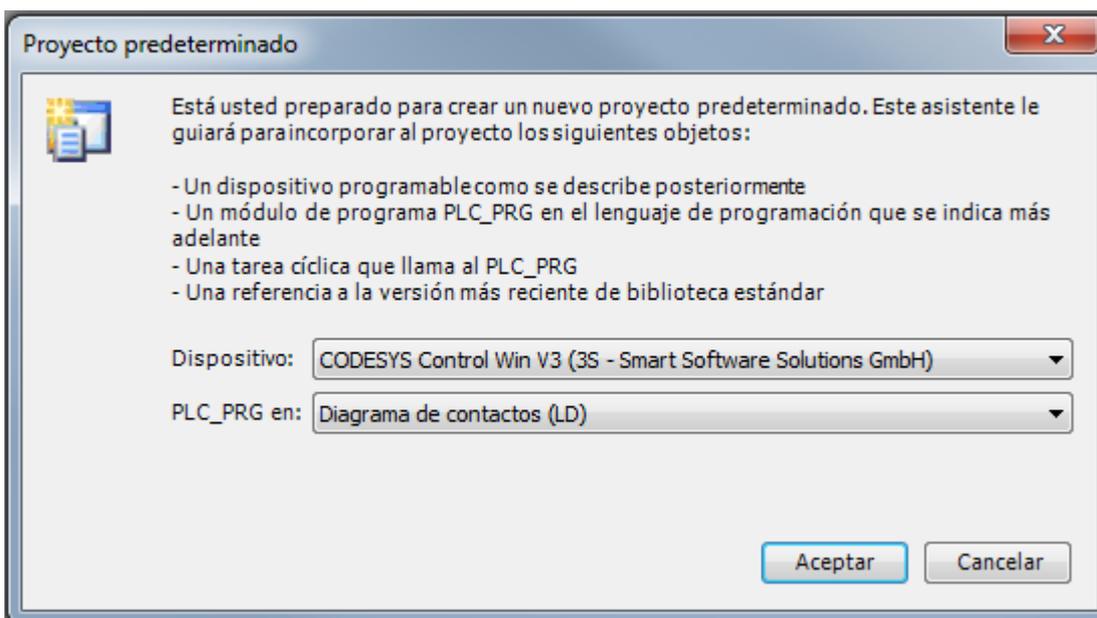
## 2. Codesys

### a. Crear proyecto

Empezamos creando un proyecto en Codesys estándar.



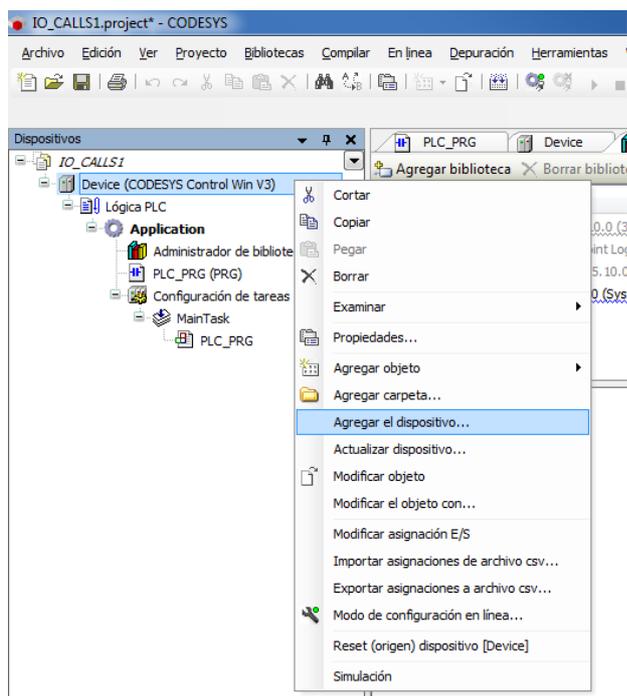
Utilizaremos el PLC simulado desde el ordenador y la programación tipo LADDER (LD).



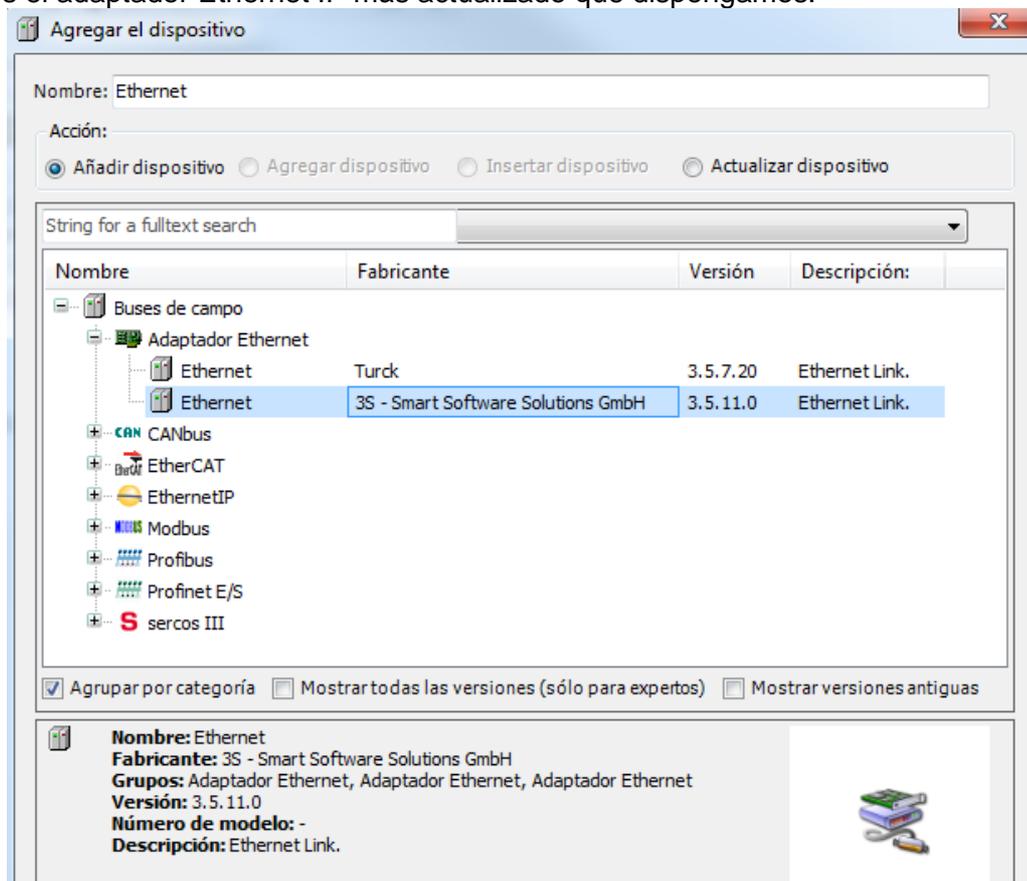
## b. Configurar Hardware

Una vez ha cargado la configuración del proyecto, debemos agregar el módulo TBEN-S2-4IOL como esclavo Modbus TCP de nuestro PLC.

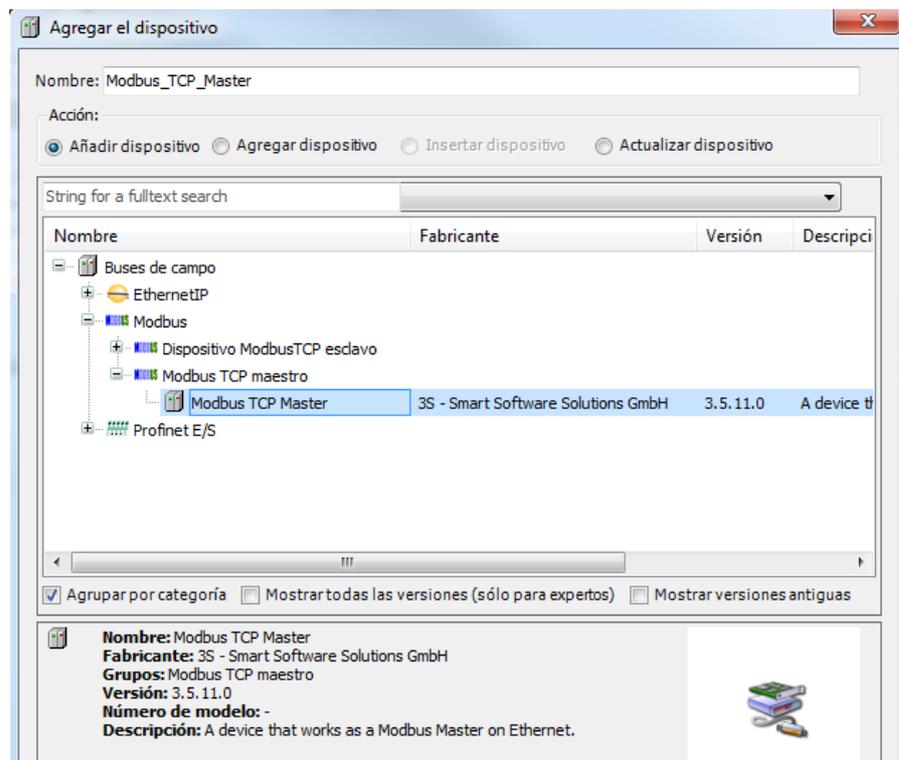
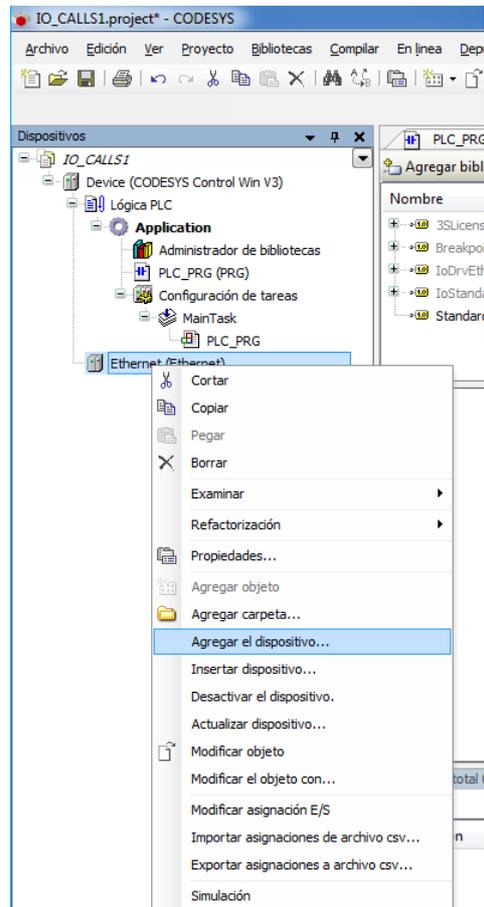
Para ello, click derecho sobre Device y agregar dispositivo que aparece en el desplegable.



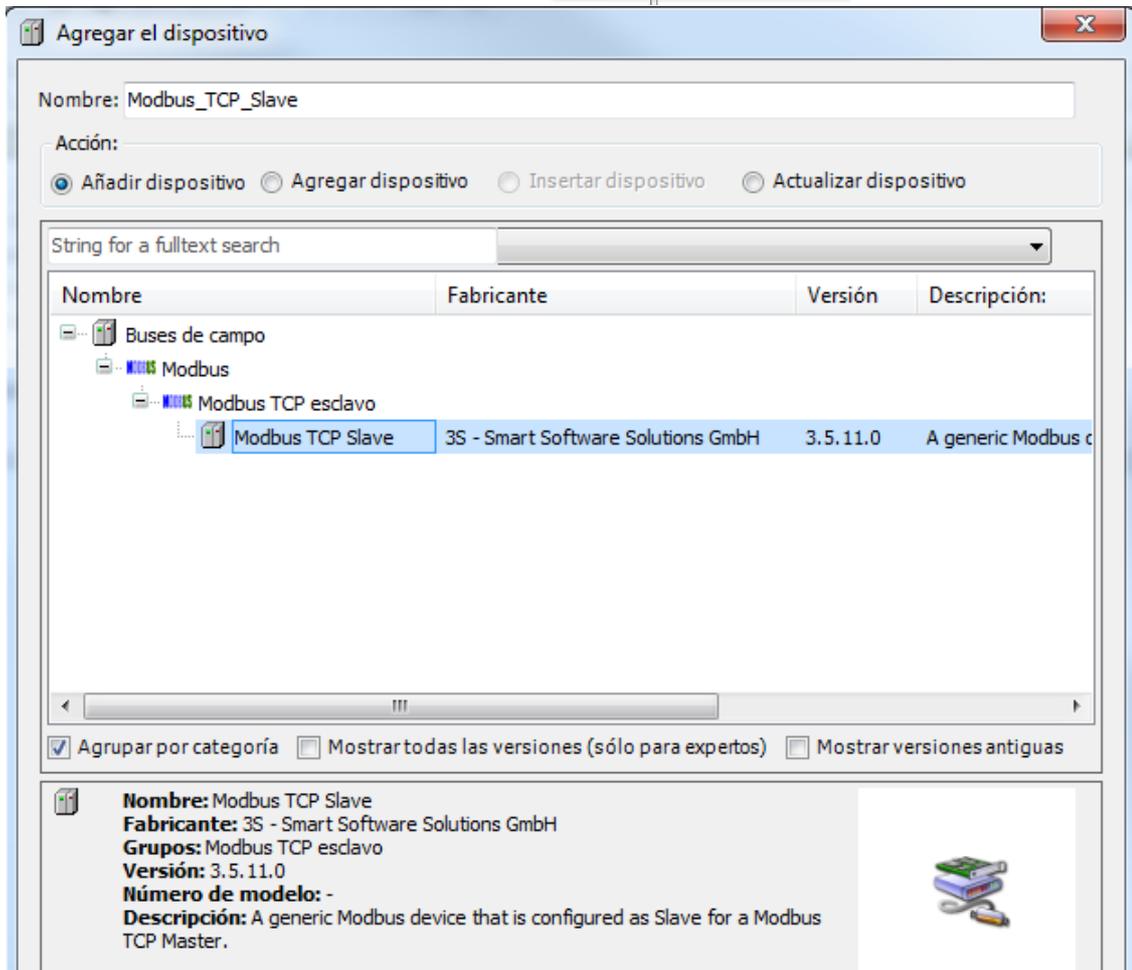
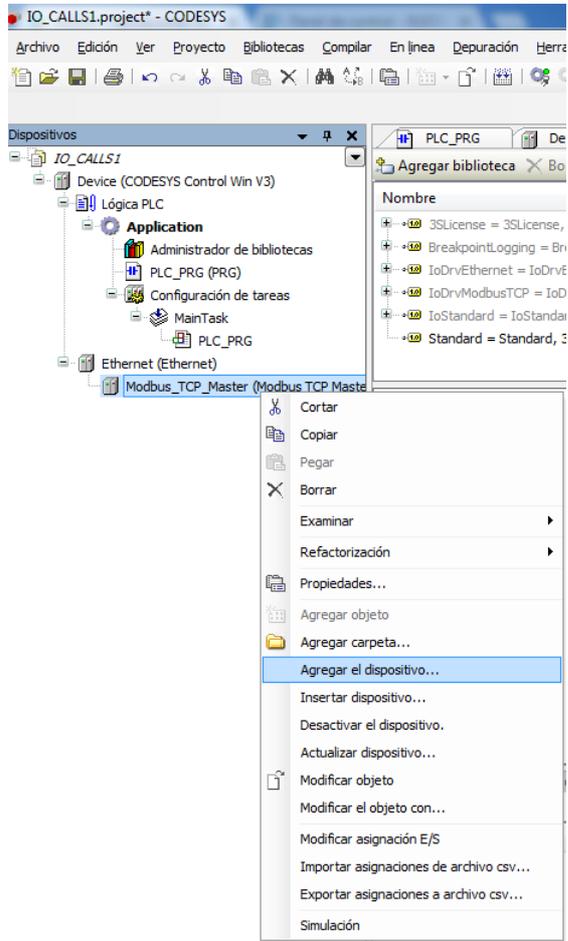
Agregaremos el adaptador Ethernet IP más actualizado que dispongamos:



Cuando se haya añadido este adaptador de Ethernet, repetiremos este proceso sobre él para agregar un maestro Modbus TCP.

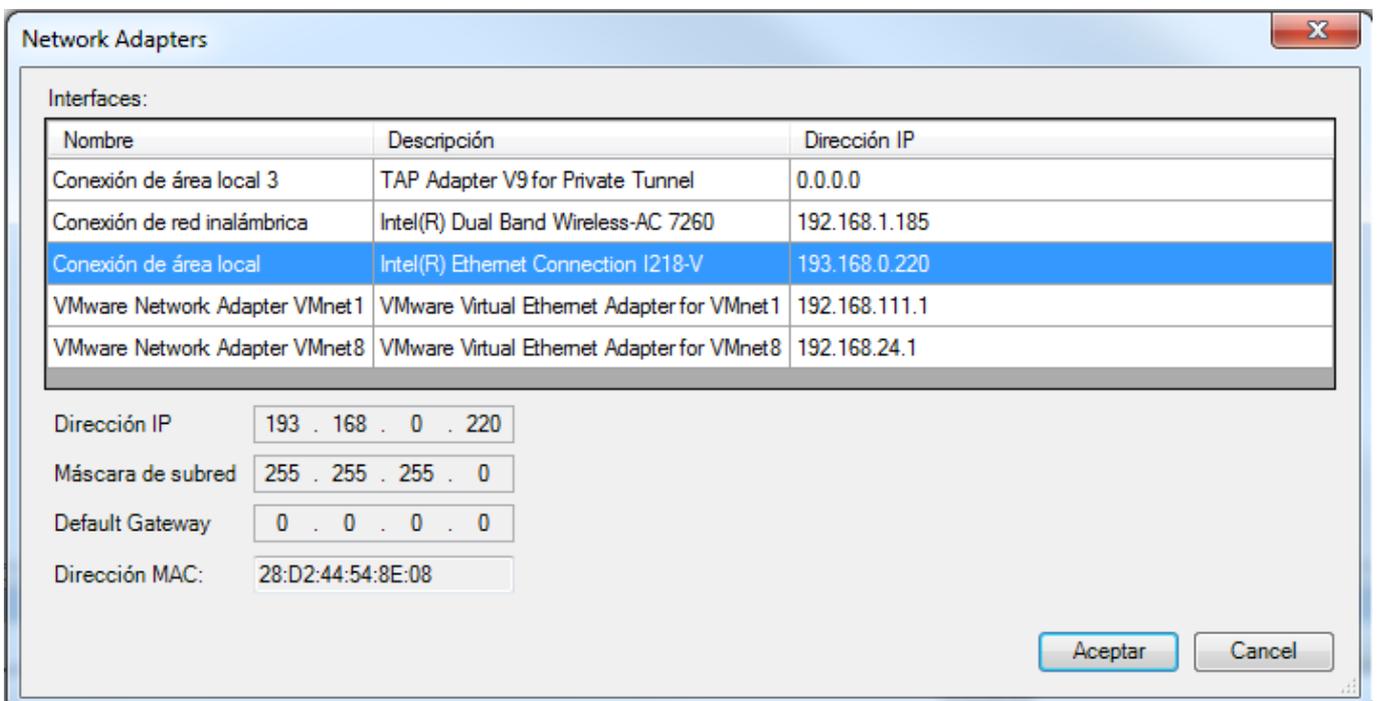
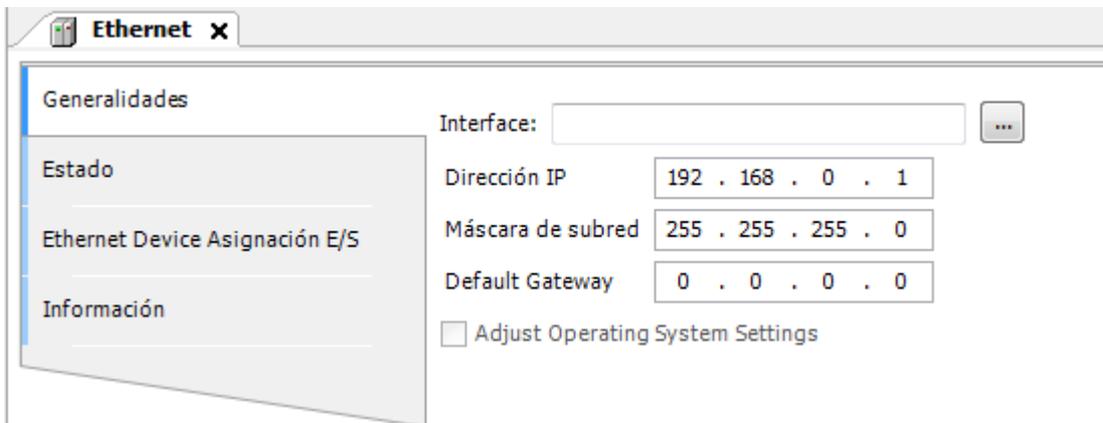


Y repetiremos una vez más este proceso sobre el Modbus TCP Master.

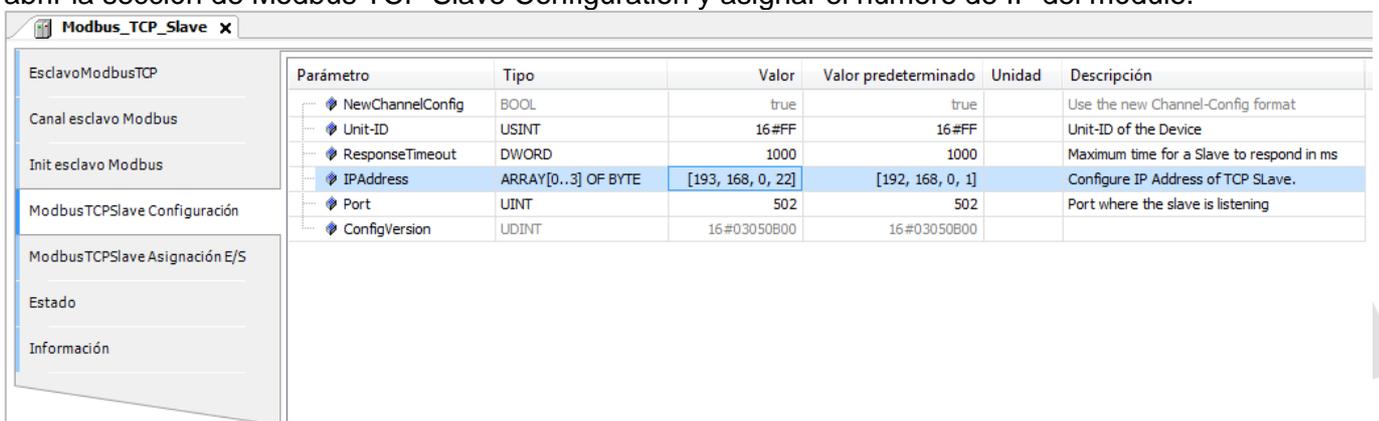


Una vez agregados estos tres elementos a nuestro proyecto, los debemos de configurar con las IP pertinentes.

Doble click sobre Ethernet (Ethernet) y en el botón de '...', aquí seleccionamos nuestra forma de conexión al módulo (Tarjeta de Red Local)



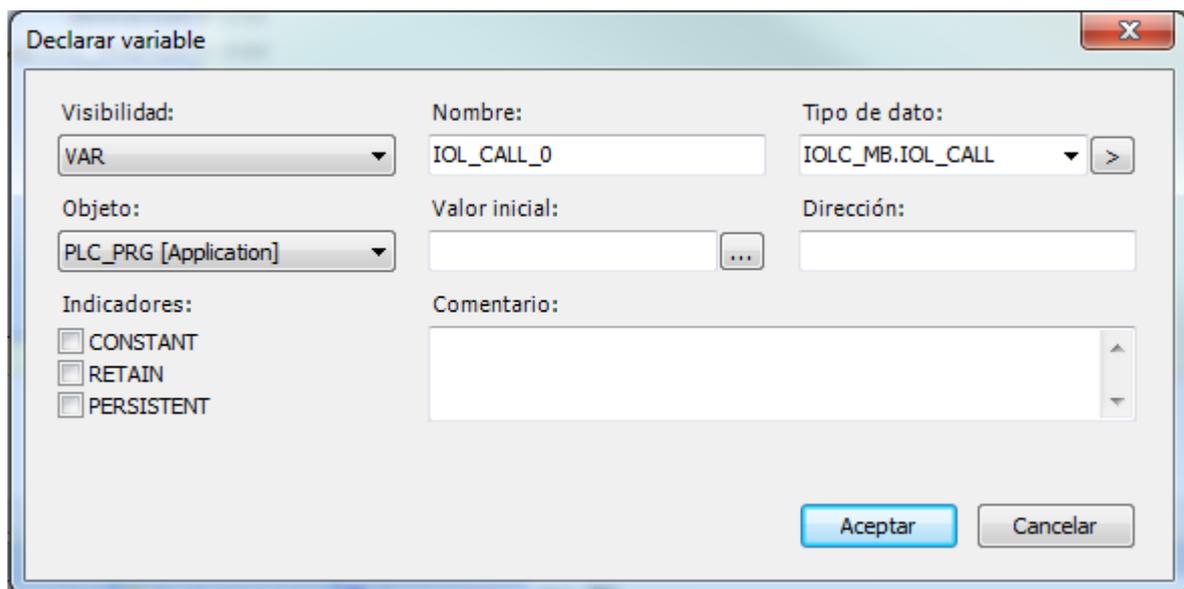
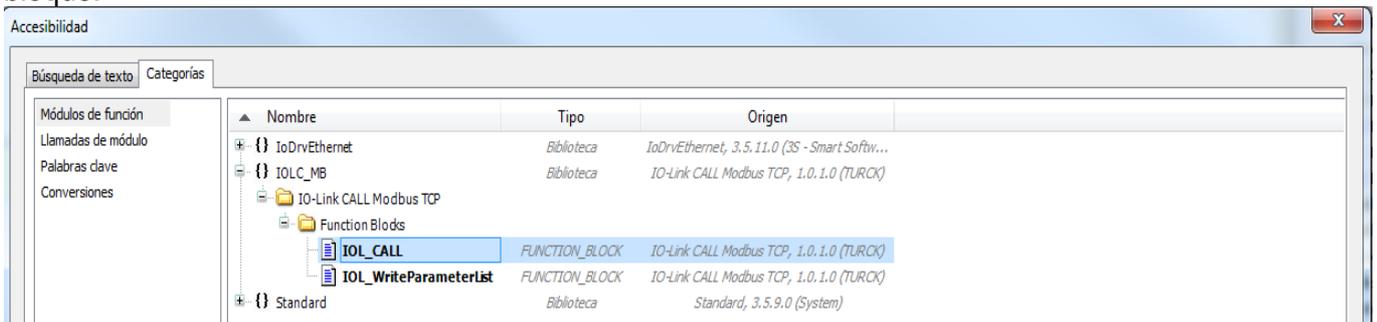
Aceptamos y hacemos doble click sobre Modbus\_TCP\_Slave, en el menú que aparece, deberemos abrir la sección de Modbus TCP Slave Configuration y asignar el número de IP del módulo.



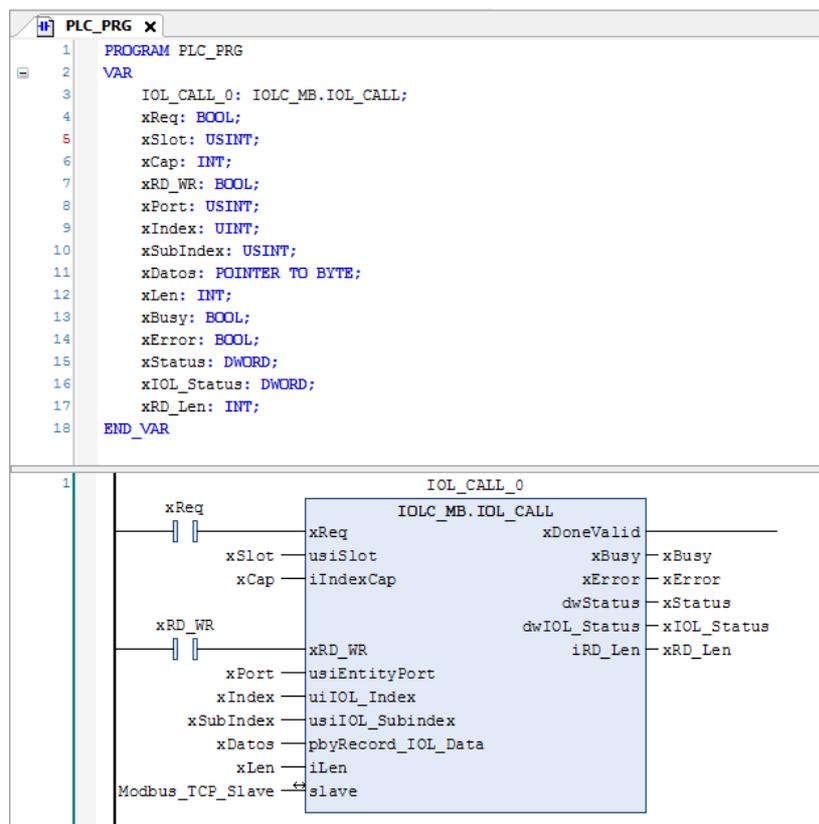
Ahora, ya tenemos configurada toda la red Hardware de la instalación.

## c. Programación

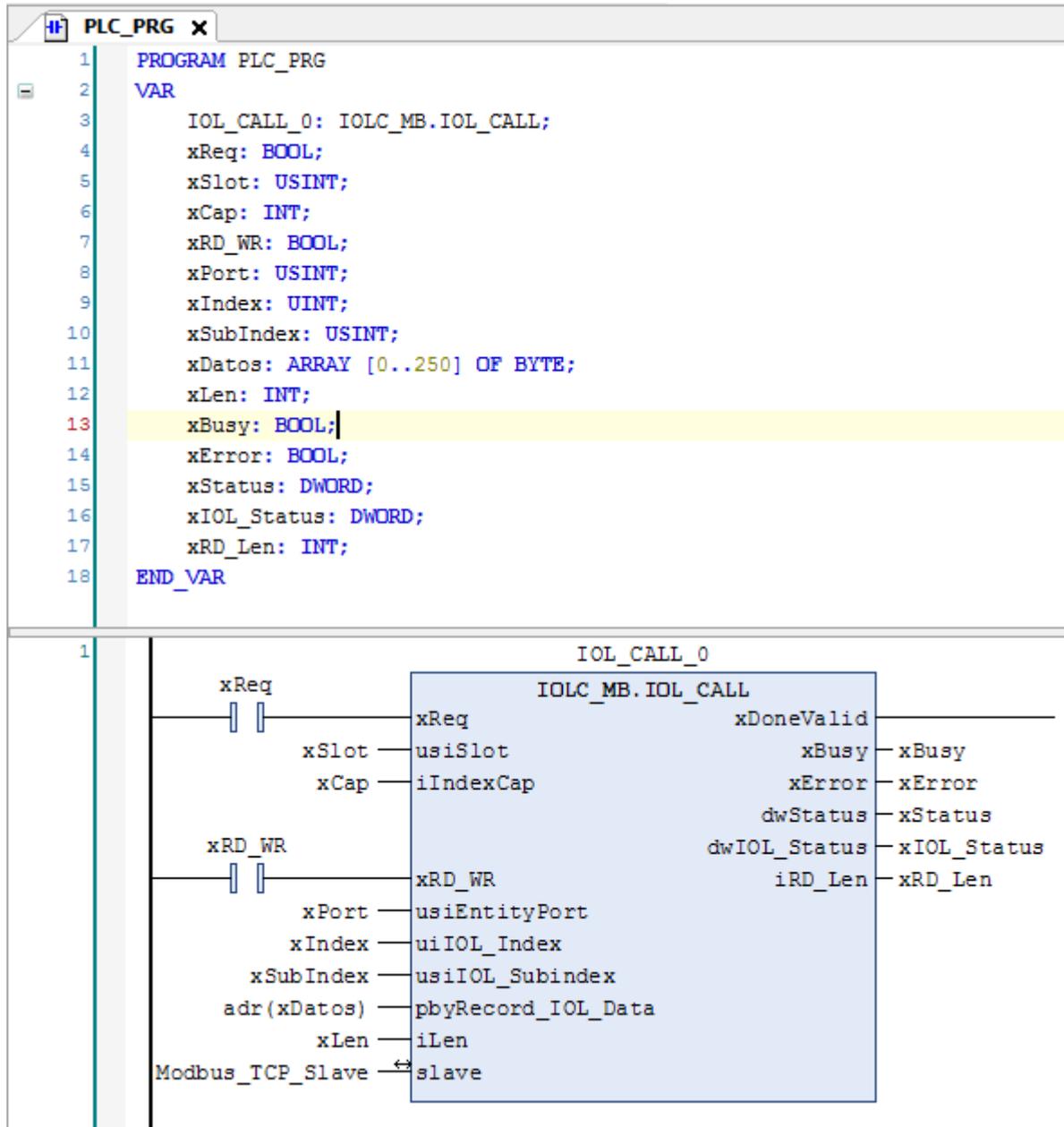
Lo primero que debemos hacer, es llamar al bloque de función y crear (automáticamente) la llamada del bloque.



El siguiente paso, será rellenar las entradas y salidas del FB, podemos escribirlas y generarlas automáticamente.



Si hemos ido creando las variables mientras las escribíamos en el FB, debemos ir a la variable asignada en la entrada pbyRecord\_IOL\_Data y cambiar el tipo POINTER TO BYTE al tipo Array [¿..?] of byte y en la misma entrada, asignar ADR(), de la forma ADR(var).  
 En la entrada Slave deberemos poner el mismo nombre que el esclavo modbus (el último bus de campo añadido y por defecto, Modbus\_TCP\_Slave).



Con estas entradas en el FB, ya está listo para funcionar. Únicamente debemos asignar los valores correctos a las variables.



| Expresión   | Tipo de datos         | Valor |
|-------------|-----------------------|-------|
| IOL_CALL_0  | IOLC_MB.IOL_CALL      |       |
| xReq        | BOOL                  | FALSE |
| xSlot       | USINT                 | 0     |
| xCap        | INT                   | 0     |
| xRD_WR      | BOOL                  | FALSE |
| xPort       | USINT                 | 0     |
| xIndex      | UINT                  | 0     |
| xSubIndex   | USINT                 | 0     |
| xDatos      | ARRAY [0..250] OF ... |       |
| xLen        | INT                   | 0     |
| xBusy       | BOOL                  | FALSE |
| xError      | BOOL                  | FALSE |
| xStatus     | DWORD                 | 0     |
| xiOL_Status | DWORD                 | 0     |
| xRD_Len     | INT                   | 0     |

The ladder logic diagram shows the IOL\_CALL\_0 function block with the following connections:

- xReq (Bool) is connected to the xReq input.
- xSlot (USINT, 0) is connected to the usiSlot input.
- xCap (INT, 0) is connected to the iIndexCap input.
- xRD\_WR (Bool) is connected to the xRD\_WR input.
- xPort (USINT, 0) is connected to the usiEntityPort input.
- xIndex (UINT, 0) is connected to the uiIOL\_Index input.
- xSubIndex (USINT, 0) is connected to the usiIOL\_Subindex input.
- adr(xDatos) is connected to the pbyRecord\_IOL\_Data input.
- xLen (INT, 0) is connected to the iLen input.
- Modbus\_TCP\_Slave is connected to the slave input.

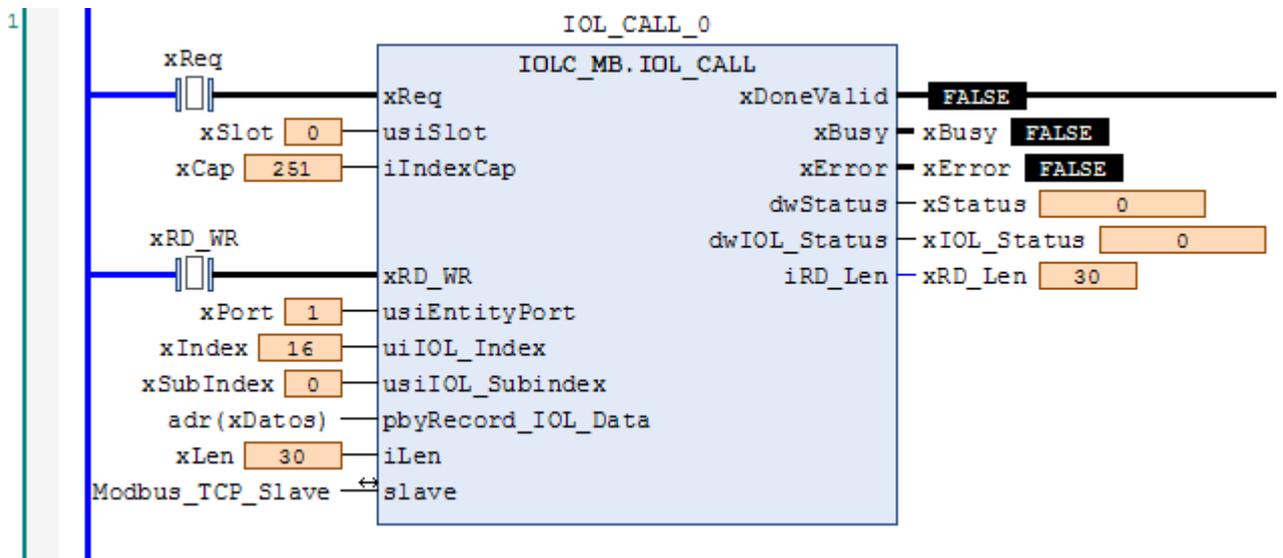
Outputs and their values:

- xDoneValid: FALSE
- xBusy: FALSE
- xError: FALSE
- dwStatus: 0
- dwIOL\_Status: xiOL\_Status (0)
- iRD\_Len: xRD\_Len (0)

Podemos observar en el menú izquierdo como toda la configuración hardware está verde, es decir, que comunica correctamente.

El FB lo tenemos en estado inicial de reposo y antes de activar xReq debemos configurar los valores.

Cuando hemos asignado estos valores, los cargamos sobre el FB y podemos activar la entrada xReq, la cual realizará la petición / activación del FB.



A diferencia del software de Siemens, en este FB es necesario asignar un valor a la variable de entrada xLen para asignar qué cantidad de Bytes queremos leer.



Al activar la entrada xReq, ejecutará la orden de lectura (ya que xRD\_WR está = OFF)

PLC\_PRG x

Device.Application.PLC\_PRG

| Expresión  | Tipo de datos         | Valor |
|------------|-----------------------|-------|
| xDatos     | ARRAY [0..250] OF ... |       |
| xDatos[0]  | BYTE                  | 66    |
| xDatos[1]  | BYTE                  | 97    |
| xDatos[2]  | BYTE                  | 110   |
| xDatos[3]  | BYTE                  | 110   |
| xDatos[4]  | BYTE                  | 101   |
| xDatos[5]  | BYTE                  | 114   |
| xDatos[6]  | BYTE                  | 32    |
| xDatos[7]  | BYTE                  | 69    |
| xDatos[8]  | BYTE                  | 110   |
| xDatos[9]  | BYTE                  | 103   |
| xDatos[10] | BYTE                  | 105   |
| xDatos[11] | BYTE                  | 110   |
| xDatos[12] | BYTE                  | 101   |
| xDatos[13] | BYTE                  | 101   |
| xDatos[14] | BYTE                  | 114   |
| xDatos[15] | BYTE                  | 105   |
| xDatos[16] | BYTE                  | 110   |
| xDatos[17] | BYTE                  | 103   |
| xDatos[18] | BYTE                  | 32    |
| xDatos[19] | BYTE                  | 67    |
| xDatos[20] | BYTE                  | 111   |
| xDatos[21] | BYTE                  | 114   |
| xDatos[22] | BYTE                  | 112   |
| xDatos[23] | BYTE                  | 111   |
| xDatos[24] | BYTE                  | 114   |
| xDatos[25] | BYTE                  | 97    |
| xDatos[26] | BYTE                  | 116   |
| xDatos[27] | BYTE                  | 105   |

Y nos devolverá en la variable tipo array tantos bytes como visualicemos en la salida xRD\_Len.



DELEGACIONES:

**Cataluña:**

Tel. 932 982 000  
elion@elion.es

**Centro:**

Tel. 913 835 709  
elionmad@elion.es

**Sur:**

Tel. 955 943 441  
egiraldez@elion.es

**Norte:**

Tel. 943 217 200  
imorales@elion.es

**Servicio Asistencia Técnica**

Farell, 5  
0814 Barcelona  
servicio.tecnico@elion.es



**ELION, S.A.**

Farell, 5  
08014 Barcelona  
Tel. 932 982 000  
Fax 934 311 800  
elion@elion.es  
www.elion.es

