

Trabajo Fin de Grado

Ingeniería en Tecnologías Industriales

Control de una maqueta de ascensores usando PLC y SCADA Citect

Autor: Óscar Cruz Collado

Tutor: Francisco Salas

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021



Proyecto Fin de Grado
Ingeniería en Tecnologías Industriales

Control de una maqueta de ascensores usando PLC y SCADA Citect

Autor:
Óscar Cruz Collado

Tutor:
Francisco Salas
Profesor titular

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2021

Trabajo Fin de Grado: Control de una maqueta de ascensores usando PLC y SCADA Citect

Autor: Óscar Cruz Collado

Tutor: Francisco Salas

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

A Francisco, Susana y Sergio

A Sira, Ana y los que ya no están

A mi equipo, Juan Manuel y Paco

A Ángela

Agradecimientos

Soy un afortunado por tener tantas personas a las que agradecerles el ser quien soy hoy. Quiero dar las gracias a todas aquellas personas que durante mi vida han estado conmigo, en las buenas y en las malas, aunque quisiera hacer hincapié en algunas de ellas.

Gracias a mi familia, por su esfuerzo y su paciencia. Por creer en mí más que yo mismo y por hacerme sentir siempre “su caballito ganador”. Por motivarme a ser mejor cada día. Si hoy estoy escribiendo estas líneas es gracias a ellos.

A María, Ana y Manu, por siempre estar y por hacerme crecer. Por aplaudirme mis aciertos y advertirme de mis errores. Por hacerme cambiar siempre para mejor.

A Ángel, por todo lo que ha aportado en mi educación, pero sobre todo a nivel personal. Por haberme planteado retos que me han hecho crecer. Por darme su confianza.

A Ángela, por escucharme, por no solo apoyarme en mis proyectos, sino hacerlos también suyos y disfrutar de ellos. Por siempre estar orgullosa de mí y hacer que yo mismo lo esté.

A mis “Industriales” y “SQDR”, por hacer estos años infinitamente más divertidos.

Sin embargo, la mención especial es para mi “Equipo de remo”, Paco y Juanma. Muchísimas gracias por estos años, por enseñarme que es posible trabajar duro y divertirse al mismo tiempo. Por apoyarnos mutuamente cuando las cosas se ponían difíciles y por aportarme cada uno un granito de vuestra personalidad, que son tan diferentes, pero a la vez tan complementarias y maravillosas. No me imagino esta etapa sin vosotros.

Óscar Cruz Collado

Alumno de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla

Sevilla, 2021

Resumen

En este proyecto se pretende realizar el control y la supervisión de una planta formada por una maqueta de ascensores y un autómata programable M340 de la marca Schneider, perteneciente al Departamento de Ingeniería de Sistemas y automática de la Escuela.

En primer lugar, se programará el autómata con el software 'Ecostruxure Control Expert'. Dicho software permitirá, entre otros, programar en tres lenguajes de programación ampliamente estudiados durante los años de grado: LD, ST y SFC.

Posteriormente, se procederá a la conexión del PLC al sistema de supervisión SCADA. Para ello, se hará uso de un router que le asignará una dirección IP al autómata a través de su puerto Ethernet, permitiendo así la conexión con un servidor OPC UA. Una vez conectado al servidor, se configurará el programa 'Citect Studio 2018 R2' como cliente de él. Dicho cliente podrá tanto leer variables del sistema como publicarlas, pudiendo tanto recibir información como enviarla a la planta.

Finalmente, se procederá al diseño de un sistema SCADA completo, en el que se podrá visualizar el estado de la planta en tiempo real, así como las plantas pedidas y gestionar sus modos de funcionamiento. Además, se implementa la creación de 'logs', que informarán en formato .txt de aquellas acciones ejercidas sobre el sistema por pulsación de botones e informes, un documento .rft en el que se hace un historial de las plantas en las que ha estado cada ascensor cada 10 segundos.

Además de esto, se incluye en la presente memoria documentación referente al conexionado de la planta, así como un manual detallado tanto de las conexiones PLC – SCADA como del diseño del SCADA.

Abstract

This project intends to build the control and supervision of an Automation and System Engineering department's plant consisting of a lift model and a programmable controller PLC M340, from Schneider Electronics.

First, the plant's PLC will be programmed by the software 'Ecostruxure Control Expert'. This software allows to program in, among many others, three different programming languages, which have been widely studied during these years of the degree: LD, ST and SFC.

After that, the connection between the PLC and the SCADA system will be made. To do that, a router will be plugged to the PLC processor's Ethernet port, so that it can have an IP direction assigned. Once this is done, it will be possible to connect it to the OPC UA Server. The last step will be the configuration the program 'Citect Studio 2018 R2' as a server client. This client will be able to read system variables as well as publish them, so it will both receive and send information to the system.

Finally, the SCADA system will be designed in the program already mentioned. This program will allow the user to visualise the plant state in real time, as well as the floors that have been called and manage the operating modes. Moreover, logs and reports are implemented, which means that two files will be created, the first one to monitor each button press, in txt format, and the second one to create a historical record of the floors where the lifts have been. This will be made every ten seconds, in rft format.

Apart from that, documentation concerning the lift model connection to the PLC is included in this memory, as well as a detailed manual both for the PLC – SCADA connection and the SCADA design.

Índice

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xiv
Índice de Tablas	xvii
Índice de Figuras	xix
1 Introducción	1
1.1 SCADA	1
1.2 Objetivos	2
1.3 Estructura del trabajo	2
2 OPC UA	3
2.1 OPC	3
2.2 OPC UA	4
2.2.1 Variedad de sistemas operativos	4
2.2.2 Seguridad	4
2.2.3 Otros	4
3 Sistemas de Elevación	5
3.1 Historia	5
3.2 Estructura	6
3.2.1 Sala de máquinas	6
3.2.2 Hueco del ascensor	8
3.2.3 Foso	10
3.3 Funcionamiento	11
3.3.1 Maniobra universal	11
3.3.2 Maniobra colectiva de descenso	11
3.3.3 Maniobra colectiva de subida y bajada	11
4 Descripción de la Planta	13
4.1 Maqueta del ascensor	13
4.1.1 Descripción física	14
4.1.2 Entradas y salidas	14
4.2 Autómata programable	16
4.2.1 Bastidor BMX XBP 0400	16
4.2.2 Módulo de alimentación BMX CPS 2000	17
4.2.3 Procesador BMX P34 2020	17
4.2.4 Módulo BMX DDI 3202K	17
4.2.5 Módulo BMX DDM 16022	18

4.3	<i>Otros elementos</i>	19
4.4	<i>Conexión PLC – Maqueta</i>	20
4.4.1	Conectores DB	20
4.4.2	Conexionado de entradas a módulos de autómatas	21
4.4.3	Conexionado de salidas a relés	23
4.4.4	Conexionado de relés a módulos de autómatas	25
5	Programas Utilizados	29
5.1	<i>Ecostruxure Control Expert</i>	29
5.1.1	Explorador de proyectos	29
5.1.2	Conexión PC - PLC	33
5.2	<i>OPC UA Server Expert</i>	35
5.2.1	Programas	35
5.3	<i>Citect Studio</i>	39
5.3.1	Introducción	39
5.3.2	Especificaciones	40
5.3.3	Programa	41
6	Programación	47
6.1	<i>Modos de funcionamiento</i>	47
6.1.1	Modo normal	48
6.1.2	Modo emergencia	48
6.2	<i>Algoritmos</i>	49
6.2.1	Modo normal	49
6.2.2	Modo emergencia	50
7	Conexión PLC – OPC UA - SCADA	51
7.1	<i>Asignación de dirección IP al PLC</i>	51
7.1.1	Búsqueda de conexión IP	51
7.1.2	Acceso a router	52
7.1.3	Configuración Ecostruxure	54
7.2	<i>Configuración servidor OPC UA</i>	58
7.2.1	Exportación de variables Ecostruxure	58
7.2.2	Configuración OPC UA	59
7.3	<i>OPC UA Server Expert</i>	61
7.3.1	Dependencia de OPC UA con una dirección IP	61
7.3.2	Dirección URL de servidor OPC UA	63
7.4	<i>Citect Studio 2018 R2</i>	64
7.4.1	Creación de un nuevo proyecto	64
7.4.2	Configuración dispositivo E/S	65
7.4.3	Configuración SCADA – OPC UA	70
7.4.4	Asistente de configuración	73
8	Diseño Visualización SCADA	77
8.1	<i>Funciones del SCADA</i>	77
8.2	<i>Diseño del SCADA</i>	79
8.2.1	Elementos estáticos	80
8.2.2	Elementos dinámicos	81
8.3	<i>Implementación de 'logs' e informes</i>	88
8.3.1	Logs	88
8.3.2	Informes	90

9 Conclusiones	93
9.1 <i>Conclusiones</i>	93
9.1.1 Programación de la planta	93
9.1.2 Conexión PLC – OPC UA – SCADA	93
9.1.3 Diseño y funciones SCADA	94
9.2 <i>Líneas de mejora</i>	94
9.2.1 Planta de ascensores	94
9.2.2 Conexiones	94
9.2.3 SCADA	94
Anexo A – Código de programación	95
Referencias	115

ÍNDICE DE TABLAS

Tabla 4.1 Relación de entradas del sistema	15
Tabla 4.2 Relación de salidas del sistema	15
Tabla 4.3 Correlación cables-conectores del primer DB 25	20
Tabla 4.4 Correlación cables-conectores del segundo DB 25	21
Tabla 4.5 Conexión primer DB 25 a módulo BMX DDI 3202K	22
Tabla 4.6 Conexión segundo DB 25 a módulo BMX DDI 3202K	22
Tabla 4.7 Conexión de motores DB 25 a relés	24
Tabla 4.8 Conexión de leds DB 25 a relés	25
Tabla 4.9 Conexión de relés a módulo BMX DDM 16022	25
Tabla 5.1 Requisitos Ecostruxure Control Expert	29
Tabla 5.2 Requisitos OPC UA Server Expert	35
Tabla 5.3 Requisitos Citect Studio	40

ÍNDICE DE FIGURAS

Figura 1.1 Pirámide de la automatización	1
Figura 2.1 Esquema OPC	3
Figura 3.1 Primera demostración pública de ascensor (Nueva York, 1853)	5
Figura 3.2 Grupo tractor	6
Figura 3.3 Limitador de velocidad	7
Figura 3.4 Cuadro de maniobra	7
Figura 3.5 Contrapeso de un ascensor	8
Figura 3.6 Esquema de las partes de un cable	9
Figura 3.7 Sistema de seguridad paracaídas	9
Figura 3.8 Amortiguador de aceite hidráulico	10
Figura 4.1 Planta a usar en este proyecto	13
Figura 4.2 Autómata usado en el proyecto	16
Figura 4.3 Diagrama de conexión BMX DDI 3202K	18
Figura 4.4 Diagrama de conexión BMX DDM 16022	18
Figura 4.5 Placa IFM DEE 25 F de M-Jay Electronics ltd	19
Figura 4.6 Esquema de conexión de motores	23
Figura 4.7 Esquema de conexión de leds	24
Figura 4.8 Relés, conectores DB y botones	26
Figura 4.9 Conexión Maqueta - PLC	27
Figura 4.10 Esquema de conexiones Maqueta - PLC	28
Figura 5.1 Explorador de proyectos Ecostruxure	30
Figura 5.2 Configuración de módulos del autómata	30
Figura 5.3 Ejemplo de variables elementales del sistema	31
Figura 5.4 Ejemplo de creación de sección	31
Figura 5.5 Condición de activación de una sección	32
Figura 5.6 Tabla de animación	33
Figura 5.7 Pantalla de operador	33
Figura 5.8 Conexión PC - Autómata	33
Figura 5.9 Transferencia de proyecto PC - Autómata	34
Figura 5.10 Transferencia de proyecto a PLC	34

Figura 5.11 Indicadores de activación de secciones	34
Figura 5.12 Programas OPC UA Server Expert	35
Figura 5.13 Diagnostics OPC UA	36
Figura 5.14 Network OPC UA	36
Figura 5.15 OPC UA Server Expert Sample Client	37
Figura 5.16 OPC UA Server Expert Sample Client II	37
Figura 5.17 OPC UA Server Expert Sample Client III	38
Figura 5.18 OPC UA Server Expert Sample Client IV	38
Figura 5.19 Diagrama de funcionamiento SCADA	39
Figura 5.20 Logo Citect Studio	40
Figura 5.21 Menú de ventanas Citect Studio	41
Figura 5.22 Ventana de Proyectos Citect Studio	42
Figura 5.23 Ventana de Topología Citect Studio	42
Figura 5.24 Listado de variables Citect Studio	42
Figura 5.25 Ventana de Páginas Citect Studio	43
Figura 5.26 Ventana de Seguridad Citect Studio	43
Figura 5.27 Compilación de variables Citect Studio	44
Figura 5.28 Importación de variables Citect Studio	44
Figura 5.29 Gestor de funcionamiento Citect Studio	44
Figura 5.30 Page Menu Citect Studio	45
Figura 5.31 Elección de página Citect Studio	45
Figura 6.1 Esquema guía GEMMA	47
Figura 6.2 Modos de funcionamiento	48
Figura 6.3 Funcionamiento modo automático	50
Figura 7.1 Configuración IP ordenador	51
Figura 7.2 Interior del router	52
Figura 7.3 Direcciones IP asignadas manualmente	52
Figura 7.4 Añadir nueva dirección IP manualmente	53
Figura 7.5 Dirección MAC automática	53
Figura 7.6 Nueva red Ethernet	54
Figura 7.7 Nueva red Ethernet 2	54
Figura 7.8 Pantalla de configuración Ethernet	55
Figura 7.9 Cambio de familia de modelo	55
Figura 7.10 Advertencia cambio de familia	55
Figura 7.11 Configuración IP	56
Figura 7.12 Validar cambios	56
Figura 7.13 Puerto Ethernet del bus PLC	56
Figura 7.14 Configuración canal Ethernet	57
Figura 7.15 Configuración canal Ethernet II	57
Figura 7.16 Exportación seleccionada	58

Figura 7.17 Formato exportación de variables	58
Figura 7.18 Visualización de la configuración del OPC UA	59
Figura 7.19 Configuración de dirección IP OPC UA	59
Figura 7.20 Importación tabla de variables	60
Figura 7.21 Configuración de seguridad OPC UA	60
Figura 7.22 Error en OPC UA	61
Figura 7.23 Reparación OPC UA Server Expert	61
Figura 7.24 Reparación OPC UA Server Expert II	62
Figura 7.25 Reparación OPC UA Server Expert III	62
Figura 7.26 Reparación OPC UA Server Expert IV	63
Figura 7.27 URL OPC UA	63
Figura 7.28 Creación nuevo proyecto SCADA	64
Figura 7.29 Creación nuevo proyecto SCADA II	64
Figura 7.30 Creación nuevo proyecto SCADA III	65
Figura 7.31 Topología SCADA	65
Figura 7.32 Configuración dispositivo E/S	65
Figura 7.33 Configuración dispositivo E/S II	66
Figura 7.34 Configuración dispositivo E/S III	66
Figura 7.35 Configuración dispositivo E/S IV	67
Figura 7.36 Configuración dispositivo E/S V	67
Figura 7.37 Configuración dispositivo E/S VI	68
Figura 7.38 Configuración dispositivo E/S VII	68
Figura 7.39 Configuración dispositivo E/S VIII	69
Figura 7.40 Configuración dispositivo E/S IIX	69
Figura 7.41 Variables importadas del proyecto	70
Figura 7.42 Configurador OPC UA	70
Figura 7.43 Configurador OPC UA II	71
Figura 7.44 Configurador OPC UA III	71
Figura 7.45 Configurador OPC UA IV	72
Figura 7.46 Configurador OPC UA V	72
Figura 7.47 Asistente de configuración Citect	73
Figura 7.48 Asistente de configuración Citect II	73
Figura 7.49 Asistente de configuración Citect III	74
Figura 7.50 Asistente de configuración Citect IV	74
Figura 7.51 Asistente de configuración Citect V	75
Figura 7.52 Asistente de configuración Citect VI	75
Figura 7.53 Asistente de configuración Citect VII	76
Figura 7.54 Asistente de configuración Citect VIII	76
Figura 8.1 SCADA Modo automático	77

Figura 8.2 SCADA Modo manual	78
Figura 8.3 SCADA Modo Reinicio	78
Figura 8.4 Editor gráfico de Citect	79
Figura 8.5 Menú de objetos Citect	79
Figura 8.6 Propiedades de Rectángulo	80
Figura 8.7 Propiedades de Texto	81
Figura 8.8 Propiedades de Grupo de Símbolos	82
Figura 8.9 Propiedades de Grupo de Símbolos II	83
Figura 8.10 Motor con indicador de estado	83
Figura 8.11 Propiedades de Polígono	84
Figura 8.12 Cabinas de ascensores	84
Figura 8.13 Propiedades de Rectángulo – Visibilidad	85
Figura 8.14 Propiedades de Texto - Expresión numérica	86
Figura 8.15 Propiedades de Botón	86
Figura 8.16 Propiedades de Botón - Abajo	87
Figura 8.17 Propiedades de Botón - Arriba	87
Figura 8.18 Creación de dispositivo	88
Figura 8.19 Propiedades de Botón - Configuración Log	89
Figura 8.20 Propiedades de Botón - Configuración Log II	89
Figura 8.21 Ejemplo Log	90
Figura 8.22 Creación de informe	90
Figura 8.23 Programación de formato de informe	91
Figura 8.24 Ejemplo informe	92

1 INTRODUCCIÓN

A lo largo de los siglos, el avance en la tecnología ha permitido a la humanidad ser cada vez más productiva. Esto ha traído consigo una mejora de la economía, que ha pasado a ser global, y por tanto una mejor calidad de vida. La primera revolución industrial, en la Inglaterra del siglo XVIII, acarrió un cambio en la economía, que pasó de ser rural a urbana, mecanizada e industrializada. Desde entonces, la industria siguió desarrollándose con la mejora de las telecomunicaciones y la automatización de procesos, dando lugar a una segunda y tercera revolución industrial.

Actualmente, ya se habla de una cuarta revolución, también llamada la ‘Industria 4.0’, cuya filosofía es la transformación de la empresa en organizaciones inteligentes, apoyándose en tecnologías como la Inteligencia Artificial, el Big Data y las conexiones IoT (Internet Of Things), que busca crear una macrored de dispositivos conectados entre sí que puedan compartir datos y beneficiarse de ellos.

1.1 SCADA

Al estar en la época del Big Data y del IoT, es necesario disponer de un software que permita aprovecharse de estas tecnologías. En el caso de la automatización, esto, en parte, se materializa en los sistemas SCADA (“*Supervisory Control and Data Acquisition*”), un sistema que es capaz de controlar remotamente y en tiempo real una instalación, integrando datos recogidos de diferentes autómatas (PLC’s), sensores y equipos mediante diferentes protocolos en un mismo lugar. Además, es capaz de compilar un histórico de dichos valores y crear informes y bases de datos que posteriormente podrán ser estudiadas para sacar un mayor rendimiento.

Así, en la web de *Omnicon*, una empresa de ingeniería a nivel mundial, podemos encontrar la llamada “Pirámide de la automatización industrial. En ella podemos observar que el SCADA estaría en el nivel número dos, en un nivel de control y supervisión, por encima del PLC, que es el encargado de gestionar el control de sensores y actuadores.



Figura 1.1 Pirámide de la automatización

Por ello, se pretende en este trabajo hacer uso de algunas de estas tecnologías que tanto están revolucionando la industria en estos últimos años.

1.2 Objetivos

El objetivo de este trabajo es la implementación de un sistema SCADA en una planta real, concretamente en una maqueta de un sistema de dos ascensores disponible en el departamento de Ingeniería de Sistemas y Automática.

Para ello, el primer paso será utilizar el programa *'Ecostruxure Control Expert'*. Esta aplicación nos permitirá programar el autómatas programable M340 de la marca Schneider conectado al sistema.

Seguidamente, utilizaremos el programa *'OPC UA Server'*, el cual nos permitirá crear un servidor de conexión entre el autómatas y el SCADA. La tendencia del mercado nos hizo decidimos por utilizar un protocolo OPC UA y no un OPC convencional, pues está demostrado las ventajas que tiene el primero frente al segundo. Estas ventajas se analizarán más adelante.

Finalmente, crearemos el SCADA con el programa Citect Studio versión 2018. Se pretende que este Trabajo de Fin de Grado sirva como manual para futuros estudiantes que deseen o necesiten utilizar estas tecnologías.

1.3 Estructura del trabajo

Este Trabajo Fin de Grado se encuentra estructurado de la siguiente manera:

En el Capítulo 2, titulado *'OPC UA'*, se explicará la necesidad que tuvo la industria de un protocolo abierto y compatible con distintos dispositivos de distintas empresas. Además, se verán las carencias que tuvo el OPC DA y como estas fueron solucionadas con la introducción del sistema OPC UA.

En el Capítulo 3, *'Sistemas de elevación'*, se verá una breve introducción de la historia de los ascensores, pues será la planta que se utilizará en este proyecto. Además, se expondrán las partes que los conforman y con qué métodos estos suelen ser programados.

En el Capítulo 4, llamado *'Descripción de la planta'*, se hará un resumen de cómo está configurada la maqueta con el autómatas y se enumerarán las entradas y salidas del sistema.

El Capítulo 5, el cual se titula *'Programas utilizados'*, hará una breve introducción a la evolución del muy conocido programa *'Unity Pro XL'*, *'Ecostruxure Control Expert'*, usado para programar autómatas de la marca Schneider. Además, se introducirán los softwares *'Citect Studio'* y *'OPC UA Server Expert'*, utilizados para el diseño del SCADA y la comunicación PLC – SCADA, respectivamente.

El Capítulo 6, llamado *'Programación'*, dará las pautas generales seguidas a la hora de idear el programa que regirá el control y el movimiento de los ascensores, aunque el código podremos encontrarlos al final de esta memoria, en los *'Anexos'*.

El Capítulo 7, titulado *'Conexión PLC – OPC UA – SCADA'*, sirve como una detallada guía del proceso de conexión entre los distintos elementos que conforman este proyecto, que ya habrán sido detallados en capítulos anteriores.

En el Capítulo 8, *'Diseño SCADA'*, se darán algunas pautas para el diseño y posterior funcionamiento del SCADA con el programa mostrado en el Capítulo 6, Citect Studio 2018 R2.

Finalmente, en el Capítulo 9, titulado *'Conclusiones y líneas de mejora'*, se hará una recopilación de los resultados obtenidos, se añadirán videos demostrativos del funcionamiento y se dejarán algunas posibles mejoras de este trabajo para una hipotética ampliación.

2 OPC UA

Durante los años 80 y 90, la industria tenía la necesidad de automatizar sus procesos. Uno de los mayores retos era comunicar la gran cantidad de equipos que formaban las plantas. Además, cada vez que se adquiría uno nuevo, este solía traer un nuevo protocolo de comunicación, nuevos drivers a instalar y mantener. Esto daba lugar a que los ordenadores de las plantas industriales tuvieran cada vez más y más protocolos instalados y que los propios ingenieros tuvieran que aprender continuamente distintas tecnologías.

Otro problema lo tenían los propios ingenieros de mantenimiento, que se encontraban con una cantidad de peticiones de soporte abiertas de distintas versiones y tecnologías. Por ello, se planteó un nuevo problema en el mundo de la automatización: crear un estándar de comunicación abierto e independiente.

2.1 OPC

Como solución a todos estos problemas, nació el protocolo OPC (*Open Protocol Communication*), que es capaz de unificar todos los protocolos anteriores en un solo protocolo de comunicación abierto, facilitando muchísimo el trabajo de los ingenieros.

Este sistema se basa en una arquitectura cliente – servidor; es decir, debe haber programas independientes: una fuente de datos que publicará información en un servidor y un cliente, que consulte dichos datos. En el caso de este proyecto:

- Fuente de datos: PLC, que gestionará la información de sensores y actuadores y la publicará.
- Cliente: SCADA, que utilizará esos datos.
- Servidor: plataforma de comunicación basada en Windows.

Un esquema de funcionamiento del OPC sería:

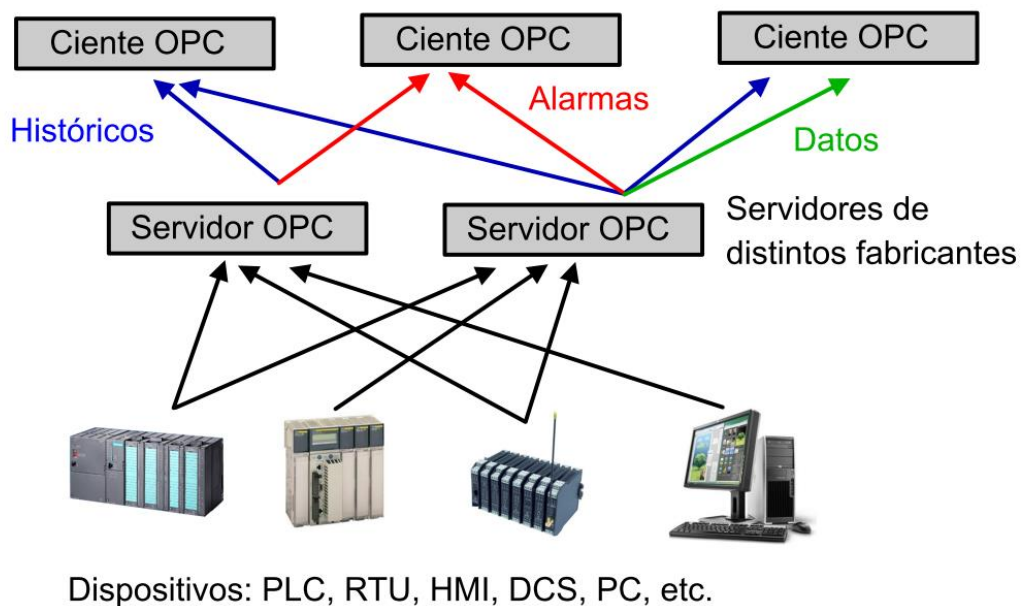


Figura 2.1 Esquema OPC

Algunos de sus beneficios son:

- Tecnología escalable, ya que el mismo software podrá usarse en el futuro con nuevos dispositivos sin coste alguno.
- Tecnología independiente de los fabricantes, por lo que no es necesario que todos los equipos sean del mismo proveedor.
- Menor mantenimiento, pues se tiene un protocolo común, y no un protocolo propio por cada equipo y/o fabricante.

Algunas desventajas son:

- Protocolo dependiente de Windows, por lo que es incompatible con cualquier otro sistema operativo.
- Seguridad: es importante, en la era del IoT (Internet of Things), contar con una seguridad robusta y varias capas de seguridad nativa. Sin embargo, en el OPC estas son limitadas.

2.2 OPC UA

El servidor OPC UA (OPC Unified Architecture) es una evolución del servicio OPC que busca solucionar las desventajas del OPC clásico. Algunas diferencias entre uno y otro son:

2.2.1 Variedad de sistemas operativos

El OPC Clásico utiliza el sistema COM / DCOM para la transmisión de archivos, por lo que es, como ya se ha dicho, un protocolo totalmente dependiente de Windows. OPC UA soluciona este problema y abre el protocolo a otros sistemas operativos como Android o Linux.

2.2.2 Seguridad

Para una empresa es muy importante asegurarse de la privacidad de sus datos. Por ello, OPC UA encripta las comunicaciones mediante certificados de seguridad que las distintas partes deben conocer antes de comunicarse.

2.2.3 Otros

Además, tiene otras ventajas como la capacidad del servidor para residir dentro del mismo PLC, ser más sencilla su conexión con el Firewall (solo es necesario un puerto para establecer las comunicaciones) y tener una mejor gestión de cada tipo de fuente de datos.

3 SISTEMAS DE ELEVACIÓN

La implementación tanto del sistema SCADA como del protocolo de comunicación OPC UA se hará sobre una maqueta de dos ascensores desarrollada en el marco de un proyecto final de carrera realizado por Francisco Rodríguez Rubio en 1981 y actualizado por Daniel Paredes García en el año 2017.

Por ser la base de este trabajo un sistema de elevación, se dedicará este capítulo a hacer una descripción tanto de la historia, como de la estructura y del funcionamiento de este tipo de planta.

3.1 Historia

El primer ascensor de la historia se le atribuye al matemático Arquímedes, en el año 236 a.C. Según escritos de la época, se trataba de un ascensor compuesto por cabinas que colgaban de una cuerda de cáñamo, accionadas a mano por personas o animales, instaladas en el monasterio de Sinaí. Con el desarrollo del ascensor, llegó a utilizarse incluso en el propio coliseo para elevar a gladiadores y animales.

Sin embargo, con el paso de los años, la humanidad tendió a reestructurarse en torno a ciudades. Esto llevó a un aumento en el precio del suelo y por tanto una necesidad de su mejor aprovechamiento, dando paso a altos edificios, por lo que un sistema de elevación seguro se hizo más necesario que nunca.

Los ascensores medievales y antiguos funcionaron con un sistema de contrapesos, base del sistema de grúas actual. Sin embargo, el ascensor no evoluciona hasta la creación de aparatos con transmisión a tornillo.

La revolución del ascensor no se da hasta 1852, cuando Elisha Otis, fundador de la empresa que actualmente lleva su nombre, inventa el primer freno de seguridad para ascensores. En ese momento, los ascensores con elevadores de vapor ya existían, pero se desplomaban con facilidad, por lo que no generaba confianza.

Otis montó el primer montacargas en un edificio pequeño en Yonkers el 20 de septiembre de 1853, aunque los contratos siguieron siendo escasos. Esto cambió cuando hizo una gran demostración de su nuevo sistema en el Palacio de Cristal de Nueva York. En esa demostración, mientras él estaba subido a gran altura, mandó cortar el único cable que sostenía el elevador. Este solo se movió unos centímetros y se paró, provocando, a partir de entonces, una gran acogida.



Figura 3.1 Primera demostración pública de ascensor (Nueva York, 1853)

Más adelante, se inventaría el ascensor hidráulico y el ascensor electromecánico, combinando engranajes y electricidad. Esto impulsó la creación de rascacielos en grandes ciudades, lo que repercutía en su economía, por lo que estos dispositivos siguieron desarrollándose hasta llegar a lo que son hoy en día, un sistema instalado en todos los edificios de nueva construcción.

3.2 Estructura

Los ascensores electromecánicos con sala de máquinas pueden dividirse en tres partes: la sala de máquinas, el hueco del ascensor y el foso. A continuación, se entrará en detalle en las siguientes partes y en los elementos que forman parte de ellas.

3.2.1 Sala de máquinas

La sala de máquinas es una habitación donde se encuentran el grupo tractor, el limitador de velocidad y el cuadro de maniobra. Esta sala es de acceso restringido, por lo que solo podrán entrar las personas responsables del mantenimiento. Suele estar encima del hueco del ascensor y no debe colocarse junto a locales habitables.

3.2.1.1 Grupo tractor

El grupo tractor es el encargado de transformar la energía eléctrica en una energía mecánica en forma de rotación, que posteriormente se transformará en lineal. El Grupo tractor está formado por algunos elementos.

- Motor eléctrico: transforma la energía eléctrica en mecánica gracias a la acción de campos magnéticos generados por bobinas y que hace girar un tornillo sin fin.
- Freno: debe activarse automáticamente en caso de pérdida de energía en los circuitos de control. Debe ser capaz de frenar de forma segura el ascensor con una carga equivalente al 125% de la carga nominal.
- Reductor: formado por un tornillo sin fin y una serie de engranajes, reduce la velocidad del motor para ajustarla a la que deseamos en el ascensor.
- Polea de tracción: en un ascensor, la polea superior será la encargada tanto de subir como de bajar el ascensor, por lo que será siempre de tracción. Por ello, se debe tener mucho cuidado con su diseño, pues, además de soportar los esfuerzos transmitidos por el cable, debe ser capaz de transmitir la tracción a este por adherencia.
- Volante de inercia: asegura que el ascensor quede equilibrado en cada piso cuando el motor es de una única velocidad. Sin embargo, estos son cada vez menos usados, pues los motores tienden a ser de más de una velocidad.

Un grupo tractor en conjunto se vería así:

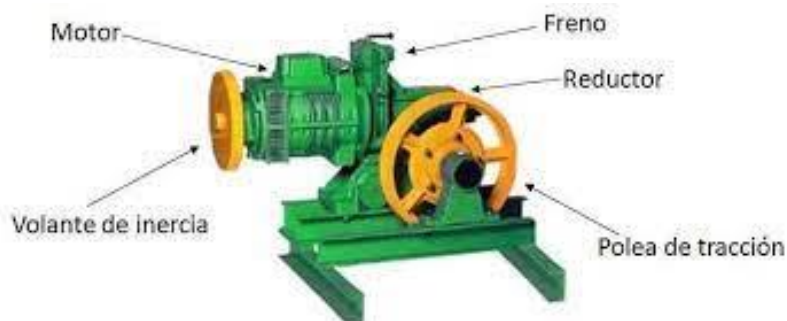


Figura 3.2 Grupo tractor

3.2.1.2 Limitador de velocidad

Sistema de seguridad formado por dos poleas acanaladas, una en la sala de máquinas (limitadora de velocidad) y otra en el foso (tensora del limitador) que bloquean el cable cuando se produce una sobre velocidad de un valor establecido.

El cable bloqueado va unido a la palanca que activa el paracaídas del ascensor, por lo que podría decirse que es un sistema de seguridad que activa otro sistema de seguridad.



Figura 3.3 Limitador de velocidad

3.2.1.3 Cuadro de maniobra

Es el elemento que controla todas y cada una de las acciones que realiza el ascensor. Es el cerebro de este y suele o puede estar compuesto por un autómata programable, como será el caso en nuestro proyecto.

Sobre este cuadro de maniobra se carga el programa informático que gobernará el movimiento del sistema. Normalmente, este programa está inspirado en algunos tipos de maniobra que se explicarán al final de este capítulo, como son la maniobra universal o colectiva en subida y/o bajada.



Figura 3.4 Cuadro de maniobra

3.2.2 Hueco del ascensor

Es el espacio disponible exclusivamente para el movimiento del ascensor y el contrapeso, no pudiendo ser utilizado por otra instalación de ningún tipo. Los elementos que conforman este elemento son la cabina las guías, el contrapeso, los cables y el paracaídas.

3.2.2.1 Cabina

Elemento portante del ascensor formado por la caja, que es el habitáculo en el cual viajan las personas mientras se realiza el movimiento del ascensor, y el bastidor, que es la estructura de metal que la sostiene y que se mueve solidario a ella.

El bastidor es muy importante, ya que, además de sostener la cabina, es el lugar al que se fijan tanto los cables se la suspensión como el paracaídas.

3.2.2.2 Guía de fijación de la cabina

Conducen a la cabina en su trayectoria y sirven como apoyo en caso de rotura de cables, por lo que deben ser resistente acorde al peso que van a soportar.

3.2.2.3 Contrapeso

Elemento utilizado para equilibrar cargas y por tanto facilitar al motor el movimiento. Su masa debe ser igual a la de la cabina más la mitad de la máxima carga autorizada, por lo que el motor no tiene que trabajar a máximo rendimiento ni con la cabina en carga máxima. También disminuye la potencia necesaria y por tanto el consumo.

El contrapeso también tiene sus propias guías que, al igual que con la cabina, sirven tanto para conducir la trayectoria como como elemento de protección.

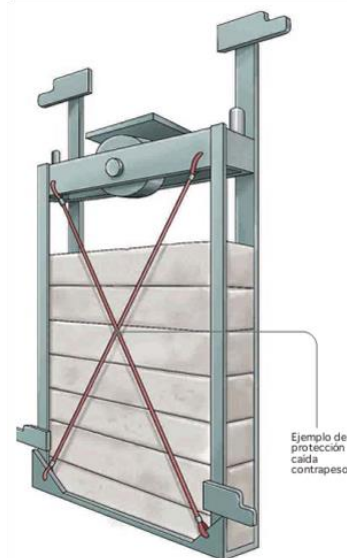


Figura 3.5 Contrapeso de un ascensor

3.2.2.4 Cables

Los cables utilizados en los sistemas de elevación actual constan de un alma impregnada de una grasa especial, rodeada por alambres de acero enrollados en grupos, denominados cordones.

Existen varios tipos según la disposición de los cordones, como el tipo Seale y Warrington. Otras características son la composición del alambre y el material del alma.

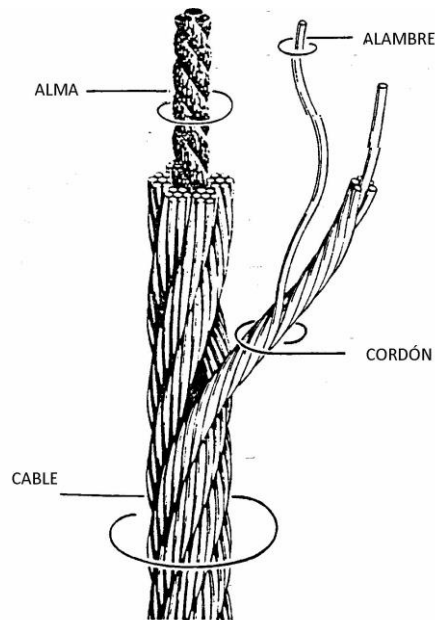


Figura 3.6 Esquema de las partes de un cable

3.2.2.5 Paracaídas

Unidad mecánica cuyo propósito es evitar la caída del ascensor en caso de desprendimiento de este. En el subapartado 3.2.1.2 se explica que el limitador de velocidad está conectado a una palanca que, cuando se produce una sobre velocidad de aproximadamente el 25%, activa el sistema de paracaídas.

Existen dos tipos de paracaídas:

- Paracaídas de acción instantánea: el cable del limitador activa las zapatas, que producen cada vez más rozamiento con las guías, llegando a producir acuñaamiento total del sistema. Es un sistema brusco, por lo que solo está permitido para ascensores de baja velocidad.
- Paracaídas de acción progresiva: las zapatas aplican una fuerza controlada y progresiva.



Figura 3.7 Sistema de seguridad paracaídas

3.2.3 Foso

Espacio vertical cerrado situado por debajo de la cota cero del ascensor. Esto hace que normalmente se encuentre por debajo del nivel de calle, por lo que, además de estar nivelado, debe estar adecuadamente impermeabilizado, ya que se enfrenta a posibles inundaciones.

En el interior del foso se encontrará un sistema de frenado, los amortiguadores. Por ello, su profundidad debe ser la suficiente como para que aun quede 0,5 metros libres con los amortiguadores totalmente comprimido.

El acceso al foso puede hacerse a través de una puerta especial de apertura hacia fuera. Sin embargo, si este supera 1,3 metros, debe instalarse una escalera.

3.2.3.1 Amortiguadores

Encargados de detener la cabina o el contrapeso si es necesario. Normalmente, se encuentran en el fondo del foso, aunque podrían también instalarse en la parte inferior de la cabina. Es muy importante que los amortiguadores tengan un sistema electrónico que no permita al ascensor seguir moviéndose hasta que estos vuelvan a su situación normal.

Se distinguen tres tipos de amortiguadores:

- Elásticos de caucho: constituidos por cilindros de caucho que son capaces de frenar ascensores con velocidades muy bajas (hasta 0,6 m/s), como montacargas.
- De resorte: la amortiguación la produce un muelle de sección circular. Puede ser utilizada con velocidades de hasta 1,75 m/s.
- Hidráulicos: formados por un émbolo hueco dentro de un cilindro lleno de un aceite especial. Cuando baja el émbolo, este entra poco a poco en su interior. La velocidad de bajada dependerá de la sección de los orificios a través de los cuales entre el fluido. Para regresar a su posición inicial se instalará un muelle en el fondo, que será oprimido al bajar el émbolo y pondrá a este en su posición inicial una vez que la fuerza pase.



Figura 3.8 Amortiguador de aceite hidráulico

3.3 Funcionamiento

A continuación, se expondrán las distintas ideas de algoritmos que se aplican hoy en día a los sistemas de elevación. En capítulos siguientes se establecerá cual será el usado en este proyecto.

3.3.1 Maniobra universal

Características:

- Un botón de llamada por planta.
- Los botones interiores tienen un tiempo de preferencia con respecto a los exteriores cuando un usuario sube al ascensor.
- No hay gestión de mayor cercanía de piso, será decisión de los usuarios quien pulsa primero.
- Solo atiende una llamada y no registra otra hasta no finalizar con la primera. Podría pasar que tuviera memoria pero que fuera en orden de llamada.
- Poco eficiente, por lo que se usa en ascensores de pocas plantas o montacargas.

3.3.2 Maniobra colectiva de descenso

Características:

- Un botón de llamada por planta.
- Los botones interiores tienen un tiempo de preferencia con respecto a los exteriores cuando un usuario sube al ascensor.
- Tiene memoria.
- En subida, atiende primero a los pedidos realizados por la botonera interna por orden de proximidad. Una vez acaba, atiende los pedidos de los rellanos.
- En bajada, atiende a los pedidos realizados por la botonera interna por orden de proximidad, pero, además, se detendrá en los rellanos que han realizado una llamada.
- Suele instalarse en bloques de viviendas, ya que la mayoría de las llamadas en rellano son para bajar a la planta cero.

3.3.3 Maniobra colectiva de subida y bajada

Características:

- Dos botones de llamada por planta; uno de subida y otro de bajada, que le permitirá al usuario elegir el sentido del movimiento.
- El sistema atiende todas las llamadas de botonera interna, y se para en aquellos rellanos que piden desplazamientos en la misma dirección.
- Tiene un excelente rendimiento, pero no es recomendable para viviendas, ya que un mal uso puede bajar dicho rendimiento.

4 DESCRIPCIÓN DE LA PLANTA

Los sistemas presentados en los capítulos 1 y 2 serán implementados en un sistema real formado por maqueta de ascensores conectada a un autómata programable donde se implementará el programa de control. A lo largo de este capítulo, se hará una descripción de ambos elementos, además de otros intermedios necesarios para que el sistema funcione correctamente.

Cabe destacar que la configuración de conexiones actual fue realizada por Daniel Paredes García en 2017, en su Trabajo de Fin de Grado *Automatización de un sistema de Ascensores*, de donde se obtiene mucha información para este Capítulo.

4.1 Maqueta del ascensor

En esta sección se realizará una descripción del sistema de forma física. Además, se analizará el número de sensores y actuadores de los que el autómata tendrá que hacerse cargo y se explicará la circuitería.



Figura 4.1 Planta a usar en este proyecto

4.1.1 Descripción física

La maqueta representa un sistema de elevación de cuatro plantas con dos ascensores. El movimiento de estos se hará desde arriba, que es donde se encuentran los motores.

Las dimensiones de la planta son de 75 cm de alto, 30 de ancho y 17 de profundidad. El material utilizado para el armazón es aluminio y para el ascensor es acero inoxidable. Estas tienen unas medidas de 10 cm de alto y una base cuadrada de 7x7 cm. También se incluyen dos pesas, una de 400 g para representar a las personas y otra de 450g para los contrapesos. El cable de tracción será hilo de sedal.

La planta usará pulsadores que representarán tanto los botones interiores de la cabina como aquellos botones de llamada de planta. De estos últimos habrá dos en las plantas intermedias, por lo que, como se vio en la Sección 3.3, será un sistema orientado a maniobra colectiva de subida y bajada. También se dispondrá de LEDs para la señalización de aquellas llamadas desde planta.

Como sensor de planta se usan unas pestañas metálicas conectadas al armazón. Los ascensores tienen un saliente en su lado izquierdo, que al pasar por una planta empujan dicha pestaña. Así, se cierra un circuito, de modo que cambia el estado de la señal y el autómatas es capaz de detectarlo.

Los ya mencionados motores funcionan con corriente continua, a 12 VCC y 3A, por lo que el consumo del par de motores será de 72 W. Cuando estos son activados, hacen girar un eje con un tornillo sin fin acoplado. Este tornillo hace girar una rueda dentada acoplada con el eje final del circuito, que es el que se encarga de transmitir el giro al hilo, gracias a las tres vueltas que le da.

4.1.2 Entradas y salidas

La planta dispone de un total de 26 entradas y 10 salidas:

De las 26 entradas, 6 corresponden a los pulsadores de planta y 12 a los pulsadores internos de cabinas, 5 por cada una de ellas. Los 8 restantes se completan con los sensores de planta, ya descritos anteriormente. Un ejemplo de los pulsadores puede verse en la Figura 4.8.

En cuanto a las 10 salidas, 4 se corresponden a los sentidos de giro de cada motor, 2 de subida y 2 de bajada, y los 6 LEDs que indican las peticiones previas del sistema.

El problema es que, como se verá en la descripción del autómatas, este solo dispone de 8 salidas. Por ello, se debió tomar la decisión de que, de los dos LEDs de las plantas intermedias, solo se tomará el de arriba como señalización de que se ha llamado a esa planta, independientemente del sentido.

A continuación, se muestra una tabla con las entradas y las salidas del sistema. También se incluye el nombre de las variables que se les ha dado al programarlas, cosa que se verá en el siguiente capítulo.

ENTRADA	DESCRIPCIÓN	NOMBRE
%I0.1.0	Botón 0 interior cabina derecha	B0D
%I0.1.1	Botón 1 interior cabina derecha	B1D
%I0.1.2	Botón 2 interior cabina derecha	B2D
%I0.1.3	Botón 3 interior cabina derecha	B3D
%I0.1.4	Botón S (Stop) interior cabina derecha	BSD
%I0.1.5	Botón A (Activar) interior cabina derecha	BAD
%I0.1.6	Botón 0 interior cabina izquierda	B0I
%I0.1.7	Botón 1 interior cabina izquierda	B1I
%I0.1.8	Botón 2 interior cabina izquierda	B2I
%I0.1.9	Botón 3 interior cabina izquierda	B3I
%I0.1.10	Botón S (Stop) interior cabina izquierda	BSI
%I0.1.11	Botón A (Activar) interior cabina izquierda	BAI
%I0.1.12	Botón 0 pasillo subida	B0
%I0.1.13	Botón 1 pasillo subida	B1S
%I0.1.14	Botón 2 pasillo subida	B2S
%I0.1.15	Botón 1 pasillo bajada	B1B
%I0.1.16	Botón 2 pasillo bajada	B2B
%I0.1.17	Botón 3 pasillo bajada	B3
%I0.1.18	Sensor planta 0 cabina izquierda	S0I
%I0.1.19	Sensor planta 1 cabina izquierda	S1I
%I0.1.20	Sensor planta 2 cabina izquierda	S2I
%I0.1.21	Sensor planta 3 cabina izquierda	S3I
%I0.1.22	Sensor planta 0 cabina derecha	S0D
%I0.1.23	Sensor planta 1 cabina derecha	S1D
%I0.1.24	Sensor planta 2 cabina derecha	S2D
%I0.1.25	Sensor planta 3 cabina derecha	S3D

Tabla 4.1 Relación de entradas del sistema

SALIDA	DESCRIPCIÓN	NOMBRE
%Q0.2.16	Motor 1 (Subida)	SUBIR1
%Q0.2.17	Motor 1 (Bajada)	BAJAR1
%Q0.2.18	Motor 2 (Subida)	SUBIR2
%Q0.2.19	Motor 2 (Bajada)	BAJAR2
%Q0.2.20	Led planta 0 pasillo subida	LED_EXT_0
%Q0.2.21	Led planta 1 pasillo subida	LED_EXT_1
%Q0.2.22	Led planta 2 pasillo subida	LED_EXT_2
%Q0.2.23	Led planta 1 pasillo bajada	NO SE USA
%Q0.2.24	Led planta 2 pasillo bajada	NO SE USA
%Q0.2.25	Led planta 3 pasillo bajada	LED_EXT_3

Tabla 4.2 Relación de salidas del sistema

4.2 Autómata programable

Un autómata programable o PLC (Controlador Lógico Programable en inglés) es un sistema electrónico y programable capaz de controlar en tiempo real procesos secuenciales; es decir, en base a las entradas (o sensores) listados anteriormente, y teniendo en cuenta el estado actual del proceso, activarán o desactivarán convenientemente las salidas (o actuadores).

Estos sistemas están omnipresentes en la industria debidos a su versatilidad. Esta versatilidad se debe, en parte, a que la mayoría de PLC actuales son modulares, lo que permite al usuario añadir una amplia gama de funcionalidades como control discreto, analógico, PID, posición, comunicación serie y red.

El PLC que completa esta planta será el Modicon M340 de Telemecanique, actualmente Schneider Electric. Como vemos en la imagen, de izquierda derecha, estará formado por un módulo de alimentación BMX CPS 2000, un procesador P34 2020, un módulo de entradas binarias BMX DDI 3202K y un módulo de E/S binarias BMX DDM 16022. Para la unión de los módulos se utilizará el bastidor de 4 ranuras BMX XBP 0400.



Figura 4.2 Autómata usado en el proyecto

4.2.1 Bastidor BMX XBP 0400

Es el elemento de unión de los distintos módulos que forman el PLC. Normalmente, estos bastidores suelen instalarse en railes tipo DIN, aunque, en este caso, este se encuentra atornillado en una pared. Además de la función mecánica, también se encarga de llevar la alimentación del primero hacia los demás.

Cuenta con 6 ranuras:

- Primera ranura (CPS): reservada para módulo de alimentación.
- Ranura 00: reservada para el procesador.
- Ranuras de 01 a 03: módulos E/S (En este caso solo se usarán dos de ellas)
- Ranura XBE: ranura de expansión que permite conectar varios bastidores para así poder ampliar el número de módulos.

4.2.2 Módulo de alimentación BMX CPS 2000

El autómata se alimenta de corriente alterna de la red eléctrica. Por tanto, es necesario un módulo de alimentación que la transforme en corriente continua.

Los elementos de este módulo:

- Panel de visualización: indicador LED verde OK, que indica que el módulo se encuentra operativo y en correcto funcionamiento e indicador LED verde 24V, que indica que la fuente de alimentación interna de 24VCC se encuentra operativa, aunque en este caso no se utilizará.
- Relé de alarma normalmente cerrado que se abre al detectar un error en el sistema. Este relé tampoco será utilizado en el proyecto.
- Botón RESET que provoca un reinicio del sistema.
- Conector de red de entrada con fusible interno conectado a la fase de entrada.

Como ya se ha indicado, para este proyecto no se utilizará la fuente de alimentación interna. Esto es debido a que ya se dispone de otra fuente externa de 24 V que nos permitirá un mayor orden en el conexionado.

4.2.3 Procesador BMX P34 2020

El procesador BMX P34 2020 es el encargado de gestionar los módulos E/S, con una capacidad de gestión de hasta 1024 E/S binarias o 256 E/S analógicas, con una memoria de 4096 Kb.

Como es de costumbre en este tipo de módulos, tienen un panel de visualización en el que, mediante LEDs, muestra el estado del procesador. Además, también cuenta con un puerto USB, un puerto Ethernet y un puerto Modbus. En este proyecto se usarán los dos primeros; el puerto USB para la programación del autómata y el puerto Ethernet para la transmisión de información al SCADA.

En la imagen también se observa una pequeña tapa o palanca con un punto azul. Si se levanta dicha palanca, se encuentra en su interior una tarjeta de memoria en la que se encuentra información sobre el autómata necesaria para su funcionamiento.

4.2.4 Módulo BMX DDI 3202K

Es un módulo de 32 entradas binarias con una tensión nominal de entrada de 24 VCC y una corriente de 2,5 mA. El tipo de lógica que utiliza este módulo es lógica positiva; es decir, las señales de tensión alta estarán representadas por un “1” binario (Tensión mayor a 11 V y corriente mayor a 2 mA) y las de tensión baja por un “0” binario (Tensión igual a 5 V y corriente inferior a 1,5 mA).

Una vez más, este módulo cuenta con un panel de visualización con números del 00 al 31. Cuando alguno de estos sensores hace que se cierre el circuito, la tensión llega y enciende el led correspondiente al sensor que ha sido activado. En la imagen del autómata pueden observarse algunos sensores activados y otros sin activar.

La conexión se hará mediante un conector de 40 pines repartidos en dos mangueras de 20 cables cada una. De ellos, 32 serán los correspondiente a las entradas del sistema.

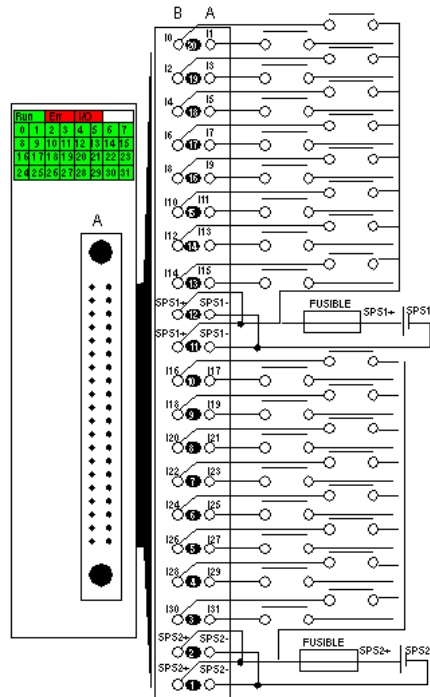


Figura 4.3 Diagrama de conexión BMX DDI 3202K

4.2.5 Módulo BMX DDM 16022

Módulo de 8 entradas y 8 salidas (razón por la cual no se pueden usar todos los leds de la planta) con una tensión nominal de 24 VCC y una corriente de 3,5 mA. En este caso, el “1” binario se conseguirá con 11 V y 3 mA, mientras que el “0” binario se conseguirá con 5 V y una corriente inferior a 1,5 mA.

Al igual que en el módulo BMX DDI 3202K, cuando el contacto de la entrada o salida está abierto, el número correspondiente en el panel de visualización estará encendido y viceversa.

La conexión, al igual que en el módulo anterior, se realizará mediante un conector, esta vez de 20 pines.

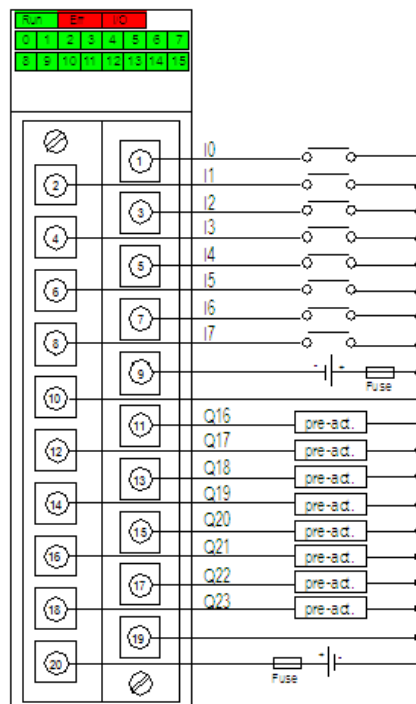


Figura 4.4 Diagrama de conexión BMX DDM 16022

4.3 Otros elementos

Ya se han descrito los dos elementos principales que conforman la planta que se va a utilizar: la maqueta de ascensores y el autómatas programable. Sin embargo, habrá otros elementos necesarios que harán que el sistema esté cerrado y todo funcione como deba.

Estos elementos son:

- Fuente de alimentación TCP 055-124 de 24 VCC y 2,3 A, que sustituirá a la fuente interna del módulo de alimentación BMX CPS 2000. Se usará para detectar los sensores y activar los LEDs.
- Fuente de alimentación TCP 120-112 de 12VCC y 8 A, que dará corriente a las 4 salidas que corresponden a la subida y bajada de cada motor.
- 8 relés de conmutación doble usados como relés de conmutación de simple. Para ello, de los 6 terminales que tiene cada uno (2 comunes, 2 NA y 2 NC) solo se conectará a uno de cada tipo, aunque siempre del mismo conmutador. Al activarse una salida en el módulo BMX DDM 16022, lo que realmente se activará será uno de estos 8 relés, que será el encargado de cerrar el circuito entre la fuente de alimentación correspondiente y el motor o el LED.
- 2 placas IFM DEE 25 F de M-Jay Electronics Ltd, que es una interfaz de bloque de terminales de 25 posiciones que será de ayuda para un cableado más cómodo.



Figura 4.5 Placa IFM DEE 25 F de M-Jay Electronics Ltd

- Dos conectores DB 25 hembra.
- 4 fusibles; dos de ellos de 0,5 mA serán utilizados a la entrada del módulo BMX DDI 3202K mientras que en el BMX DDM 16022 se utilizará uno de 0,5 mA para la entrada y otro de 2 mA a la salida.
- 4 resistencias de 2K Ω para evitar que los LEDs se quemen.
- 3 mangueras de 20 cables cada una.

La distribución de los sensores de las mangueras y el conexionado de las entradas y salidas de la maqueta con el autómatas se verá a continuación.

4.4 Conexión PLC – Maqueta

4.4.1 Conectores DB

Al tener tantas entradas y salidas, hay una gran cantidad de cables. Por ello, se recurre a dos conectores DB 25 macho, que los recogerán y facilitarán la conexión y desconexión entre autómatas y maqueta. Hay que tener en cuenta que para transportar la planta es conveniente desconectar ambas partes, por lo que es una situación que se da de forma relativamente frecuente.

A continuación, se expone la relación de cables de cada uno de estos dos conectores.

CONECTOR	DESCRIPCIÓN
PIN 1	Botón 0 interior cabina derecha
PIN 2	Botón 1 interior cabina derecha
PIN 3	Botón 2 interior cabina derecha
PIN 4	Botón 3 interior cabina derecha
PIN 5	Botón S (Stop) interior cabina derecha
PIN 6	Botón A (Activar) interior cabina derecha
PIN 7	Libre
PIN 8	Libre
PIN 9	Botón 0 interior cabina izquierda
PIN 10	Botón 1 interior cabina izquierda
PIN 11	Botón 2 interior cabina izquierda
PIN 12	Botón 3 interior cabina izquierda
PIN 13	Botón S (Stop) interior cabina izquierda
PIN 14	Botón A (Activar) interior cabina izquierda
PIN 15	Libre
PIN 16	Libre
PIN 17	Botón 0 pasillo subida
PIN 18	Botón 1 pasillo subida
PIN 19	Botón 2 pasillo subida
PIN 20	Botón 1 pasillo bajada
PIN 21	Botón 2 pasillo bajada
PIN 22	Botón 3 pasillo bajada
PIN 23	Libre
PIN 24	Libre
PIN 25	Libre

Tabla 4.3 Correlación cables-conectores del primer DB 25

Para el segundo conector:

CONECTOR	DESCRIPCIÓN
PIN 1	Común de los leds
PIN 2	Motor 1 (Subida)
PIN 3	Motor 1 (Bajada)
PIN 4	Libre
PIN 5	Sensor planta 0 cabina izquierda
PIN 6	Sensor planta 1 cabina izquierda
PIN 7	Sensor planta 2 cabina izquierda
PIN 8	Sensor planta 3 cabina izquierda
PIN 9	Sensor planta 0 cabina derecha
PIN 10	Sensor planta 1 cabina derecha
PIN 11	Sensor planta 2 cabina derecha
PIN 12	Sensor planta 3 cabina derecha
PIN 13	Libre
PIN 14	Común pulsadores
PIN 15	Motor 2 (Subida)
PIN 16	Motor 2 (Bajada)
PIN 17	Led planta 0 pasillo subida
PIN 18	Led planta 1 pasillo subida
PIN 19	Led planta 2 pasillo subida
PIN 20	Led planta 1 pasillo bajada
PIN 21	Led planta 2 pasillo bajada
PIN 22	Led planta 3 pasillo bajada
PIN 23	Libre
PIN 24	Libre
PIN 25	Libre

Tabla 4.4 Correlación cables-conectores del segundo DB 25

4.4.2 Conexión de entradas a módulos de autómatas

4.4.2.1 Conexión

Es indispensable conectar las fuentes de alimentación de 24V (LEDS) y 12V (Motores) a la corriente eléctrica, así como el módulo de alimentación BMX CPS 2000. Sin embargo, la instalación solo cuenta con una clavija eléctrica, por lo que se ha conectado a ella los cables de la fuente de 24VDC y se les ha hecho un puente a las demás fuentes.

Posteriormente, una vez que los conectores DB macho se conecten a los conectores hembra, la señal pasará de los cables a los terminales de salida de las placas IFM DEE 25 F. Estos terminales serán los que se conecten, mediante cableado, a los módulos E/S del autómatas.

En la siguiente tabla se muestran las conexiones DB – Terminal – BMX DDI 3202K realizadas:

CONECTOR	DESCRIPCIÓN	TERMINAL	COLOR	MANGUERA	CANAL
PIN 1	Botón 0 interior cabina derecha	1	Blanco	1	I0
PIN 2	Botón 1 interior cabina derecha	2	Marrón	1	I1
PIN 3	Botón 2 interior cabina derecha	3	Verde	1	I2
PIN 4	Botón 3 interior cabina derecha	4	Amarillo	1	I3
PIN 5	Botón S (Stop) interior cabina derecha	5	Gris	1	I4
PIN 6	Botón A (Activar) interior cabina derecha	6	Rosa	1	I5
PIN 9	Botón 0 interior cabina izquierda	9	Azul	1	I6
PIN 10	Botón 1 interior cabina izquierda	10	Rojo	1	I7
PIN 11	Botón 2 interior cabina izquierda	11	Negro	1	I8
PIN 12	Botón 3 interior cabina izquierda	12	Morado	1	I9
PIN 13	Botón S (Stop) interior cabina izquierda	13	Gris-Rosa	1	I10
PIN 14	Botón A (Activar) interior cabina izquierda	14	Rojo-Azul	1	I11
PIN 17	Botón 0 pasillo subida	17	Blanco-Verde	1	I12
PIN 18	Botón 1 pasillo subida	18	Marrón-Verde	1	I13
PIN 19	Botón 2 pasillo subida	19	Blanco-Amarillo	1	I14
PIN 20	Botón 1 pasillo bajada	20	Amarillo-Marrón	1	I15
PIN 21	Botón 2 pasillo bajada	21	Blanco	2	I16
PIN 22	Botón 3 pasillo bajada	22	Marrón	2	I17

Tabla 4.5 Conexión primer DB 25 a módulo BMX DDI 3202K

CONECTOR	DESCRIPCIÓN	TERMINAL	COLOR	MANGUERA	CANAL
PIN 5	Sensor planta 0 cabina izquierda	5	Verde	2	I18
PIN 6	Sensor planta 1 cabina izquierda	6	Amarillo	2	I19
PIN 7	Sensor planta 2 cabina izquierda	7	Gris	2	I20
PIN 8	Sensor planta 3 cabina izquierda	8	Rosa	2	I21
PIN 9	Sensor planta 0 cabina derecha	9	Azul	2	I22
PIN 10	Sensor planta 1 cabina derecha	10	Rojo	2	I23
PIN 11	Sensor planta 2 cabina derecha	11	Negro	2	I24
PIN 12	Sensor planta 3 cabina derecha	12	Morado	2	I25

Tabla 4.6 Conexión segundo DB 25 a módulo BMX DDI 3202K

Como se observa por las tablas, el primer y segundo conector DB se diferencian en que, mientras que el primer DB contiene exclusivamente entradas, el segundo contiene tanto entradas como salidas. Por otro lado, las entradas se conectan, tal y como se ve, directamente del terminal correspondiente al módulo de entradas, mientras que las salidas irán del terminal a un relé y del relé al autómatas.

4.4.2.2 Alimentación de módulo BMX DDI 3202K

Como se observa en la Figura 4.3, para la alimentación del módulo BMX DDI 3202K son necesarios 8 cables: 2 SP1+, 2 SP1-, 2 SP2+ y 2 SP2-. Para ellos se usarán los cables Blanco-Gris, Blanco-Rosa, Marrón-Gris y Marrón-Rosa de ambas mangueras del módulo, como se podrá ver con más detalle en la Figura 4.10.

La conexión de alimentación se hará en tres pasos:

1. Se realiza un puente entre los cables Blanco-Gris y Blanco-Rosa de la primera y de la segunda manguera, de forma separada. También se realiza un puente entre los cuatro cables Marrón-Gris y Marrón-Rosa, esta vez de forma conjunta.
2. Cada uno de los puentes Blanco-Gris y Blanco-Rosa se soldarán a un cable gris, mientras que el puente de cuatro cables Marrón-Gris y Marrón-Rosa se soldarán directamente a un cable negro.
3. El cable negro se conectará directamente al negativo de la fuente 24 VCC, mientras que el cable gris, que irá protegido por un fusible 0,5A, irá conectado al positivo.

4.4.3 Conexión de salidas a relés

Los relés utilizados en este proyecto serán relés dobles pero usados como relés simples. Estos relés dobles tienen 6 terminales cada uno: dos normalmente abiertos (NA) en los terminales 2 y 5, dos normalmente cerrados (NC) en los terminales 3 y 6 y dos comunes en los terminales 1 y 4. Al ser usados de forma simple, solo se usará uno de cada, pero siempre en el mismo conmutador, por lo que siempre se usará entre los números 1-3 o 4-6.

En cuanto a la nomenclatura, los relés están nombrados alfabéticamente desde la letra I a la P (ocho letras para ocho relés), por lo que se podrá nombrar un terminal determinado de un relé mediante su número y su letra.

Cabe destacar que los relés tienen su propio fusible para impedir sobretensiones.

4.4.3.1 Conexión de motores

Los motores van alimentados por la fuente de alimentación externa de 12VCC. Para su conexión, se utilizarán un total de 4 relés, dos para cada uno (subida y bajada). Así, se evita que se le pueda dar la “orden” a cualquiera de los motores de subir y bajar al mismo tiempo, quemando el sistema.

El esquema de conexión de los motores es el siguiente:

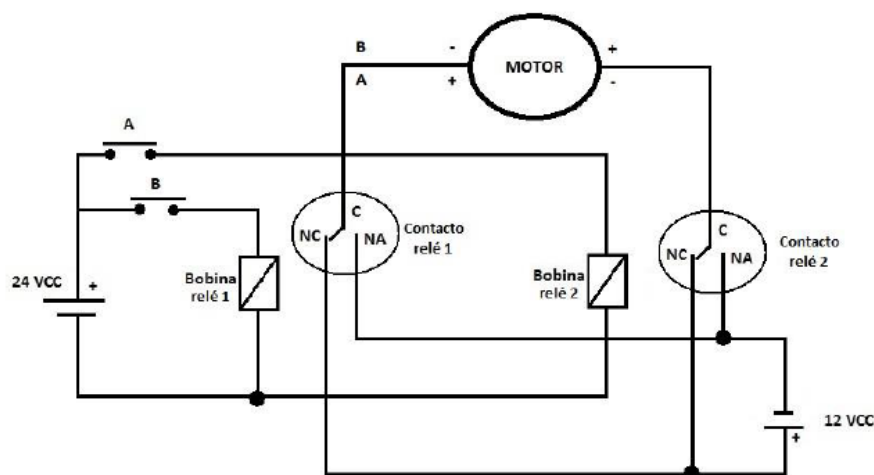


Figura 4.6 Esquema de conexión de motores

Como se puede ver en el esquema, el motor estará conectado a los comunes, mientras que los NC y NA irán conectados a la fuente de 12 VCC que alimenta al motor. Los interruptores A y B se interpretan como “0” o “1” lógico, y se representa a la fuente de 24 VCC porque alimentará a las bobinas de los relés.

A continuación, se examinan los distintos casos que pueden darse en este circuito:

1. Si no se pulsa ningún interruptor, el motor se encuentra en un circuito cerrado sin alimentación, por lo que no se moverá.
2. Si se pulsa A y no B, el segundo relé pasará a NA, por lo que se cerrará el circuito con 12V, lo que hará que el motor gire.
3. Si se pulsa B y no A, el primer relé pasará a NA, por lo que ocurrirá lo mismo que en el caso anterior, pero en el sentido contrario.
4. Si se pulsan A y B ocurre lo mismo que en el caso 1, se crea un bucle cerrado sin alimentación, por lo que el motor no se movería. Así, se evita el quemado del motor.

Por ello, se hará una conexión desde los terminales de la placa IFM DEE 25 F correspondiente a los motores hasta los relés. Para ello se utilizará cable azul y quedará la siguiente tabla de conexión.

CONECTOR	DESCRIPCIÓN	TERMINAL DB	TERMINAL RELÉ (C)
PIN 2	Motor 1 (Subida)	2	4I
PIN 3	Motor 1 (Bajada)	3	4J
PIN 15	Motor 2 (Subida)	15	4K
PIN 16	Motor 2 (Bajada)	16	4L

Tabla 4.7 Conexión de motores DB 25 a relés

Por otro lado, tal y como se observa en la Figura 4.6, los NA se encuentran conectados al terminal negativo de la fuente de 12 VCC, por lo que se ha realizado un puente entre ellos y se han conectado al negativo de la fuente. Para ello se ha usado cable de color negro.

En cuanto a los NC, se aprecia que están conectados al terminal positivo, por lo que se les realiza un puente y se conectan al terminal positivo con cables de color gris.

4.4.3.2 Conexión de leds

Como ya se ha dicho anteriormente, el autómat PLC solo dispone de 8 salidas. Cuatro de ellas están destinadas a la subida y bajada de los motores, por lo que las 4 restantes serán para los leds, aunque en realidad la maqueta tiene seis. Por ello, se decide conectar los leds de las plantas 0 y 3 y los leds de subida de las plantas 1 y 2.

Los leds van alimentados por la fuente externa de 24 VCC y protegidos por una resistencia de 2kΩ cada uno para evitar que la intensidad que circula por ellos sea lo suficientemente grande como para quemarlos.

Harán falta, por tanto, cuatro relés. El circuito de conexión es el siguiente:

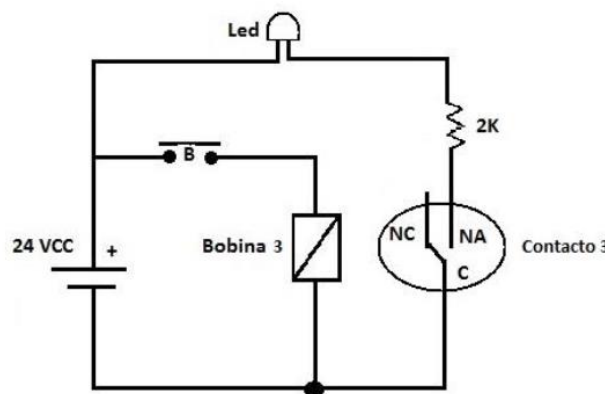


Figura 4.7 Esquema de conexión de leds

El análisis de casos será más sencillo que en el circuito de los motores:

1. Si no se pulsa B, el circuito estará abierto, por lo que no habrá corriente a través del led y no se encenderá.
2. Si se pulsa B, se cerrará el circuito, circulará la corriente a través de la resistencia y llegará al led, encendiéndolo.

Por tanto, se realizará una conexión desde la placa a los relés parecida a la realizada en el caso de los motores. Teniendo en cuenta que, según el diagrama, los NA de los relés están conectados a los cátodos de los leds:

CONECTOR	DESCRIPCIÓN	TERMINAL DB	TERMINAL RELÉ (NA)
PIN 17	Led planta 0 pasillo subida	17	2M
PIN 18	Led planta 1 pasillo subida	18	2N
PIN 19	Led planta 2 pasillo subida	19	2O
PIN 22	Led planta 3 pasillo bajada	22	2P

Tabla 4.8 Conexión de leds DB 25 a relés

Por otro lado, los comunes están conectados al terminal negativo de la fuente 24 VCC, por lo que se les realiza un puente y se conectan. Los NC quedan al aire.

4.4.3.3 Comunes de entradas y leds

Una vez conectadas las entradas y salidas de la 2ª DB, solo quedarán los terminales 1 y 14 sin conectar, los cuales corresponden al común de las entradas y al común de los leds; es decir, sus ánodos, respectivamente. Estos dos terminales se conectarán al terminal positivo de la fuente de alimentación, para cerrar sus respectivos circuitos.

4.4.4 Conexionado de relés a módulos de autómatas

4.4.4.1 Conexionado

Con las entradas y salidas conectadas a sus respectivos relés, solo queda la conexión de los relés al módulo de salida BMX DDM 16022. A este módulo se conectarán los terminales de las bobinas de los relés, pues, activándolas, esta hará la conmutación del relé y por tanto del estado del actuador.

Las conexiones relé – autómatas quedan de la siguiente forma:

TERMINAL RELÉ	COLOR	CANAL
I	Negro	Q16
J	Morado	Q17
K	Gris-Rosa	Q18
L	Azul-Rojo	Q19
M	Blanco-Verde	Q20
N	Marrón-Verde	Q21
O	Blanco-Amarillo	Q22
P	Marrón-Amarillo	Q23

Tabla 4.9 Conexión de relés a módulo BMX DDM 16022

El negativo de las bobinas debe conectarse con el negativo de la fuente 24 VCC para poder cerrar su circuito.

4.4.4.2 Alimentación de módulo BMX DDM 16022

Tal y como se ve en la Figura 4.4, la alimentación de este módulo se hará mediante cuatro de los canales de este:

1. Los canales 9 y 19 (Marrón-Gris y Marrón-Rosa, respectivamente) se conectarán al negativo de la fuente de alimentación 24 VCC.
2. El canal 10 (Blanco-Gris) se conectará a un fusible 0,5 mA y, seguidamente, al positivo de la fuente de alimentación 24 VCC.
3. El canal 20 (Blanco-Rosa) se conectará a un fusible 2 mA, ya que es la alimentación de las salidas, y posteriormente al positivo de la fuente de alimentación.

A continuación, se muestra una imagen de los relés, los conectores DB y parte de los botones de entrada al sistema:



Figura 4.8 Relés, conectores DB y botones

Y una imagen trasera, donde se aprecia todo el conexionado:

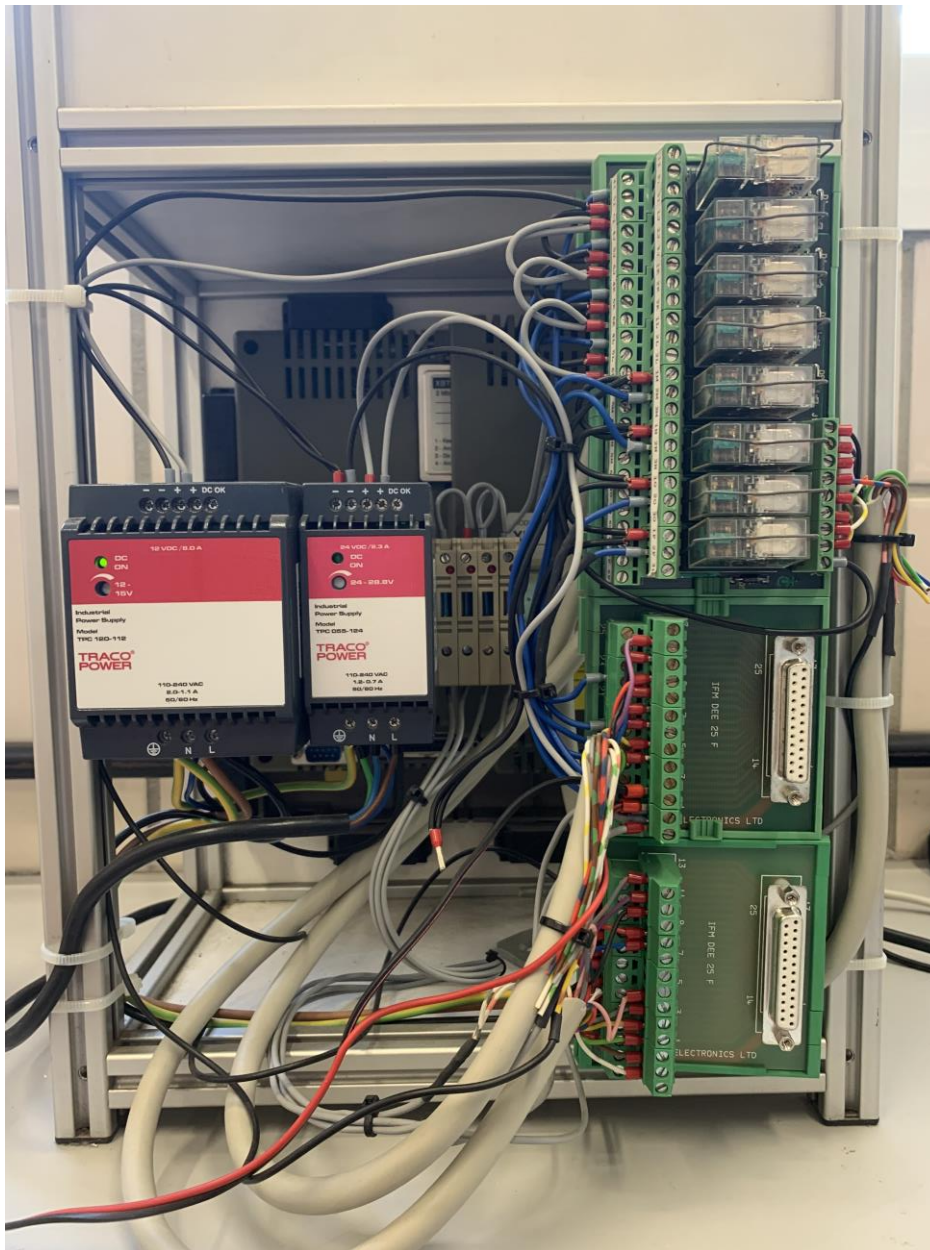


Figura 4.9 Conexionado Maqueta - PLC

Además, se adjunta un esquema de todo el conexionado, ocultando los cables de red.

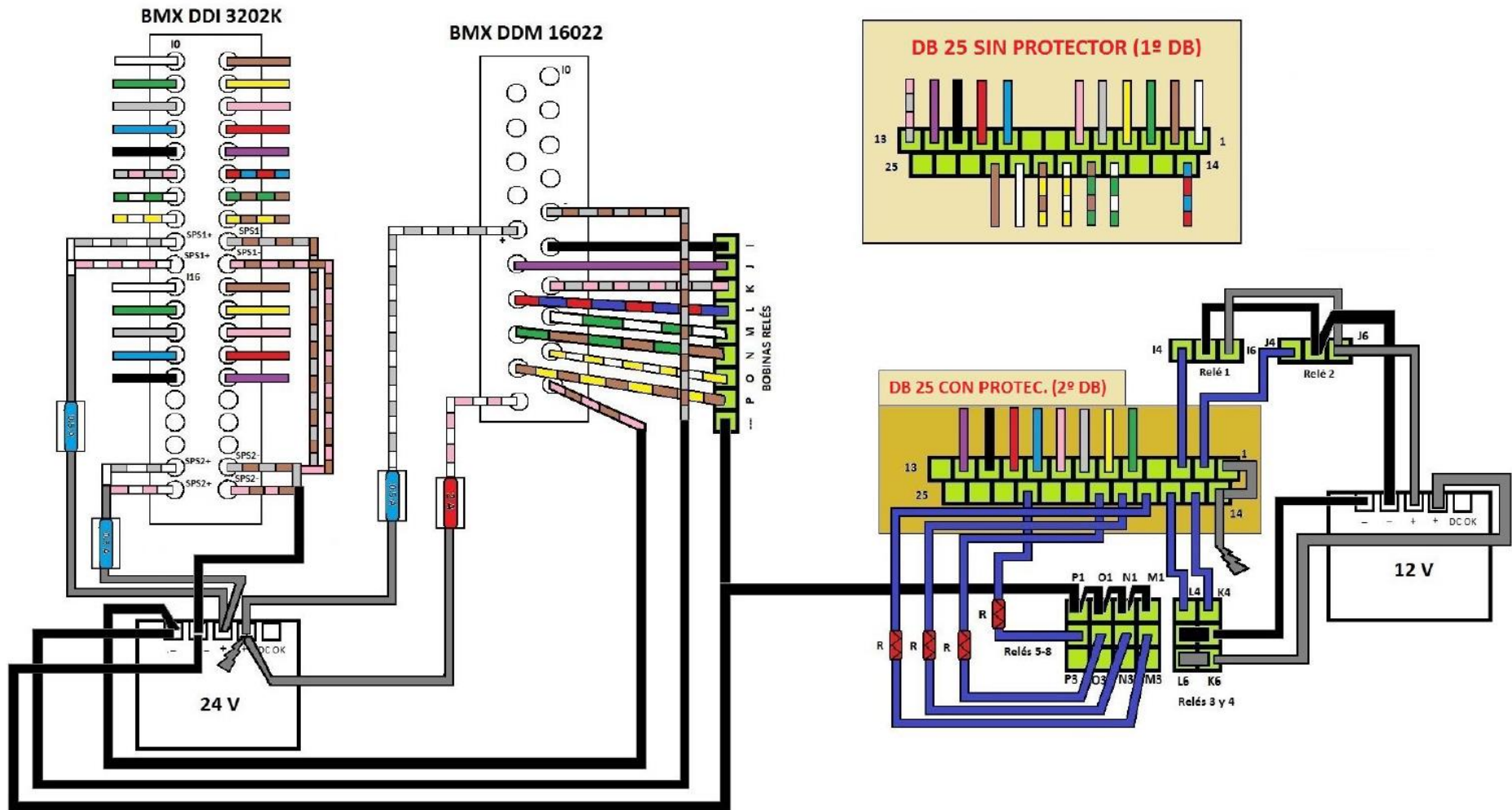


Figura 4.10 Esquema de conexiones Maqueta - PLC

5 PROGRAMAS UTILIZADOS

El presente proyecto puede dividirse en tres partes fundamentales: la programación del autómata, la comunicación entre el PLC y el sistema SCADA y, finalmente, el diseño de este. Este capítulo recogerá una breve introducción sobre los tres programas que se utilizarán para implementar cada una de las partes: ‘Ecostruxure Control Expert’, ‘OPC UA Server Expert’ y ‘Citect Studio 2018 R2’.

5.1 Ecostruxure Control Expert

Ecostruxure Control Expert es un software diseñado por ‘Schneider Electric’ usado para programar, resolver fallos y operar con autómatas de las gamas Modicon M340 (el usado en este proyecto), M580, M580S, Premium, Momentum y Quantum. Así, permitirá acceder a los datos de la planta y hacer una integración acelerada con sistemas IT (Information Technology) y OT (Operation Technology).

Como ya se vio en el capítulo anterior, el autómata integrado en la planta es de la marca Telemecanique, empresa que pertenece a Schneider Electric, por lo que será el software utilizado para programar la planta.

A continuación, se expondrán las especificaciones de ordenador necesarias para soportar el software. Seguidamente, se hará una breve introducción al programa.

	MINIMO	RECOMENDADO
Procesador	Pentium Core i3 2.4 GHz	Pentium Core i7 3 GHz
RAM	4 Gb	8 Gb
Espacio mínimo disco duro	8 Gb	20 Gb

Tabla 5.1 Requisitos Ecostruxure Control Expert

5.1.1 Explorador de proyectos

Al abrir un nuevo proyecto tendrá que elegirse el procesador que se va a utilizar, en este caso, el P34 2020. Posteriormente, aparecerá un explorador de proyectos desde donde se accederá a todas sus funcionalidades.

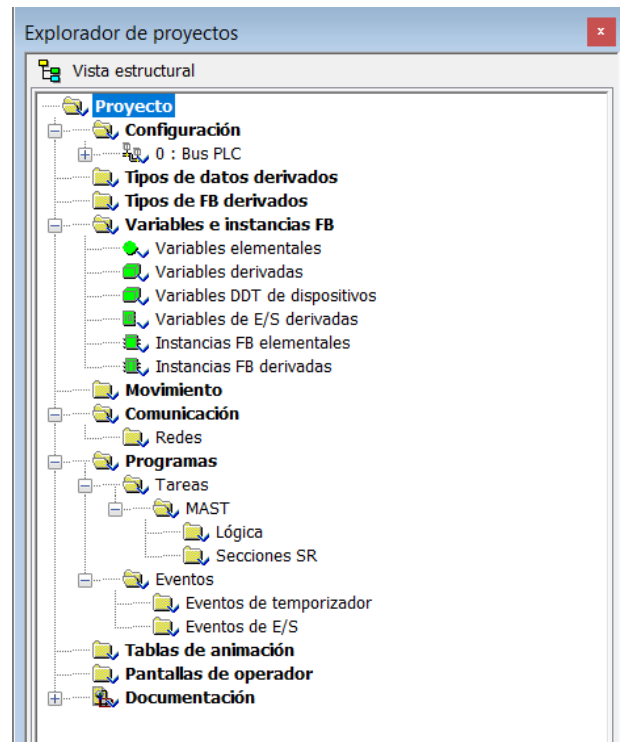


Figura 5.1 Explorador de proyectos Ecostruxure

En orden, se irán describiendo algunos de los apartados más importantes:

5.1.1.1 Configuración

Será el apartado donde se declarará los módulos del autómeta que tendrán acceso al proyecto. En el Capítulo anterior ya se analizó módulo por módulo el PLC de la planta a utilizar en este proyecto. En el programa:

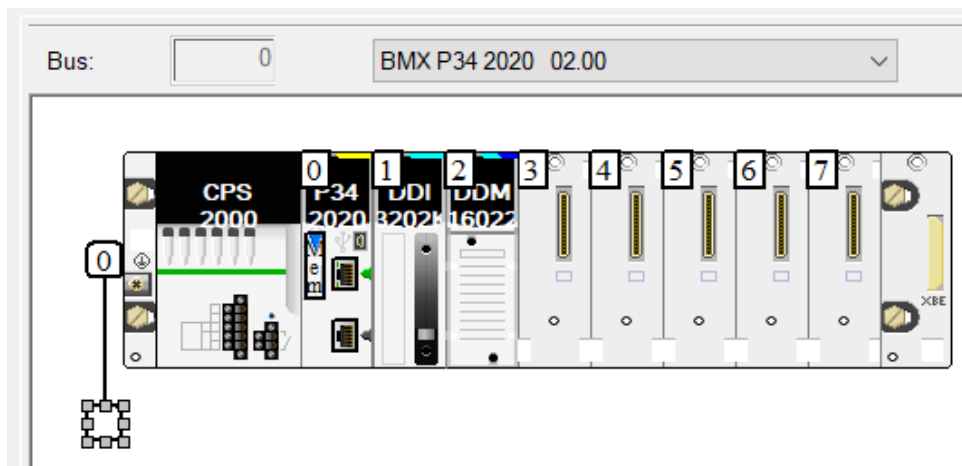


Figura 5.2 Configuración de módulos del autómeta

5.1.1.2 Variables elementales

En este apartado tendrán que declararse todas aquellas variables que aparecerán en el código, indicando de que tipo son. En el siguiente capítulo se analizarán las variables usadas en este proyecto. Sin embargo, a modo de ejemplo, puede verse como son declarados en el programa los botones, tanto interiores como exteriores:

Nombre	Tipo	Valor	Comentario	Alias	Alias de	Dirección	Variable de HMI	Derechos de L/E de la variable referenciado
B0	EBOOL		Botón exterior 0			%I0.112		
B0D	EBOOL		Botón interior 0 A1			%I0.10		
B0I	EBOOL		Botón interior 0 A2			%I0.16		
B1B	EBOOL		Botón exterior 1 Baja...			%I0.115		
B1D	EBOOL		Botón interior 1 A1			%I0.11		
B1I	EBOOL		Botón interior 1 A2			%I0.17		
B1S	EBOOL		Botón exterior 1 Subi...			%I0.113		
B2B	EBOOL		Botón exterior 2 baja...			%I0.116		
B2D	EBOOL		Botón interior 2 A1			%I0.12		
B2I	EBOOL		Botón interior 2 A2			%I0.18		
B2S	EBOOL		Botón exterior 2 Subi...			%I0.114		
B3	EBOOL		Botón exterior 3			%I0.117		
B3D	EBOOL		Botón interior 3 A1			%I0.13		
B3I	EBOOL		Botón interior 3 A2			%I0.19		
BAD	EBOOL		Botón interior A A1			%I0.15		
BAI	EBOOL		Botón interior A A2			%I0.111		

Figura 5.3 Ejemplo de variables elementales del sistema

5.1.1.3 Redes

Este apartado será muy importante para la creación de la red Ethernet, que permitirá al autómatas conectarse con el servidor OPC UA, que a su vez se conectará con el SCADA. Esto se verá en el Capítulo 7.

5.1.1.4 Programas

Con un clic derecho en la carpeta 'Lógica', el programa nos dará la posibilidad de crear una nueva Sección, que es algo parecido a lo que sería un Script en programación convencional. Podemos crear tantas secciones como deseemos con la precaución de que, en principio, todas se ejecutarán constantemente.

Cuando se crea una nueva sección, es necesario especificar en qué lenguaje se programará:

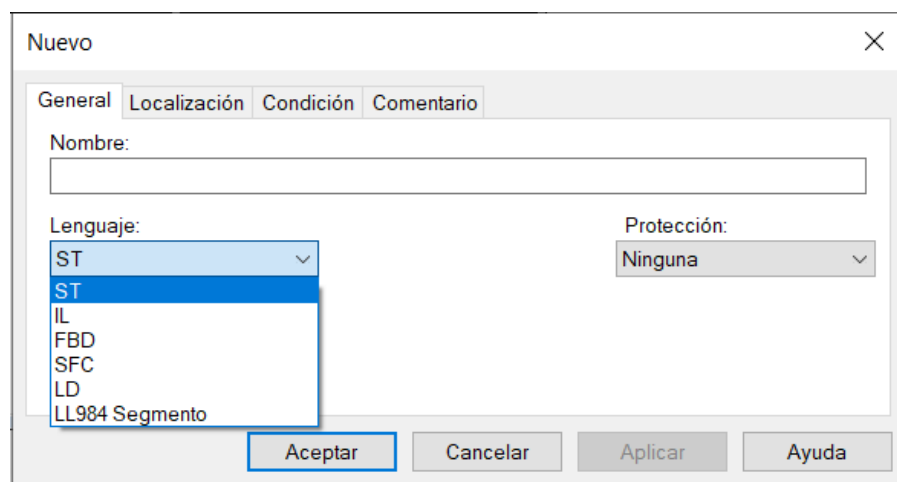


Figura 5.4 Ejemplo de creación de sección

Por ello, es importante comprender los tres lenguajes principales, que serán los tres usados en este proyecto:

- LD: El lenguaje LD o lenguaje de contactos es un lenguaje gráfico que representa circuitos eléctricos usando contactos (entradas binarias) para dar un valor a las bobinas (salidas binarias). Este es el lenguaje de programación más parecido a la lógica cableada, por lo que es muy sencillo para un electricista acostumbrado a fabricar cuadros de automatismos.

Además de los contactos y bobinas se pueden encontrar muchos más elementos basados en estos dos. Algunos ejemplos son los contactos negados, contactos con flancos de subida o bajada, bobinas negadas, bobinas con flancos o bobinas especiales set y reset.

Aunque se verá en el siguiente Capítulo, este lenguaje ha sido utilizado en este proyecto para algunas secciones destinadas por ejemplo a la gestión del modo manual o, gracias a los contactos y bobinas con flancos, a secciones que gestionaban su detección. También suele ser muy utilizado para la gestión de temporizadores o contadores.

- SFC: El lenguaje SFC deriva de las ‘redes de Petri’, que son muy utilizadas para modelar matemáticamente sistemas de eventos discretos secuenciales, en el que el estado siguiente de un sistema dependerá del estado actual y de las entradas en ese momento del propio sistema.

Por ello, se creará un diagrama formado por ‘pasos o etapas’, donde se realizarán acciones o se activarán variables o actuadores y ‘transiciones’, que serán las condiciones que tendrán que ocurrir para pasar de un paso a otro.

En este proyecto se ha usado este lenguaje como gestor de todo el movimiento de los ascensores y las transiciones entre modos de funcionamiento.

- ST: El lenguaje ST o lenguaje estructurado es un lenguaje escrito de alto nivel muy parecido a C. Podemos encontrar funciones básicas como IF-ELSE o bucles, aunque con una sintaxis más trabajosa que en otros lenguajes.

A diferencia de los otros dos lenguajes, ST trabaja de forma muy cómoda con variables, vectores y entradas y salidas analógicas. Por ello, este lenguaje se ha usado para programar la sección de control del autómeta.

Un apunte importante y a veces difícil de encontrar es como hacer que una sección se ejecute o no en función de una variable. Esto es de especial importancia para la gestión de modos y se hace en las propiedades de la sección a controlar.

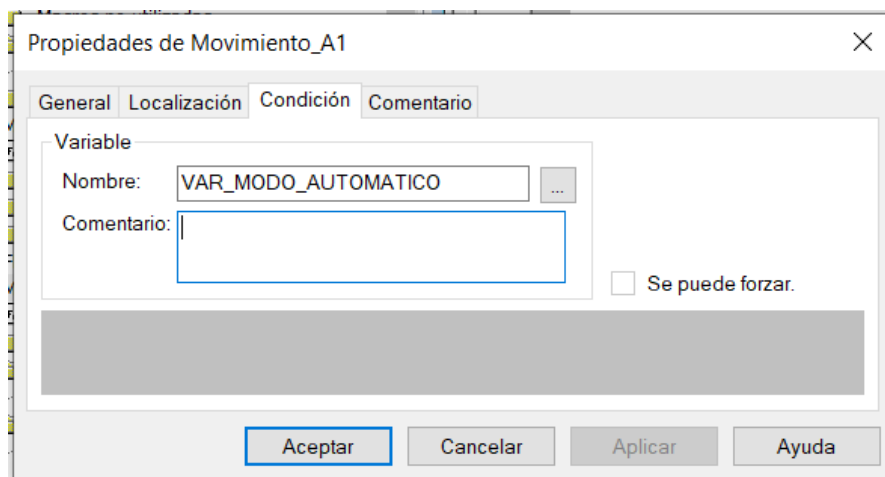


Figura 5.5 Condición de activación de una sección

Por ejemplo, se ve que, en la imagen de arriba, la sección ‘Movimiento_A1’ solo se ejecutará cuando la variable VAR_MODAL_AUTOMÁTICO esté activa o TRUE.

5.1.1.5 Tabla de animación

La tabla de animación permite al usuario forzar variables. Esta función es de especial importancia para hacer pruebas y ha sido de especial importancia en este proyecto.

Nombre	Valor	Tipo	Comentario
● SUBIR1		EBOOL	
● BAJAR1		EBOOL	
● SUBIR2		EBOOL	
● BAJAR2		EBOOL	
● LED_EXT_0		EBOOL	
● LED1		EBOOL	
● LED2		EBOOL	
● LED3		EBOOL	

Figura 5.6 Tabla de animación

5.1.1.6 Pantallas de operador

La pantalla de operador tiene una función similar al SCADA, pero desde el ordenador local. En ella, se pueden crear simuladores y botones virtuales, que podrán interactuar con la planta en tiempo real.

En este proyecto se ha utilizado como una botonera virtual para gestionar el modo manual y automático y, dentro de este, la subida y bajada de los ascensores.

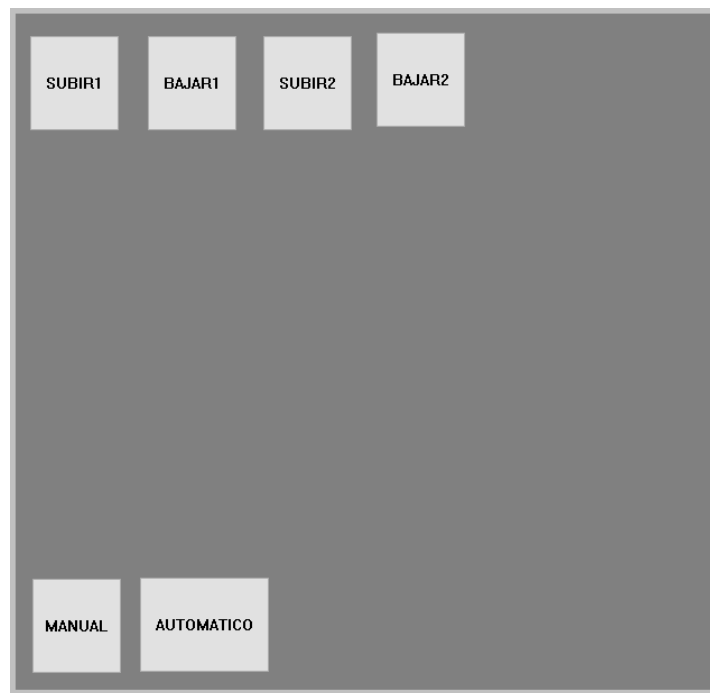


Figura 5.7 Pantalla de operador

5.1.2 Conexión PC - PLC

La conexión entre el PC y el autómatas se realizará por medio del puerto serie del autómatas que se encuentra en el módulo procesador BMX P34 2020. Una vez conectado, en el programa Ecostruxure tendremos que darle a conectar.

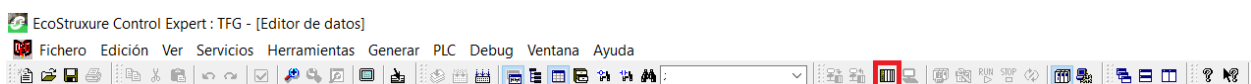


Figura 5.8 Conexión PC - Autómatas

Entonces, se activarán los símbolos de la izquierda, que significan, de izquierda a derecha, exportar proyecto a un PLC o importar un proyecto desde un PLC.

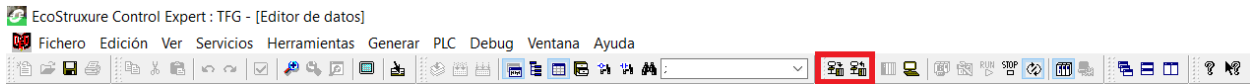


Figura 5.9 Transferencia de proyecto PC - Autómata

Cuando se le da al de la izquierda, para transferir el proceso al autómata, saldrá una ventana como la siguiente, donde se podrá activar la casilla “Ejecución del PLC después de la transferencia”.

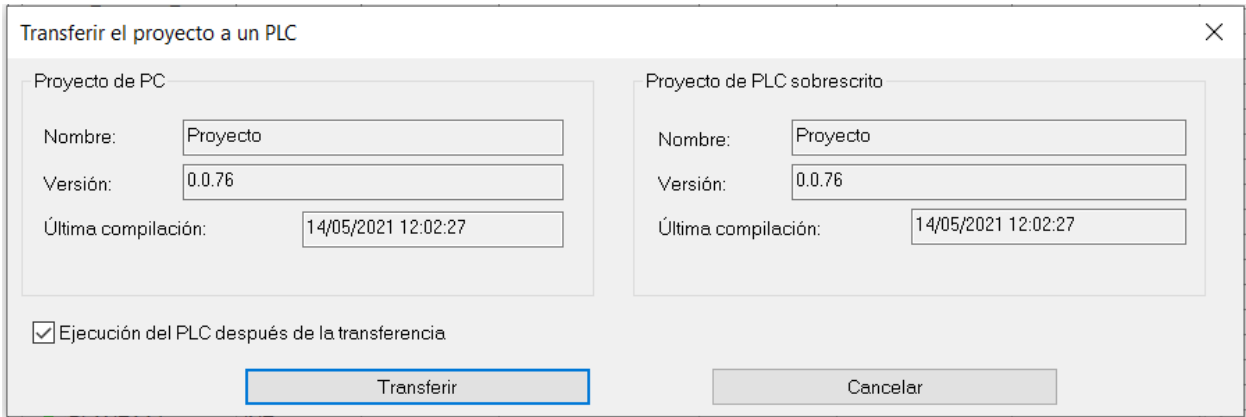


Figura 5.10 Transferencia de proyecto a PLC

Una vez acabe el proceso, se podrá comprobar que todo ha ido bien ya que saldrá luces verdes y rojas al lado de las secciones activas e inactivas, respectivamente.

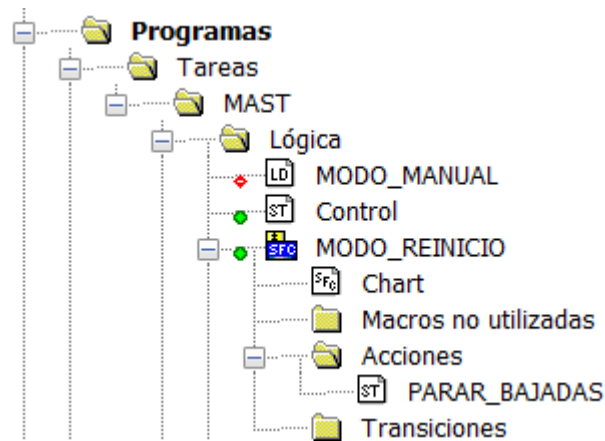


Figura 5.11 Indicadores de activación de secciones

Después de esta comprobación, el autómata debería estar funcionando.

5.2 OPC UA Server Expert

‘OPC UA Server Expert’, como ‘Ecostruxure Control Expert’, es un software diseñado por ‘Schneider Electronics’, evolución del clásico OFS (‘OPC Factory Server’), pero con las ventajas, ya expuestas en el Capítulo 2 de esta memoria.

Las especificaciones mínimas del sistema operativo necesarias para este software son:

MINIMO	
Plataforma	32 o 64 bits
RAM	4 Gb

Tabla 5.2 Requisitos OPC UA Server Expert

5.2.1 Programas

Al instalar ‘OPC UA Server Expert’, además del gestor de licencias, se nos instalarán tres programas que nos servirán para gestionar y configurar el sistema:



Figura 5.12 Programas OPC UA Server Expert

5.2.1.1 OPC UA Server Expert Configuration Tool

Es el encargado de la configuración del servidor OPC UA. Nos da las herramientas para conectar dicho servidor a una dirección IP conocida, importar una tabla de variables, configurar la seguridad, etc.

En el siguiente Capítulo se encuentra una explicación mucho más en profundidad, ya que se explicará, de forma práctica, como configurar el servidor al PLC y al SCADA al mismo tiempo.

5.2.1.2 OPC UA Server Expert

Es el eje central del servidor, encargado de lanzarlo. Genera un enlace que será el que habrá que proporcionarle al cliente para que se conecte.

En la pestaña ‘Diagnostics’ podemos ver mensajes del sistema:

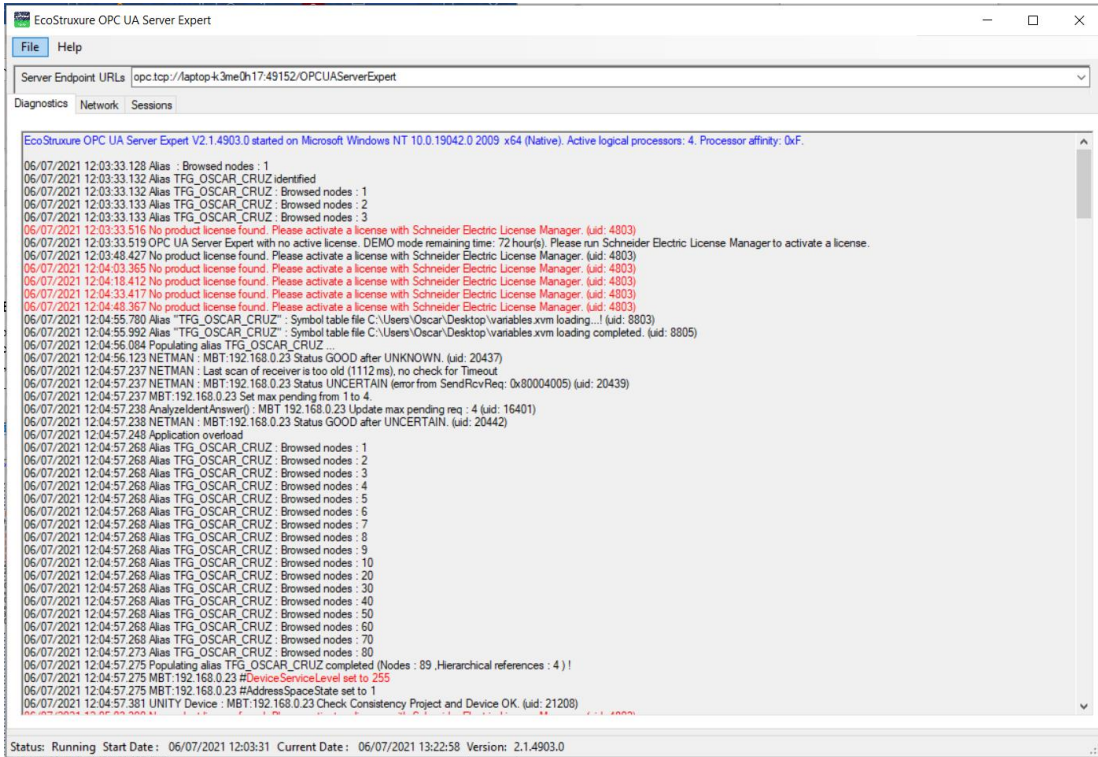


Figura 5.13 Diagnostics OPC UA

Mientras que en la pestaña ‘Network’ podemos observar algunos otros datos del sistema, como el número de nodos o ítems monitorizados:

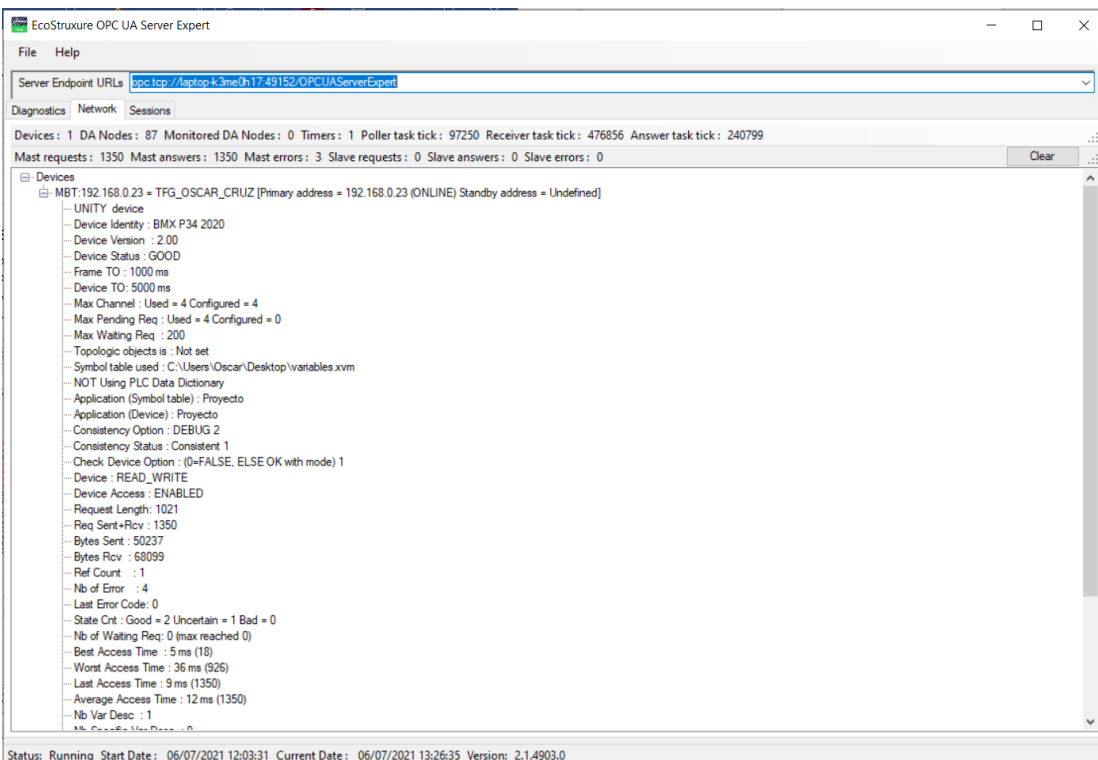


Figura 5.14 Network OPC UA

La estructura de la dirección URL del servidor será:

`opc.tcp://Server_IP_Address:Port_number/OPCUAServerExpert`

5.2.1.3 OPC UA Server Expert Sample Client

Por último, el programa trae un cliente de ejemplo que, en caso de querer hacer una prueba, haría la función del SCADA. Al abrirlo nos encontramos con un cliente vacío.

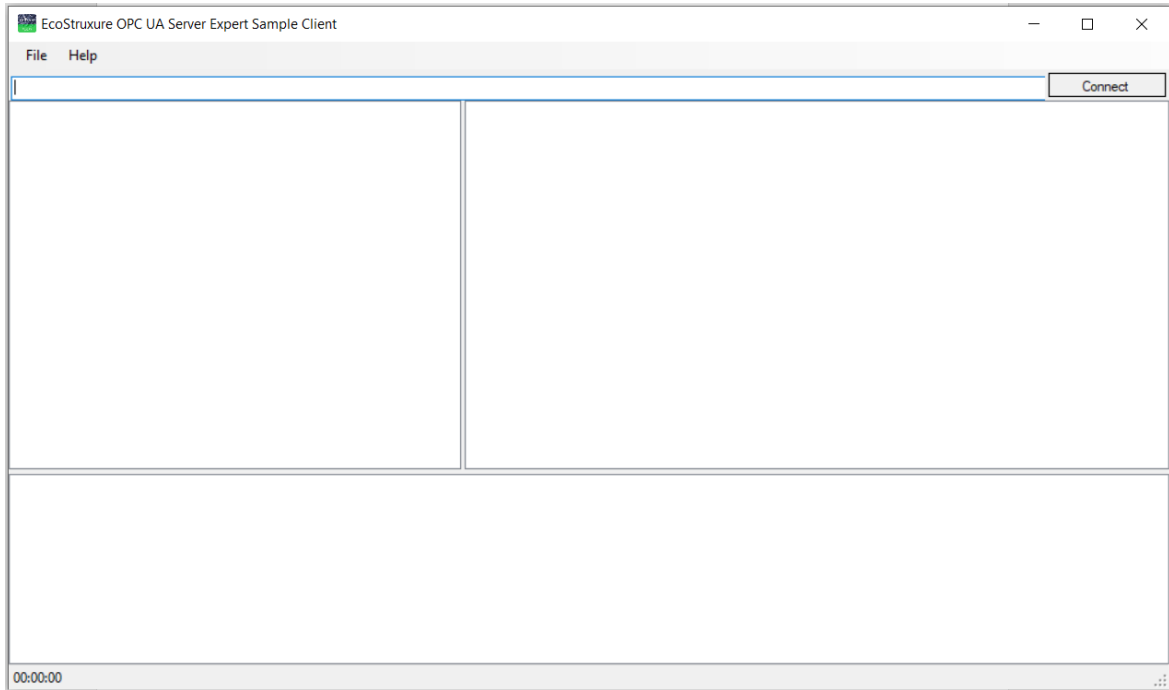


Figura 5.15 OPC UA Server Expert Sample Client

Al introducirle la dirección del servidor, aparecerá una ventana para crear una sesión. Aceptando, cliente y servidor se conectarán.

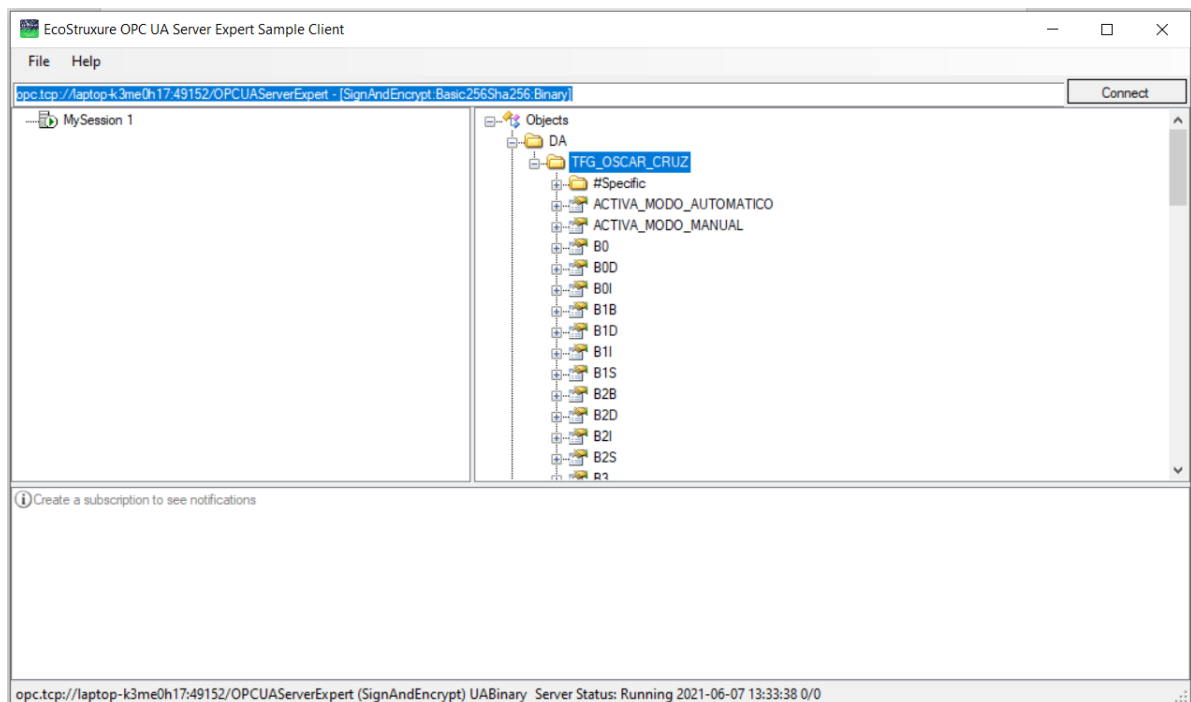


Figura 5.16 OPC UA Server Expert Sample Client II

Haciendo clic derecho y dándole a ‘View attributes’, podemos ver el valor de la variable que deseemos, además de otros datos.

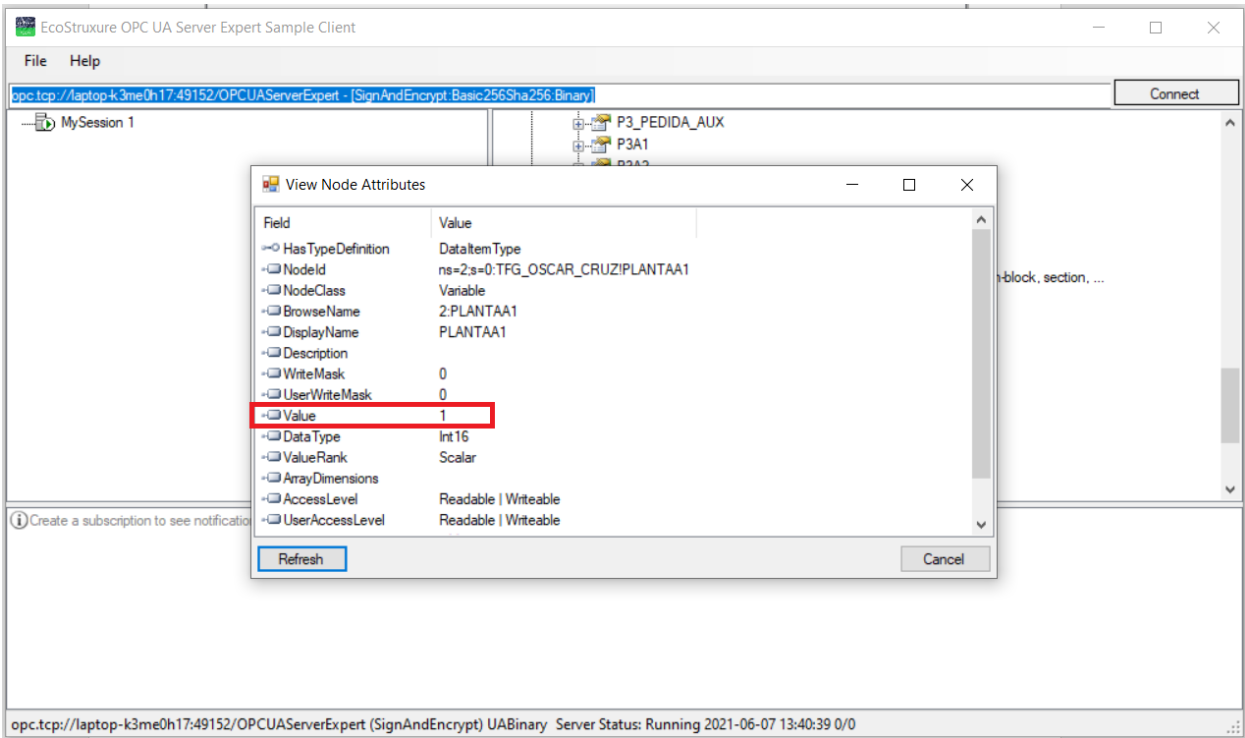


Figura 5.17 OPC UA Server Expert Sample Client III

En ese mismo menú al darle al clic derecho nos encontramos otras opciones como ‘Read’, donde también podremos leer la variable, o ‘Write’, en el caso de que queramos modificarla.

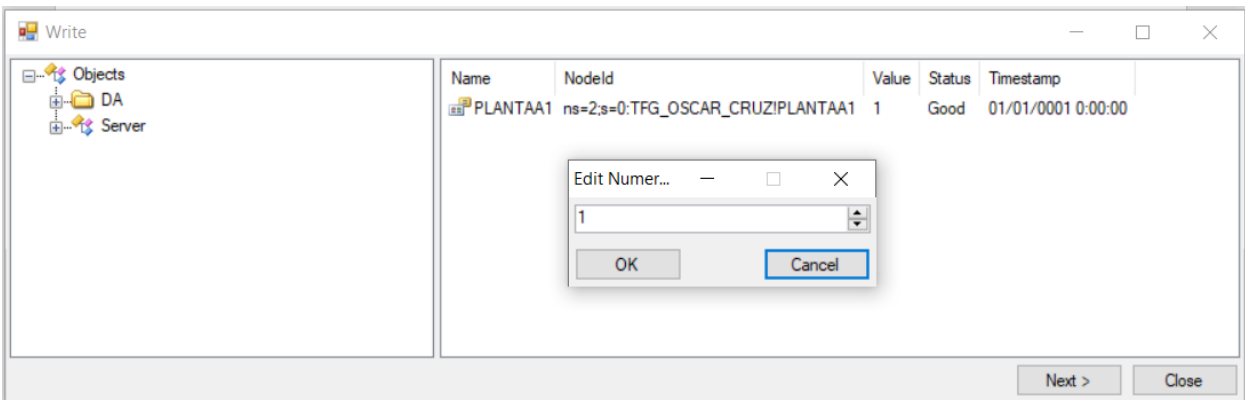


Figura 5.18 OPC UA Server Expert Sample Client IV

Sin embargo, aunque ya se ha visto que a través del cliente ejemplo de ‘OPC UA Server Expert’ es posible hacer una lectura y escritura total de las variables del PLC, es necesario un sistema con una interfaz apropiada, que nos permita entender el sistema de una forma visual.

Ante esa necesidad surge el SCADA, el sistema que funcionará como cliente del servidor OPC UA y que, además de facilitar el trabajo de lectura y escritura del autómatas, ofrece un sinfín de posibilidades.

5.3 Citect Studio

5.3.1 Introducción

Como ya se dijo en la introducción de este proyecto, un SCADA es un sistema de supervisión y control de sistemas automatizados, que surge ante la necesidad de centralizar información para el control y toma de decisiones con respecto a dicho sistema.

Además de lo ya comentado, otras funciones propias del SCADA son:

1. Activación de alarmas ante ciertas circunstancias.
2. Historización de valores.
3. Graficación de valores.
4. Registro y restricción de accesos; es decir, permitir a ciertos usuarios realizar una acción y tener control de quien la realiza.
5. Realización de acciones sobre el ordenador/servidor en el que está instalado.

Un diagrama básico de funcionamiento, donde se muestran todas las posibilidades de un sistema SCADA es el que se encuentra en la página web del centro de formación técnica para la industria Aula 21:

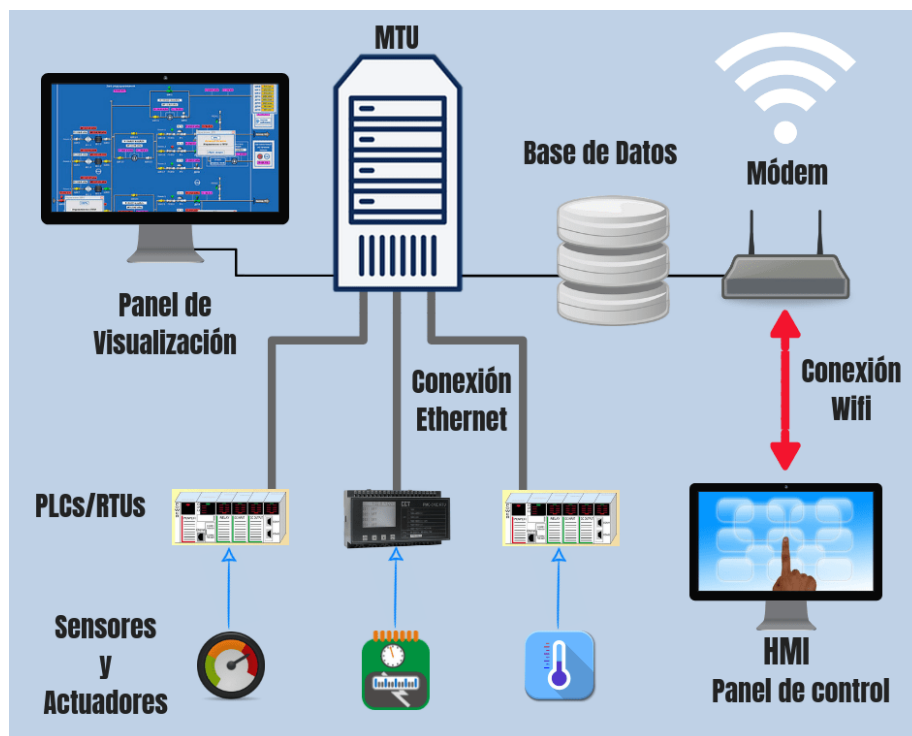


Figura 5.19 Diagrama de funcionamiento SCADA

En él se observa como los PLC o RTUs (Unidades Terminales Remotas) envían, a través de una conexión ethernet la información a ordenadores servidores SCADA, que procesan, distribuyen y muestran los datos al resto de la red.

Así, se puede tener, de forma remota, información a tiempo real del sistema, lo que permite tener un control preciso de la máquina y evitar errores humanos.

Hoy en día, muchísimos sectores, tanto públicos como privados, utilizan sistemas SCADA para controlar e incrementar la eficiencia gracias a los datos que se pueden obtener, así como para conseguir una más rápida comunicación de problemas del sistema para reducir el tiempo de inactividad. Algunos ejemplos son: el sector energético, alimenticio, de fabricación, petróleo y gas, etc.

5.3.2 Especificaciones

En este caso, se utilizará el programa Citect Studio 2018 R2 o simplemente Citect Studio, de ‘Schneider Electronics’, pues es una versión compatible con el servidor OPC UA.



Figura 5.20 Logo Citect Studio

Al ser un programa tan potente, este tendrá unos requerimientos mínimos para funcionar correctamente en el sistema en el que se instale. Estas especificaciones variarán dependiendo de si dicho SCADA está diseñado para ser cliente o servidor.

En la siguiente tabla se supondrá un servidor pequeño, llegando a poder ser las especificaciones más exigentes para servidores más grandes:

	CLIENTE	SERVIDOR
CPU PassMark*	2000	4500
Cores	2	4
RAM	4 GB	8 GB
Gráficos	DirectX 9, 128 MB de VRAM dedicada	
Resolución	1920 x 1080	

Tabla 5.3 Requisitos Citect Studio

*PassMark es una empresa encargada de hacer mediciones sobre CPUs relacionadas con la máxima velocidad de reloj. Estas pueden hacerse vía web.

Normalmente, un sistema SCADA requerirá del 25-50% de la CPU, dependiendo del nivel de gráficos del sistema, si es un sistema multi-monitor, etc. El programa también soporta una resolución 4K UHD (3840 x 2160), aunque con un mayor coste de CPU.

5.3.3 Programa

Citect Studio es un programa con una interfaz cómoda y fácil de utilizar. Todo el programa se estructura en base a un menú lateral, que permitirá al usuario moverse a través de todas las ventanas.

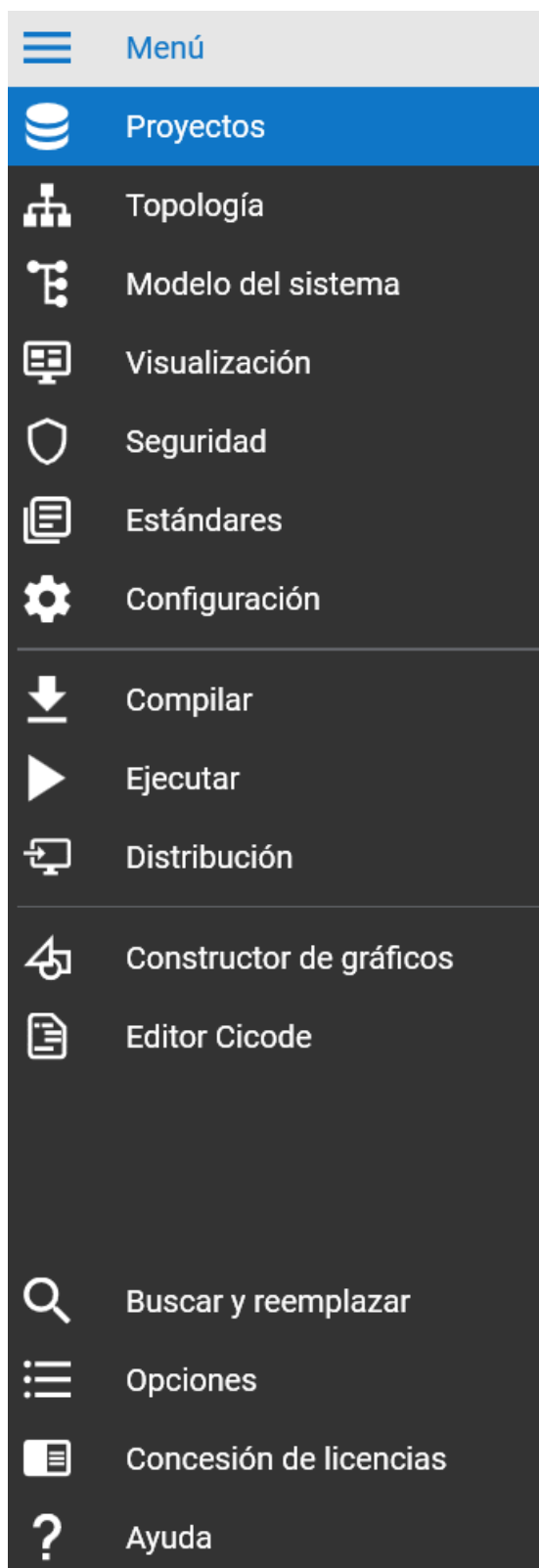


Figura 5.21 Menú de ventanas Citect Studio

A continuación, se mostrarán las ventanas más importantes:

5.3.3.1 Proyectos

Ventana que nos permite gestionar los proyectos existentes en el sistema. Se tendrá que decidir qué proyecto estará activo.

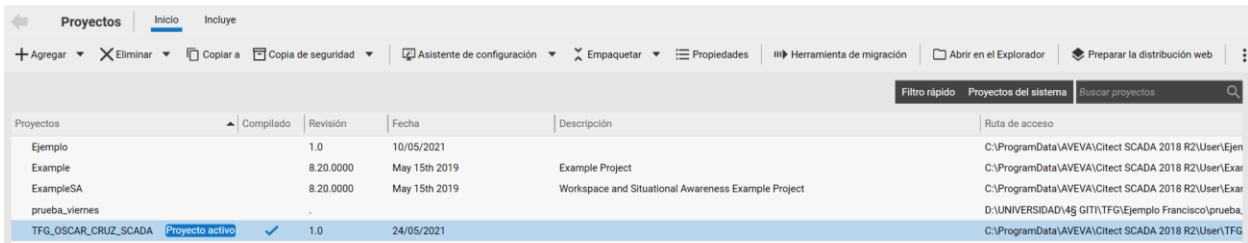


Figura 5.22 Ventana de Proyectos Citect Studio

5.3.3.2 Topología

Como su propio nombre indica, en esta ventana se podrá definir la topología del sistema; es decir, los equipos que lo conforman, además de los perfiles de usuario. Sin embargo, la pestaña más importante será ‘Dispositivos E/S’. En el siguiente Capítulo se explicará como configurar el servidor OPC UA en dicha pestaña haciendo clic en ‘Dispositivo nuevo’.

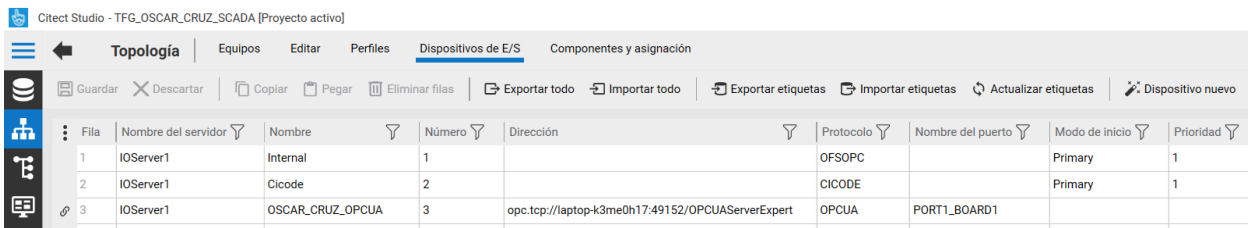


Figura 5.23 Ventana de Topología Citect Studio

5.3.3.3 Modelo del Sistema

Será la ventana en la que se definirán las variables y las alarmas del sistema, además de otros elementos como equipos con una configuración propia, acorde a nuestras necesidades. En el caso de una configuración OPC UA con importación de variables, como se verá en el próximo capítulo, estas nos aparecerán directamente en la pestaña ‘Variables’

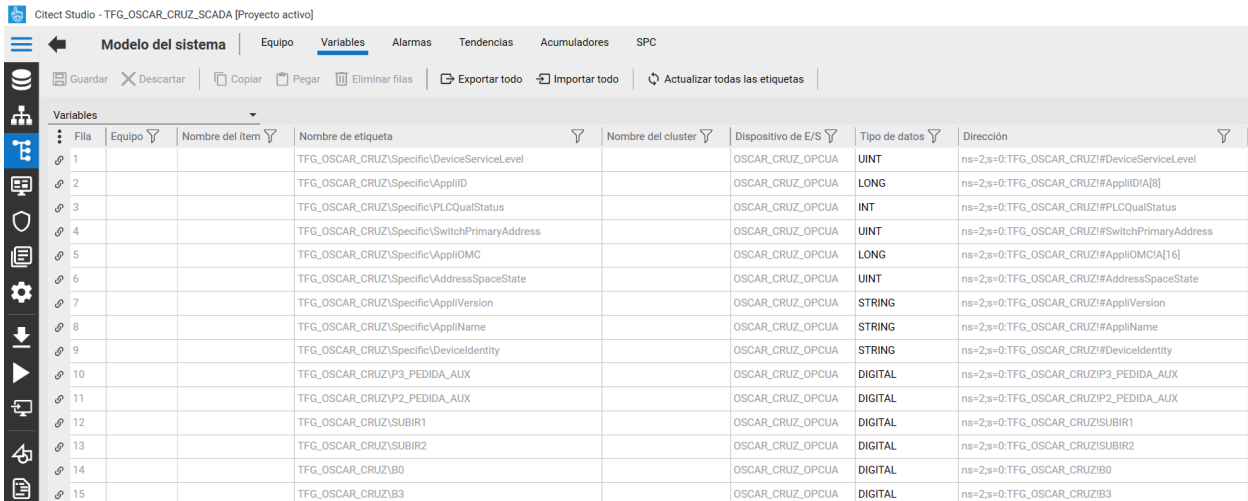


Figura 5.24 Listado de variables Citect Studio

5.3.3.4 Visualización

En esta ventana, en la pestaña ‘Paginas’, se puede gestionar las páginas creadas por el usuario para la visualización del sistema. En un principio vienen muchísimos ejemplos que pueden ser utilizados como guía, aunque también se pueden crear páginas desde cero, como en el caso de este proyecto, y que se explicará en el Capítulo 9 de esta memoria.

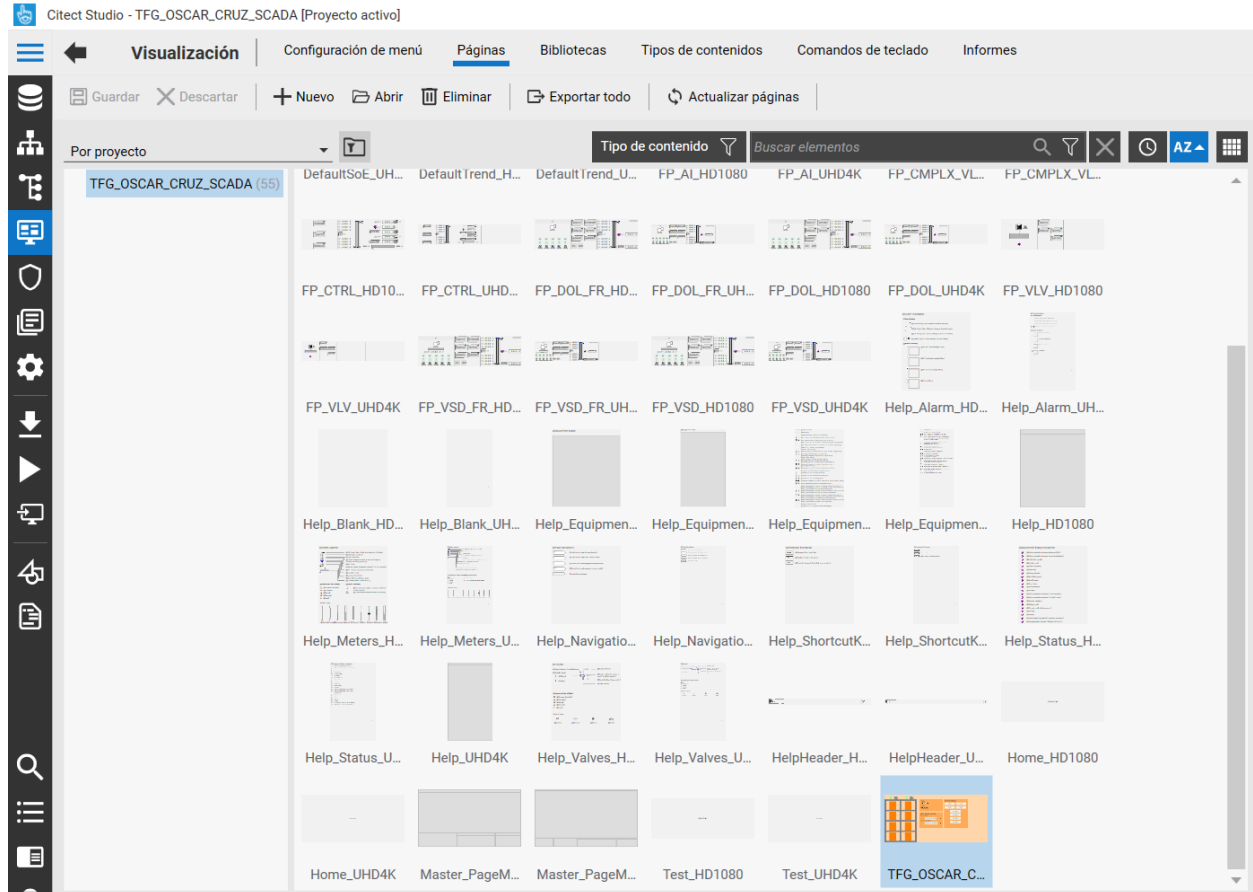


Figura 5.25 Ventana de Páginas Citect Studio

5.3.3.5 Seguridad

En el caso de querer añadir seguridad al proyecto o una jerarquía de usuarios, o si se quiere llevar un control de que usuario ha cambiado en algún momento el estado de la planta, esta ventana nos permitirá crear dichos usuarios y crear la mencionada jerarquía.

En el caso de este proyecto, al haber un solo usuario que es a su vez administrador, no ha sido necesario aplicarle ningún tipo de seguridad.

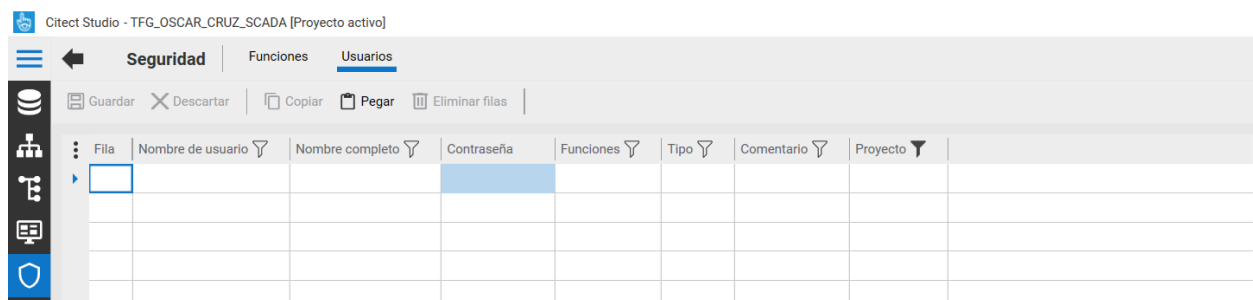


Figura 5.26 Ventana de Seguridad Citect Studio

5.3.3.6 Compilar

Citect Studio tiene un lenguaje de programación propio parecido a C llamado Cicode, aunque será muy poco usado en este proyecto. Cicode es un lenguaje compilado, por lo que será necesario compilar los proyectos para corroborar que no hay ningún error. Existe un botón para ello, como bien se ve en la figura. Una vez compilado, se podrá ejecutar el SCADA en el botón correspondiente.

Como es común en este tipo de programas, aunque este no esté compilado puede ejecutarse, ya que el propio sistema preguntará si desea el usuario que se compile el código.

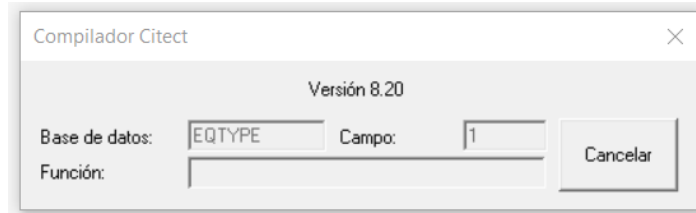


Figura 5.27 Compilación de variables Citect Studio

Cabe destacar que cuando se le dé al botón de compilar (y prácticamente cada vez que se pulse cualquier botón) saldrá el siguiente mensaje:

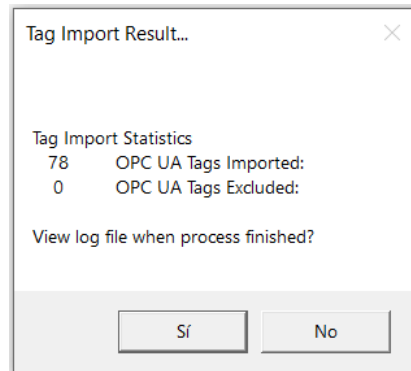


Figura 5.28 Importación de variables Citect Studio

Este mensaje quiere decir que las variables se han importado correctamente desde el servidor OPC UA y será una prueba muy importante de que todo funciona correctamente. El archivo log al que se refiere es un archivo .txt donde podrán verse datos y direcciones de las variables importadas. Algo parecido a lo que se puede ver en la pestaña 'Variables' de la ventana 'Modelo del sistema'.

5.3.3.7 Ejecutar

Al pulsar 'Ejecutar', aparecerá una ventana llamada 'Gestor de funcionamiento', que es el encargado de ejecutar los distintos procesos que existen dentro del proyecto (visualización, alarmas, tendencias, etc).

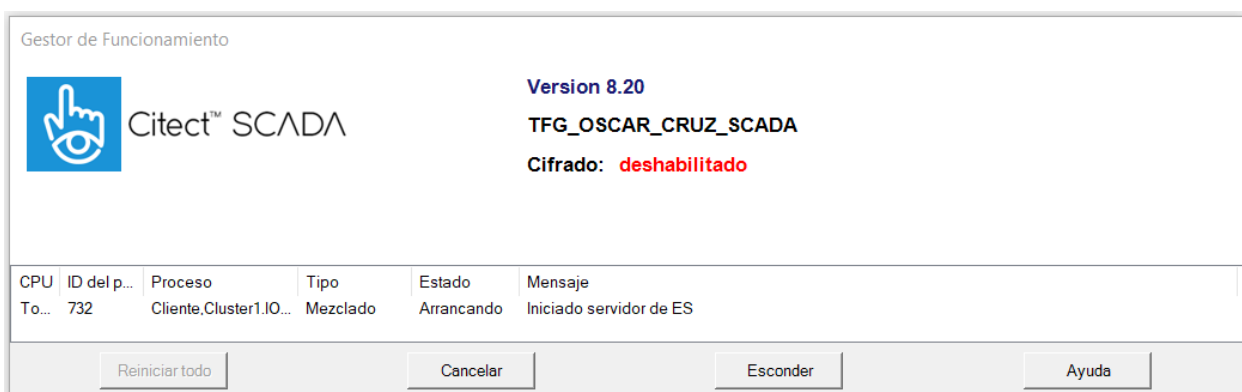


Figura 5.29 Gestor de funcionamiento Citect Studio

Posteriormente, se ejecutará el 'Page Menu', que preguntará, al no detectar licencia, si se quiere entrar en modo demostración:

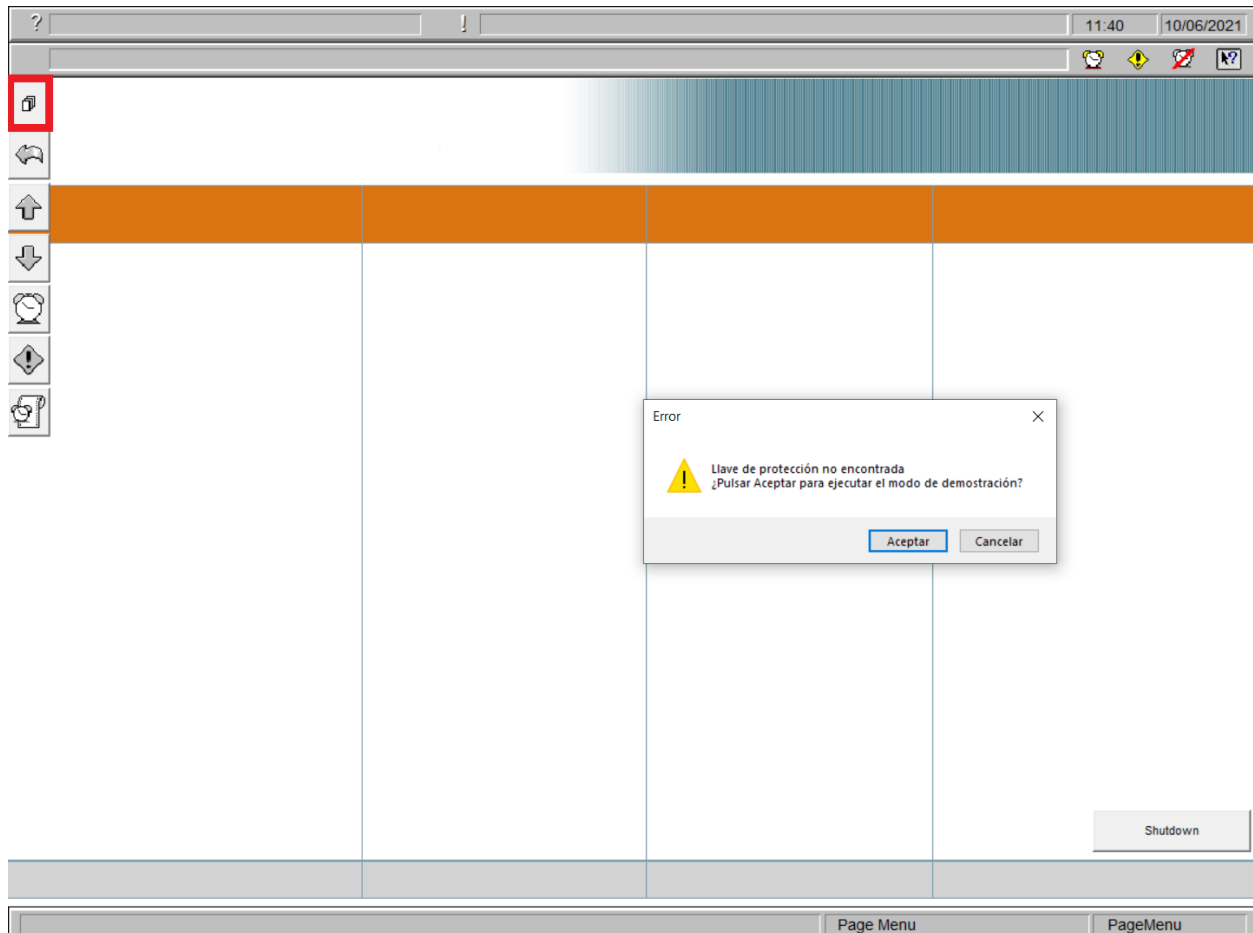


Figura 5.30 Page Menu Citect Studio

Finalmente, en el icono marcado en rojo en la imagen anterior puede elegirse la pantalla que se desea ver:

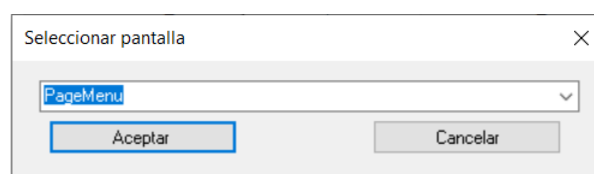


Figura 5.31 Elección de página Citect Studio

Una vez se elija, se ejecutará la visualización que el usuario haya diseñado.

5.3.3.8 Constructor de gráficos Citect

Será el eje central del programa y la ventana donde podrá diseñarse la visualización del SCADA. Se hablará de esta ventana con muchísimo detalle en el Capítulo 9, una vez que se tenga conectado el PLC al servidor OPC UA y el OPC UA al SCADA, que es lo que se verá en el próximo capítulo.

6 PROGRAMACIÓN

Una vez descrita la planta, incluyendo sus entradas y sus salidas, y el programa ‘Ecostruxure Control Expert’, se procede a la explicación del algoritmo utilizado para la programación de la planta. Se explicarán los modos de funcionamiento inspirados en la guía GEMMA y se hablará de la función de cada una de las secciones. Sin embargo, dichas secciones podrán encontrarse en los anexos.

Por otro lado, cabe destacar que en el punto 3.3 de esta memoria se describieron las bases de los algoritmos más comunes de funcionamiento de ascensores. No es difícil darse cuenta de que, en este caso, al tener nuestro ascensor dos botones intermedios, está orientado a hacer una programación con ‘maniobra colectiva de subida y bajada’.

6.1 Modos de funcionamiento

La guía GEMMA, creada en Francia en 1981, busca crear un estándar de automatización donde se definen los posibles estados por los que debe pasar un sistema automatizado.

Francisco Javier Ramos Ramirez, en su trabajo *Control y supervisión de célula de fabricación flexible* dedica el Capítulo 8 a la Guía GEMMA. Este documento se puede encontrar fácilmente en la biblioteca virtual de la Universidad de Sevilla y, en él, se incluye el siguiente esquema:

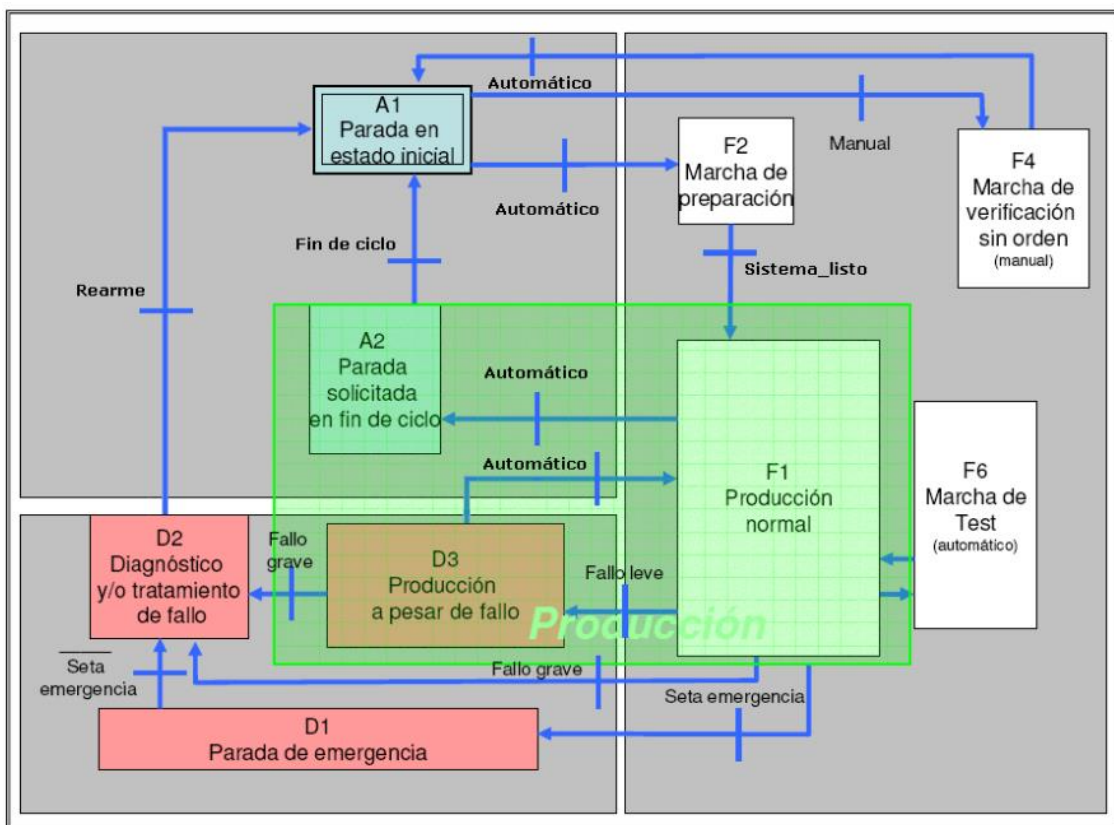


Figura 6.1 Esquema guía GEMMA

Como se ve, la guía tiene estados de producción, de preparación, de parada y de emergencia. Todos ellos se verán reflejados en el algoritmo.

6.1.1 Modo normal

Este será el modo en el que estará trabajando normalmente el autómatas, a menos que ocurra una emergencia. Estará compuesto de tres “submodos”, que se irán alternando en función de las necesidades del operario.

Estos tres modos serán:

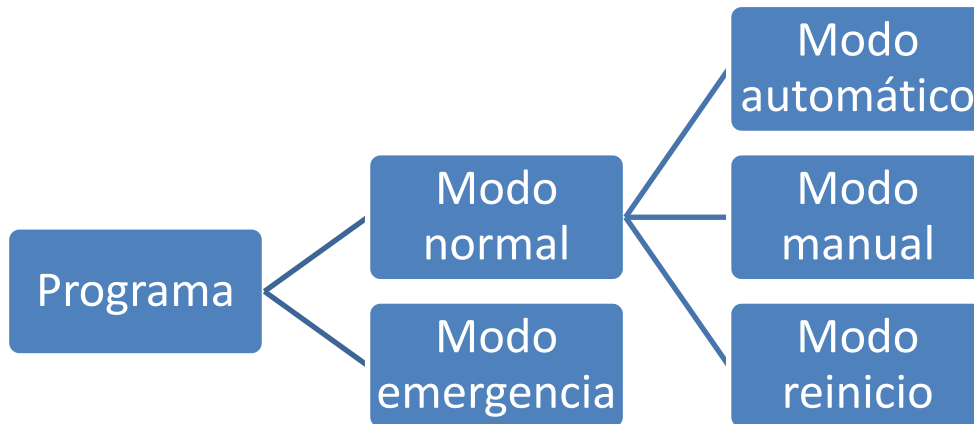


Figura 6.2 Modos de funcionamiento

6.1.1.1 Modo automático

Será el modo en el que estará el autómatas prácticamente la totalidad de su tiempo, ya que será el modo que implemente el algoritmo de ‘maniobra colectiva de subida y bajada’. Este modo, en realidad, estará implementado por una serie de secciones, que se verán activadas con la variable ‘VAR_MODO_AUTOMÁTICO’.

6.1.1.2 Modo manual

Todo sistema necesita de un modo manual para que el operario pueda manejarlo en caso de haber alguna incidencia. Este permitirá usar los actuadores de la planta a placer con tan solo pulsar un botón creado digitalmente. Se verá activado por la variable VAR_MODO_MANUAL.

6.1.1.3 Modo reinicio

Una vez que el operario ha manipulado el autómatas, se debe volver al modo automático, que es el verdadero modo de funcionamiento. Sin embargo, el modo automático suele estar diseñado con unas condiciones iniciales, por lo que se debe volver a ellas antes de regresar.

En este caso, las condiciones iniciales serán los dos ascensores en la planta 0. Por tanto, una vez que se pulse el botón de modo automático, en realidad se activará el modo reinicio, que bajará los dos ascensores y, entonces, dará paso al modo para el que el botón fue pulsado.

Se activará con VAR_MODO_REINICIO y, cuando acabe, activará una variable llamada FIN_REINICIO.

6.1.2 Modo emergencia

En cualquier momento puede pulsarse cualquiera de los botones “S” de cualquiera de los ascensores, o activar el modo emergencia a través del SCADA. Esto hará que todos los actuadores y todos los modos se paren, los buffers se reinicien y los SFC de movimiento vuelva al inicio.

Se podrá reactivar el sistema o bien pulsando el botón “A” de cualquiera de los ascensores, o bien activando el modo manual.

6.2 Algoritmos

En este apartado, se examinarán con el mayor detalle posible y ordenadas por modos de funcionamiento las secciones más representativas que conforman todo el programa. Se pretende que estas páginas sirvan como apoyo a los anexos para comprender la filosofía del algoritmo.

6.2.1 Modo normal

Este modo está gestionando por una sección el SFC llamada MODO_NORMAL. En los distintos estados activará variables condición como por ejemplo VAR_MODO_MANUAL, VAR_MODO_AUTOMATICO o VAR_MODO_REINICIO que ejecutarán o no las secciones correspondientes.

Para las transiciones, se tendrán en cuenta variables que se activarán por la pantalla de operador. Estas variables serán del tipo ACTIVA_MODO_XXX. La única excepción será la transición Reinicio – Automático, para la que se utilizará la variable FIN_REINICIO, que será activada por la propia sección del modo de reinicio.

6.2.1.1 Modo manual: sección MODO_MANUAL (LD)

Es el modo más sencillo de todos y estará compuesto exclusivamente por una sección.

Los botones de la pantalla de operador activarán distintas variables recogidas en el vector VAR_MANUALES. Cuando cada una de estas componentes se activen, activarán los correspondientes actuadores de la planta.

En la misma pantalla de operador también se harán las transiciones entre los distintos modos de funcionamiento del modo normal.

6.2.1.2 Modo reinicio: sección MODO_REINICIO (SFC)

Esta sección se activará con la variable VAR_MODO_REINICIO y su objetivo será llevar ambos ascensores a su posición inicial, que será la planta cero. Para ello, bajará ambos ascensores hasta que detecte que uno de ellos ha llegado a la planta baja. Entonces, seguirá bajando al otro hasta que llegue al mismo lugar.

Una vez que termine de bajar ambos, activará la variable FIN_REINICIO, que le dirá a la sección MODO_MANUAL (SFC) que ya se ha realizado el reinicio y que el sistema está preparado para entrar en modo automático.

6.2.1.3 Modo automático

Es el modo más extenso, ya que estará formado por una serie de secciones funcionando de forma concurrente que formarán un algoritmo cerrado que gestionará las decisiones y el movimiento de los ascensores.

- Control (ST): Es el “cerebro” del sistema. Es el encargado de leer los sensores de planta y en cual se encuentra el ascensor. También gestiona la llamada, la decisión de que ascensor irá a cada petición y la llegada. Para ello, usa un buffer de cinco componentes para cada ascensor, donde irá “apuntando” las plantas a las que debe ir cada uno. Cuando se produzca una llegada, borrará la casilla correspondiente.
- FLANCOS (LD): Activan ciertas variables cuando detectan un flanco de subida de los botones tanto interiores (cabina) como exteriores (petición de planta). Ejemplo: si el botón B0 tiene un flanco de subida, se activará FSB0 (flanco de subida botón 0 exterior). Esto se hace ya que el autómata está en un bucle constante muy rápido. Por ello, si no se usaran flancos de subida, podría detectar que se ha pulsado el botón un número indefinido de veces.

- **MOVIMIENTO_A1 (SFC) o MOVIMIENTO_A2 (SFC):** Siempre que la variable **MARCHA** este activada, el ascensor esperará a que sea llamado. Esto se producirá cuando la sección **Control (ST)** escriba una planta en la primera componente de su buffer, por lo que esta será la condición de movimiento. Entonces, comparará si la planta pedida es mayor, menor o igual que la actual y activará el actuador pertinente (**SUBIRX** o **BAJARX**). Una vez que llegue, parará durante 3 segundos para la apertura y cierre de puertas y volverá a la espera.
- **LUCES (LD):** La luz de cada planta se activará cuando la variable **PX_PEDIDA** correspondiente se active, siendo **X** el número de cada planta. Esta variable se activa cuando un botón de planta exterior es pulsado.
- **MODO_SEMIAUTOMÁTICO (ST):** Cada vez que se detecte un flanco de subida (Sección **FLANCOS**) en un botón virtual del **SCADA** del panel de modo semiautomático, se añadirá al buffer correspondiente la planta seleccionada. Así, se añadirán plantas “manualmente”, aunque el ascensor realizará la acción de forma automática.

Una línea temporal del modo automático sería:

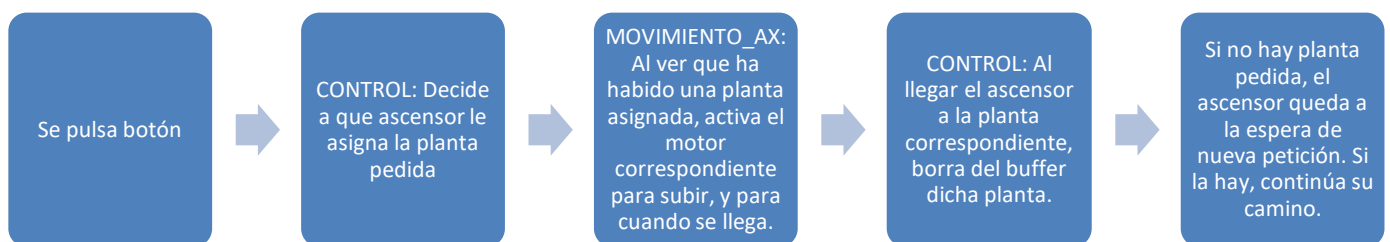


Figura 6.3 Funcionamiento modo automático

6.2.2 Modo emergencia

Este modo servirá para parar en seco el sistema en caso de emergencia. Estará conformado por dos secciones:

- **ACTIVA_EMERGENCIA (LD):** Será la encargada de gestionar los botones de emergencia, que en este caso serán los botones ‘**A**’ de ambos ascensores, o el botón del **SCADA**. Al pulsarlos, se activará la variable **VAR_MODO_EMERGENCIA**, que activará el modo. En cambio, si pulsamos algún botón ‘**S**’ de algún ascensor o el modo manual, se desactivará dicha variable y por tanto el modo emergencia.
- **MODO_EMERGENCIA (ST):** Si la variable **VAR_MODO_EMERGENCIA** está activa, desactivará manualmente los motores, el modo automático y reinicio. También se reiniciarán los buffers y los **SFC** de movimiento. Si en ese momento se activa el modo manual, se desactivará el modo emergencia.

7 CONEXIÓN PLC – OPC UA - SCADA

En este capítulo, abordaremos las conexiones realizadas entre el PLC y el SCADA. Para ello, necesitaremos un router genérico, que no tendrá otro papel que el de asignar direcciones IP a nuestros dispositivos. Este irá conectado a través de un cable Ethernet tanto al ordenador portátil en el puerto correspondiente como al puerto de mismo nombre del PLC, que se encuentra en el módulo BMX P34 2020.

7.1 Asignación de dirección IP al PLC

Un router, como ya hemos dicho, asigna dinámicamente direcciones a los dispositivos a los que se conecta. Sin embargo, para la conexión con el servidor OPC UA necesitamos que este tenga una IP estática, por lo que tendremos que configurarlo de tal modo.

Para ello, seguiremos los siguientes pasos:

7.1.1 Búsqueda de conexión IP

Necesitaremos, en primer lugar, encontrar tanto la dirección IP asignada a la computadora como la dirección IP del propio router, ya que luego tendremos que entrar en su interior. Para ello, nos iremos a las propiedades de la red conectada por cable y sin conexión (recordemos que no está conectada a ninguna red WIFI ni cable de red, tan solo lo necesitamos para que asigne conexiones).

Configuración de IP

Asignación de IP: Automático (DHCP)

Editar

Propiedades

Velocidad de vínculo (recepción/transmisión): 100/100 (Mbps)

Dirección IPv6 local de vínculo: fe80::88b:78fe:fc04:fb39%21

Dirección IPv4: 192.168.0.33

Servidores DNS IPv4: 192.168.0.1

Fabricante: Realtek

Descripción: Realtek PCIe GBE Family Controller #2

Versión del controlador: 10.25.119.2018

Dirección física (MAC): 48-BA-4E-AC-1E-19

Copiar

Figura 7.1 Configuración IP ordenador

Allí, vemos que las direcciones IP son las siguientes:

- IP Router: 192.168.0.1
- IP Ordenador: 192.168.0.33

Una vez tenemos la IP propia del router, tendremos que darle una dirección IP fija al autómata, para lo que tendremos que acceder a la configuración interna del router.

7.1.2 Acceso a router

Para ver la configuración interna, será suficiente con escribir su IP propia (192.168.0.1) en cualquier navegador. Una vez dentro, veremos una pantalla como la siguiente:

The screenshot shows the TP-Link web interface for a 150M Wireless Lite N Router. The browser address bar shows 192.168.0.1. The left sidebar contains a menu with 'DHCP' highlighted. The main content area is divided into sections: Status, LAN, Wireless, and WAN. The Status section shows Firmware Version: 3.12.11 Build 110831 Rel 60718n and Hardware Version: WR740N v4 00000000. The LAN section shows MAC Address: F8-D1-11-42-7F-22, IP Address: 192.168.0.1, and Subnet Mask: 255.255.255.0. The Wireless section shows Wireless Radio: Enable, Name (SSID): Router_Multiprocess, Channel: Auto (Current channel 6), Mode: 11bgn mixed, Channel Width: Automatic, Max Tx Rate: 150Mbps, MAC Address: F8-D1-11-42-7F-22, and WDS Status: Disable. The WAN section shows MAC Address: F8-D1-11-42-7F-23. A Status Help section on the right provides detailed information about the router's status and configuration options.

Figura 7.2 Interior del router

Haremos clic en DHCP, que nos permitirá asignar direcciones manualmente:

The screenshot shows the DHCP Address Reservation page in the TP-Link web interface. The browser address bar shows 192.168.0.1. The left sidebar contains a menu with 'DHCP' highlighted, and 'Address Reservation' is selected under the DHCP settings. The main content area displays a table of reserved IP addresses. Below the table are buttons for 'Add New...', 'Enable All', 'Disable All', and 'Delete All'. The table has the following data:

ID	MAC Address	Reserved IP Address	Status	Modify
1	00-80-F4-02-B4-61	192.168.0.4	Enabled	Modify Delete
2	F4-6D-04-E4-97-76	192.168.0.2	Enabled	Modify Delete
3	00-80-F4-02-B9-48	192.168.0.23	Enabled	Modify Delete

Below the table are buttons for 'Add New...', 'Enable All', 'Disable All', and 'Delete All'. There are also 'Previous' and 'Next' buttons at the bottom of the table. An 'Address Reservation Help' section on the right provides detailed information about the reservation process, including steps to reserve IP addresses and modify reserved IP addresses.

Figura 7.3 Direcciones IP asignadas manualmente

Como vemos, en el router de ejemplo ya hay algunas direcciones IP asignadas. Las dos primeras pertenecen a dispositivos que se usaban anteriormente con este router y la tercera sería el PLC asignado a la planta. Para añadirlo, debemos darle a Add New...

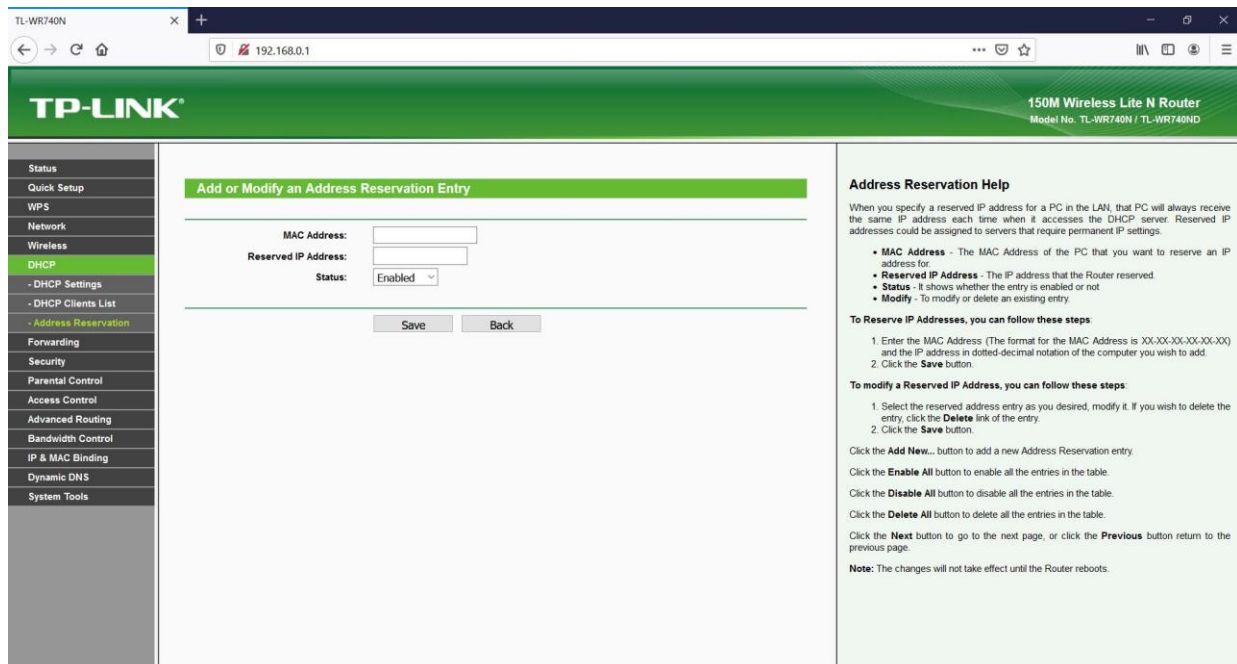


Figura 7.4 Añadir nueva dirección IP manualmente

Para añadir una dirección manualmente necesitaremos:

- Dirección MAC: identificador único, asignado por el fabricante, a cada una de las tarjetas red de cada dispositivo. Un mismo dispositivo podría tener dos tarjetas de red distintas y, por tanto, llegar a tener dos direcciones MAC.
- Dirección IP por reservar: Debemos escoger una dirección libre del router. Estas van desde 192.168.0.2 a 192.168.0.199.

En nuestro caso, la dirección IP que reservaremos, de forma arbitraria, será la 192.168.0.23. La dirección MAC de nuestro PLC podemos verla escrita en nuestro módulo BMX P34 2020.



Figura 7.5 Dirección MAC automática

7.1.3 Configuración Ecostruxure

7.1.3.1 Creación de red Ethernet

El primer paso será crear una nueva red Ethernet dentro de ‘Ecostruxure’:

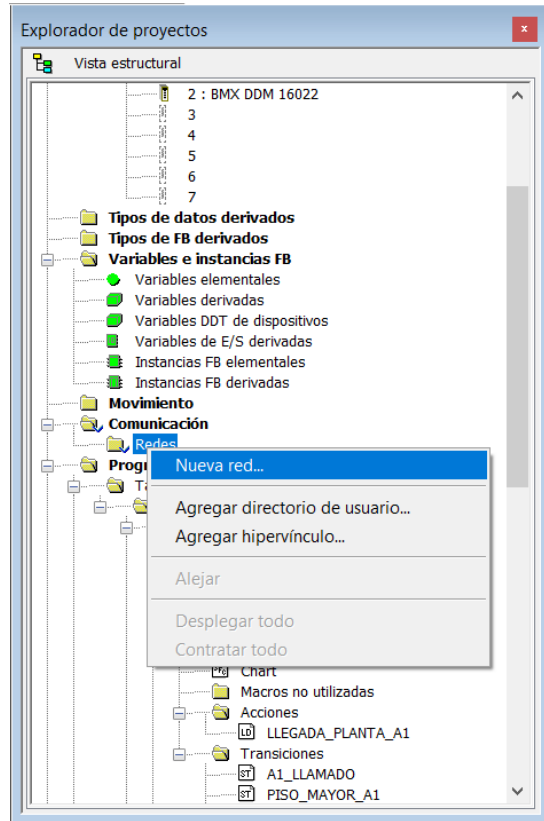


Figura 7.6 Nueva red Ethernet

Una vez le demos a ‘Nueva Red...’:

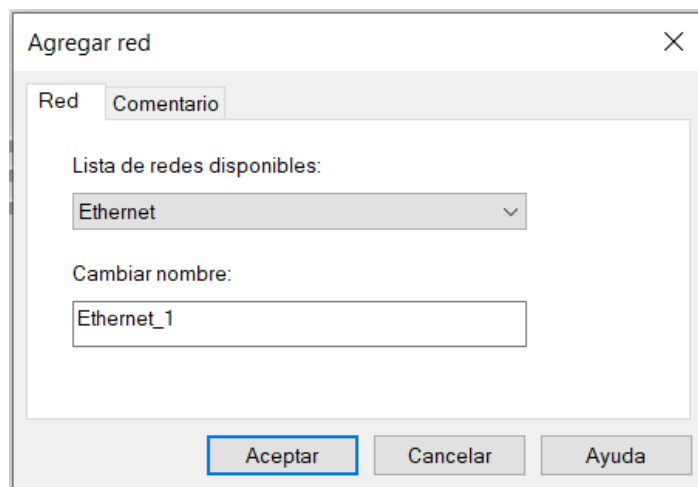


Figura 7.7 Nueva red Ethernet 2

7.1.3.2 Configuración de red Ethernet

Haremos doble clic en nuestra nueva red y nos aparecerá una pantalla como la siguiente:

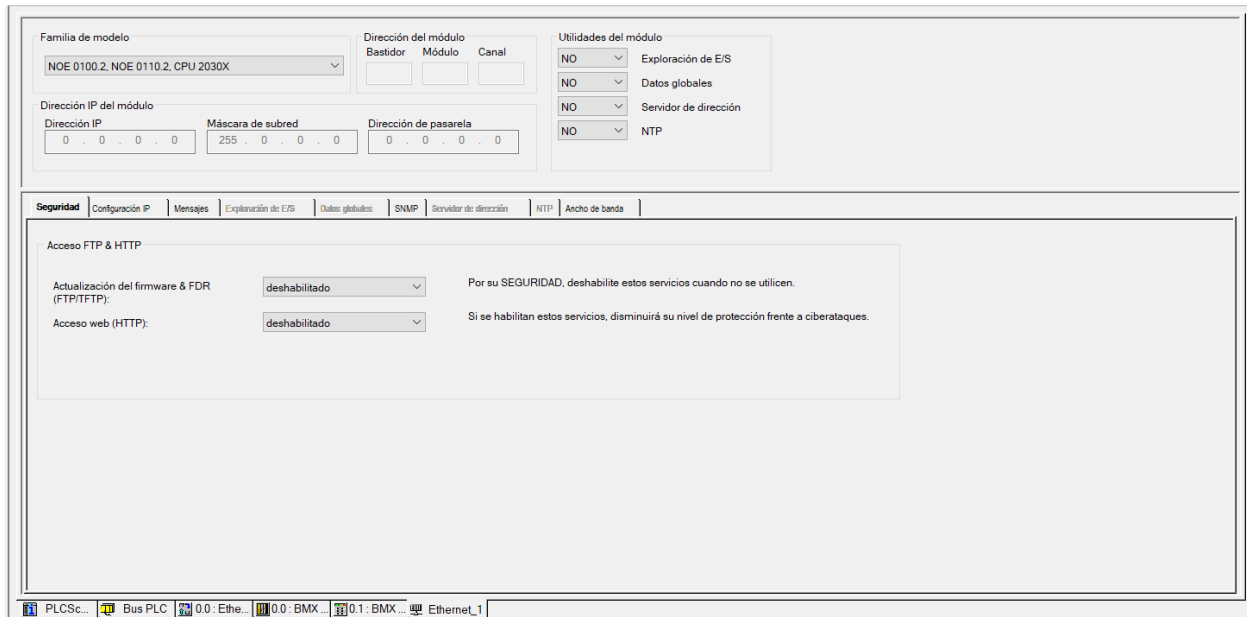


Figura 7.8 Pantalla de configuración Ethernet

Cambiaremos la familia de modelo a 'CPU2020, CPU2030 (>=V02.00), PRA 0100':

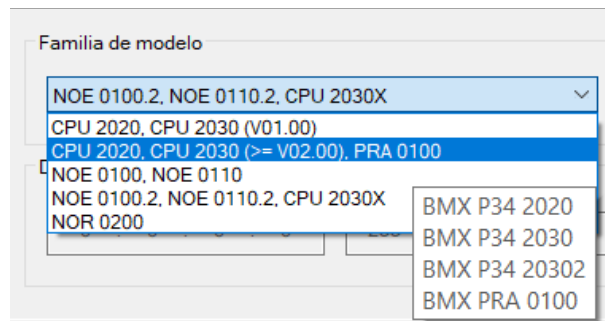


Figura 7.9 Cambio de familia de modelo

Nos saldrá la siguiente advertencia, que aceptaremos:

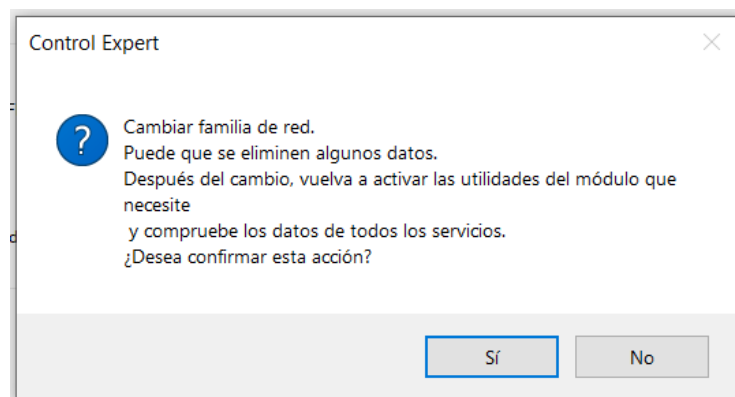


Figura 7.10 Advertencia cambio de familia

Clicamos en la pestaña de ‘Configuración IP’, y rellenamos los datos convenientemente:



Figura 7.11 Configuración IP

Es importante que posteriormente validemos cambios. Para ello, clicamos en el botón ‘Validar cambios’ de la barra superior.

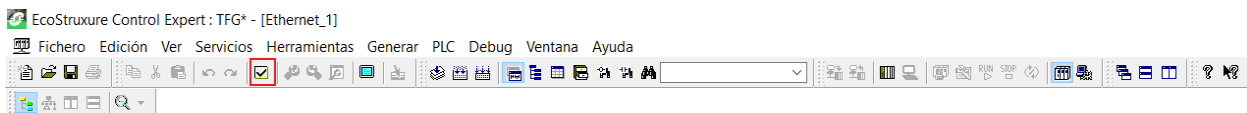


Figura 7.12 Validar cambios

7.1.3.3 Configuración de canal

Expandimos nuestro Bus PLC y clicamos en el puerto Ethernet de nuestro BMX P34 2020:

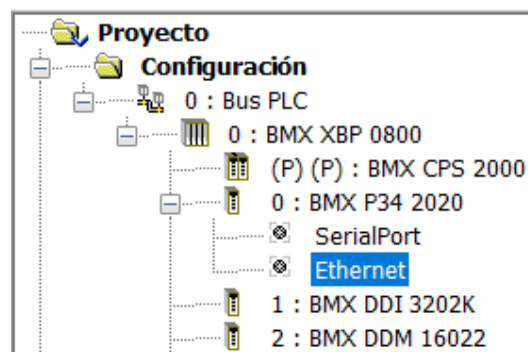


Figura 7.13 Puerto Ethernet del bus PLC

Así, accederemos a la configuración del puerto Ethernet, donde podremos configurar el canal que utilizaremos. Para ello:

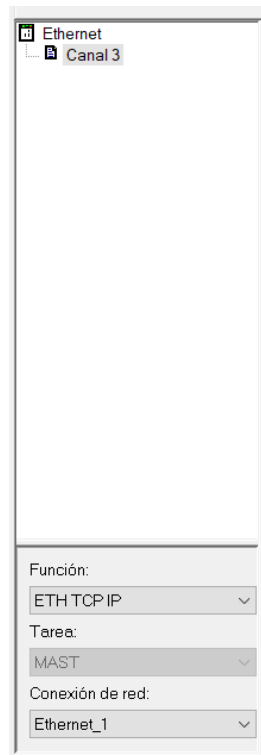


Figura 7.14 Configuración canal Ethernet

Entonces, validaremos cambios, conectaremos el PLC al ordenador vía USB y transferiremos el proyecto. Nuestro PLC ya tendrá asignada la dirección IP deseada.

Para comprobarlo, podemos observar el punto verde al lado del 'Canal 3':

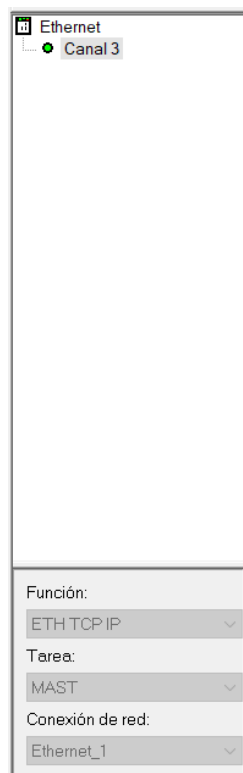


Figura 7.15 Configuración canal Ethernet II

7.2 Configuración servidor OPC UA

Como ya vimos, el OPC UA será el encargado de recibir las ‘publicaciones’ del PLC y permitir al SCADA su lectura. Esta comunicación se hará vía TCP / IP, que es la razón por la que asignamos, en el paso anterior, una dirección fija al PLC de la planta.

Existe una opción en la que el OPC UA puede directamente importar las variables del PLC, llamada ‘Diccionario de datos’. Sin embargo, esta opción no es compatible con nuestro dispositivo, por lo que tendremos que importar manualmente las variables de nuestro proyecto y entregárselas al servidor.

7.2.1 Exportación de variables Ecostruxure

Para importar variables de nuestro proyecto, seleccionaremos aquellas que queremos compartir, haremos clic derecho y le daremos a ‘Exportación seleccionada’:

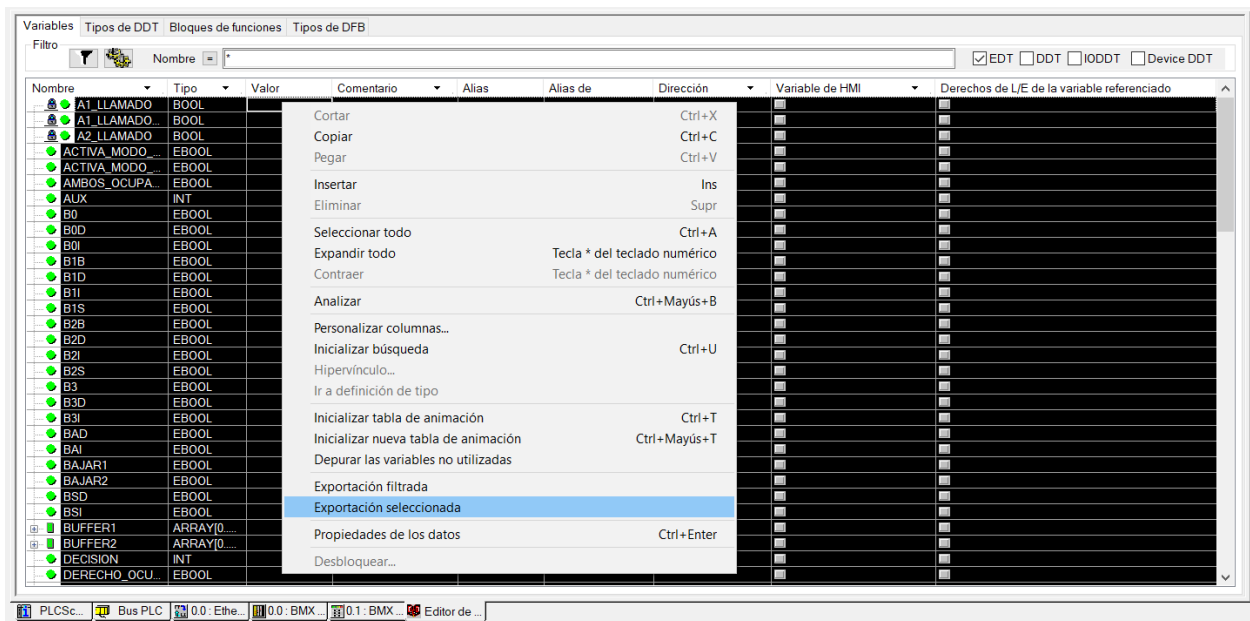


Figura 7.16 Exportación seleccionada

Necesitaremos que el fichero de salida esté en formato .xvm, por tanto:

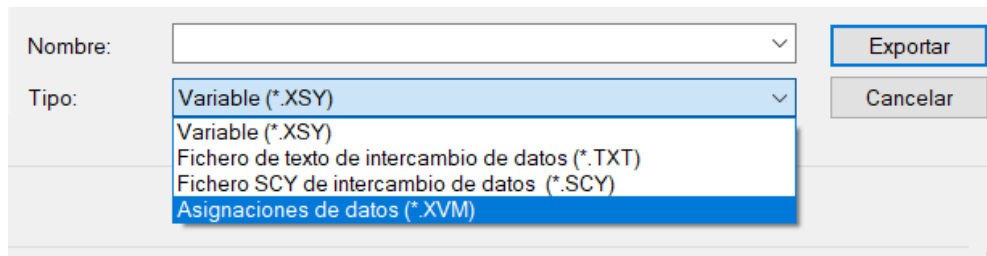


Figura 7.17 Formato exportación de variables

Así, conseguiremos un fichero .xvm que podremos entregar al servidor OPC UA para que sea capaz de leer y escribir valores sobre esa lista de variables.

7.2.2 Configuración OPC UA

7.2.2.1 Configuración IP

Al abrir el programa de configuración del OPC UA, ‘OPC UA Server Expert Configuration Tool’, aparecerá una pantalla como la siguiente:

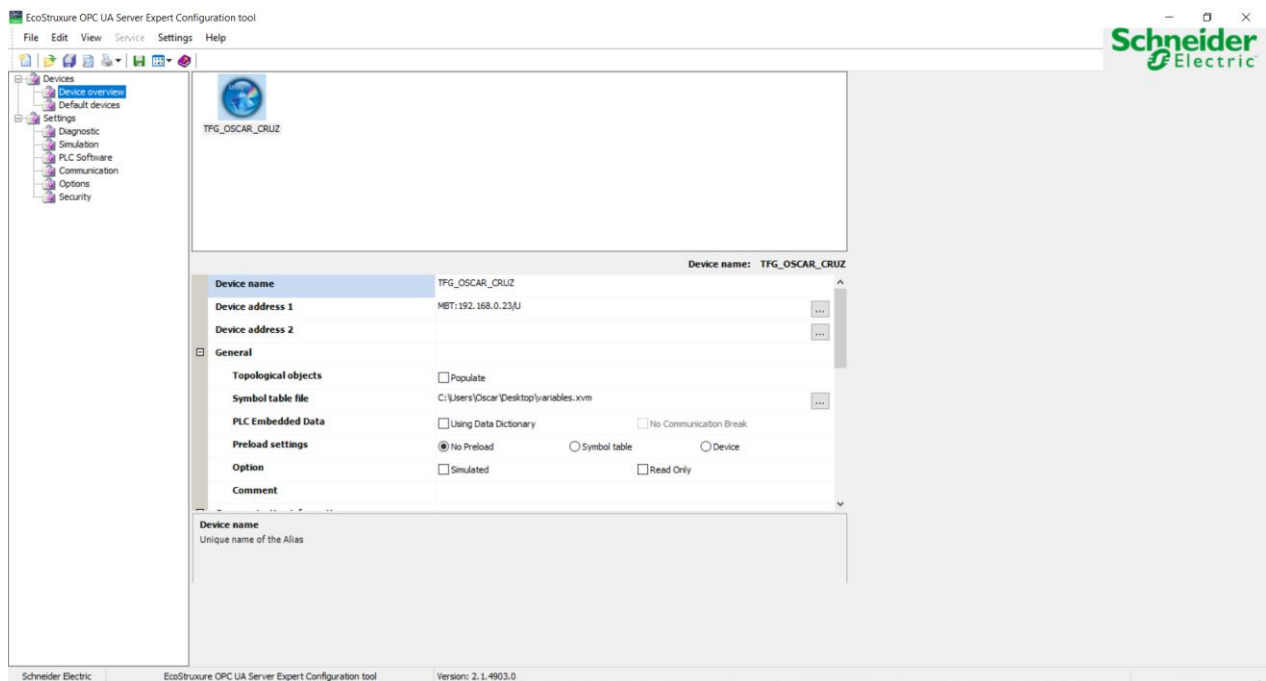


Figura 7.18 Visualización de la configuración del OPC UA

Primero, le pondremos un nombre a nuestro dispositivo y nos centraremos en el campo Device address 1. Clicando en los tres puntos de dicho campo:

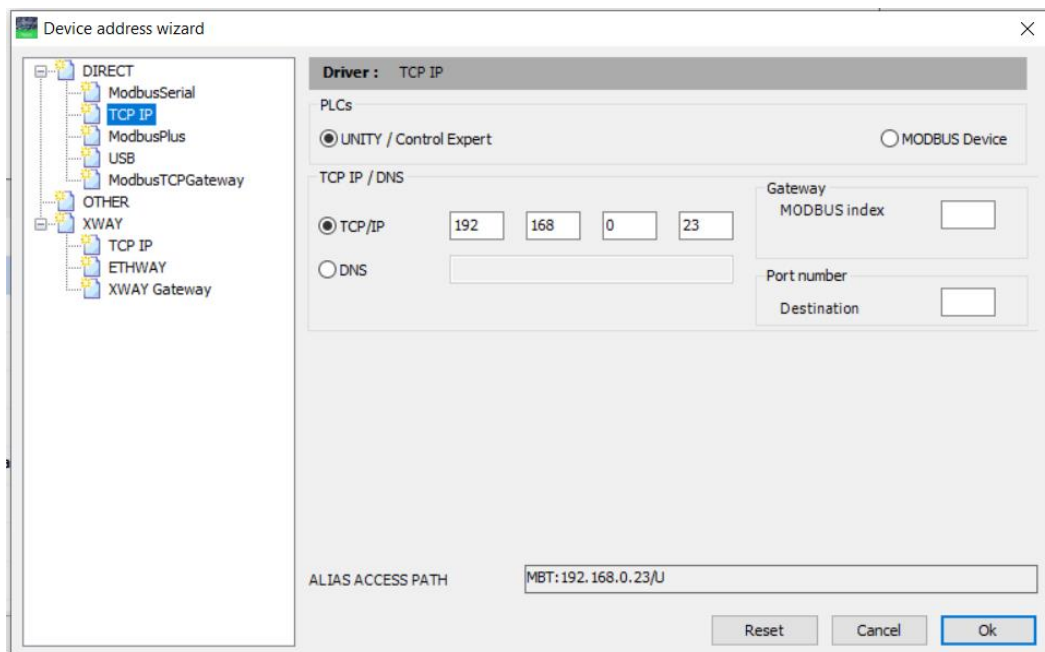


Figura 7.19 Configuración de dirección IP OPC UA

En esta pantalla, podremos configurar la dirección IP de nuestro PLC.

7.2.2.2 Importación de tabla de variables

En el campo ‘Symbol Table File’, podremos importar el fichero de variables .xvm que exportamos anteriormente del programa ‘Ecostruxure’:

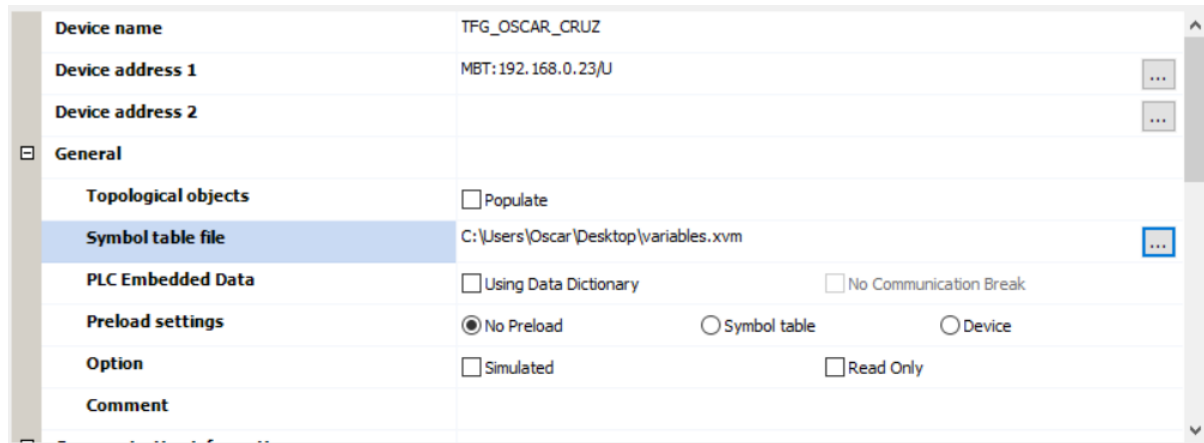


Figura 7.20 Importación tabla de variables

7.2.2.3 Desactivación de seguridad

Pese a que una de las grandes ventajas del OPC UA es su seguridad y su capacidad para crear usuarios, en este caso desactivaremos dichas opciones, ya que solo habrá un usuario. Para ello, en el panel izquierdo clicamos en la sección ‘Security’ y lo configuramos de la siguiente manera:

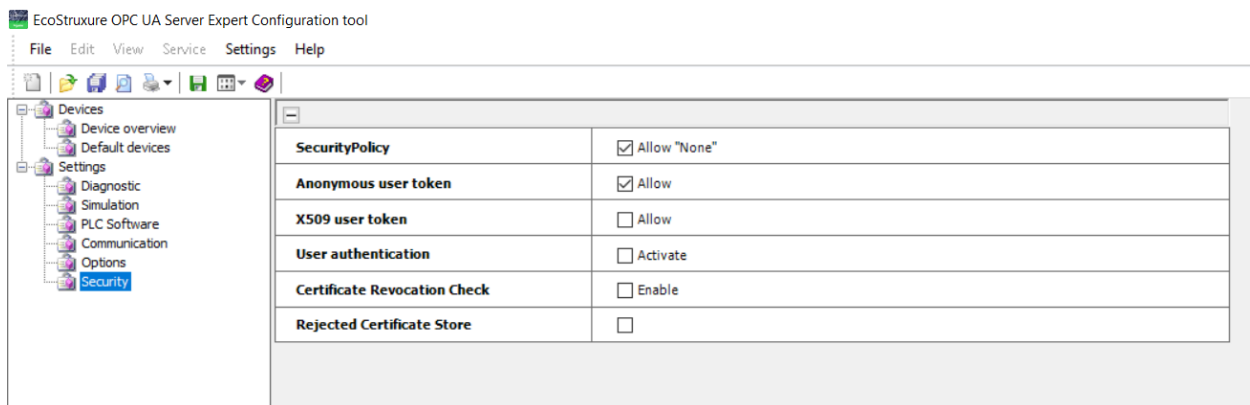


Figura 7.21 Configuración de seguridad OPC UA

Finalmente, guardaremos la configuración y abriremos el programa principal ‘OPC UA Server Expert’.

7.3 OPC UA Server Expert

Una vez configurado, abriremos el programa principal. En esta sección veremos que problemas podemos encontrar y que necesitamos para conectar este servidor con el proyecto de SCADA.

7.3.1 Dependencia de OPC UA con una dirección IP

Como se ha repetido varias veces en esta memoria, el OPC UA se caracteriza por su seguridad. Por ello, le otorga un certificado a la dirección IP en la que se instala y no admite una distinta. Entonces, puede ocurrir que, al abrir el programa, se muestre el siguiente mensaje:

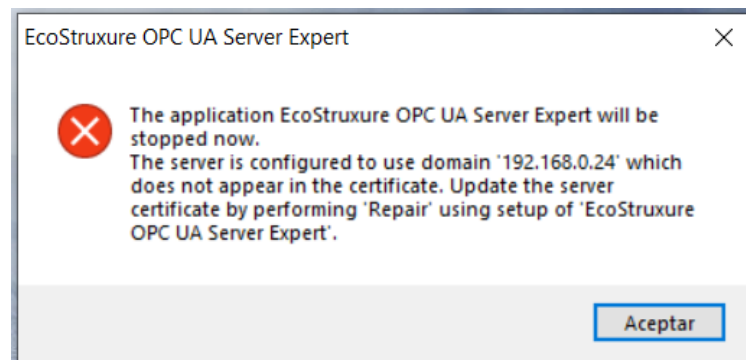


Figura 7.22 Error en OPC UA

Como vemos, la dirección 192.168.0.24 no se encuentra en el certificado, por lo que el programa nos indica que lo reparamos usando la configuración del 'Ecostruxure OPC UA Server Expert'.

Esto se refiere a que usemos el mismo ejecutable con el que instalamos el programa para repararlo:

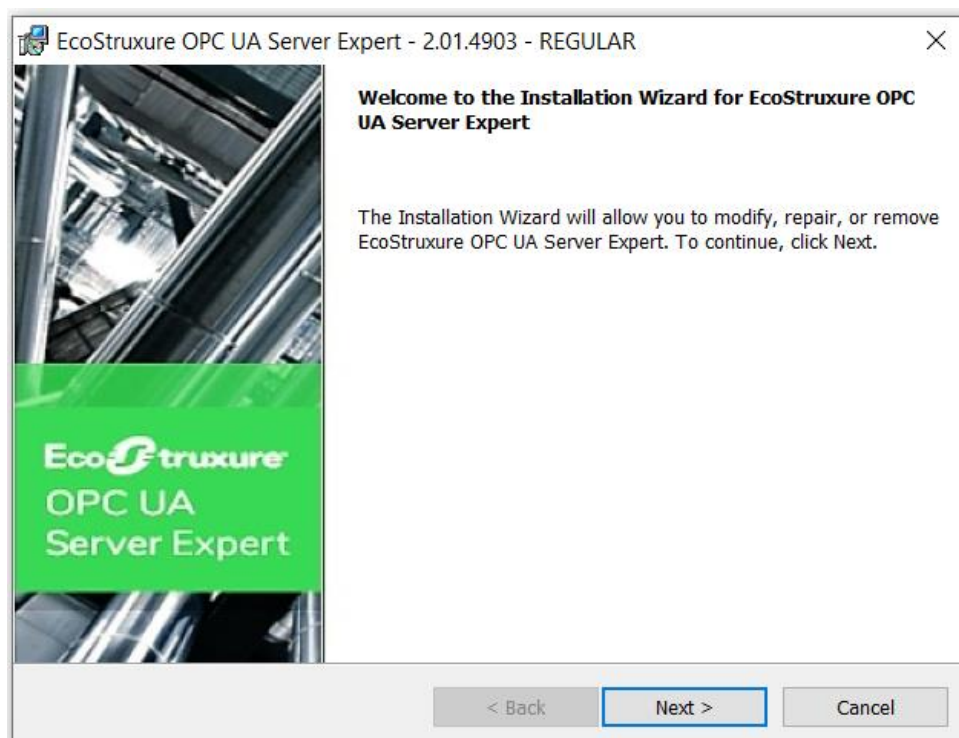


Figura 7.23 Reparación OPC UA Server Expert

Dándole a Next:

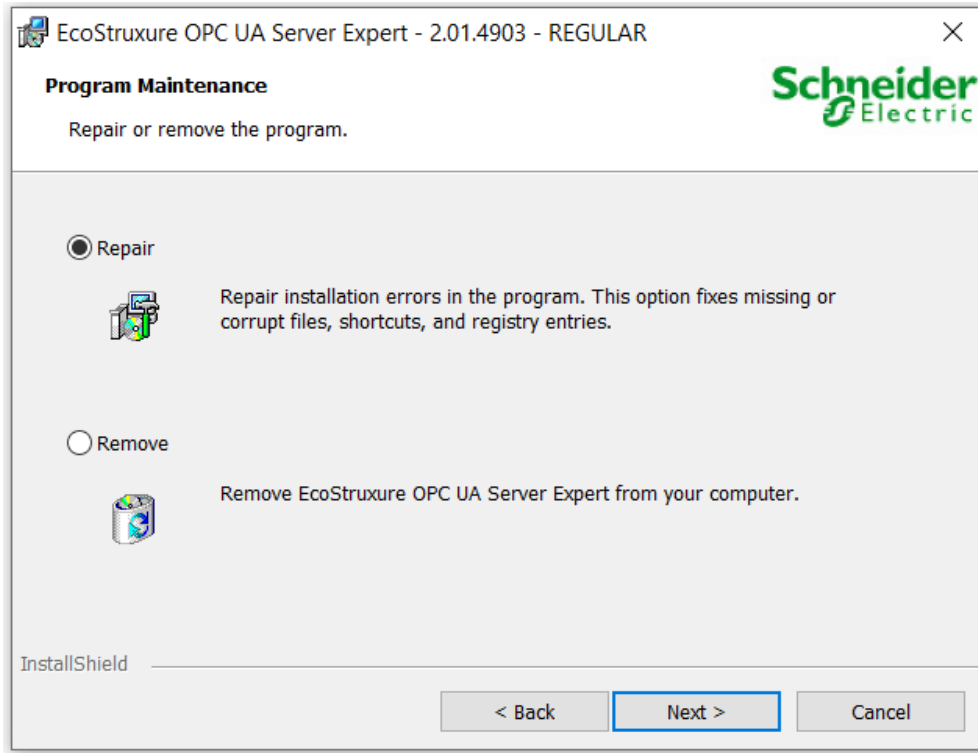


Figura 7.24 Reparación OPC UA Server Expert II

Siguiendo con la reparación:

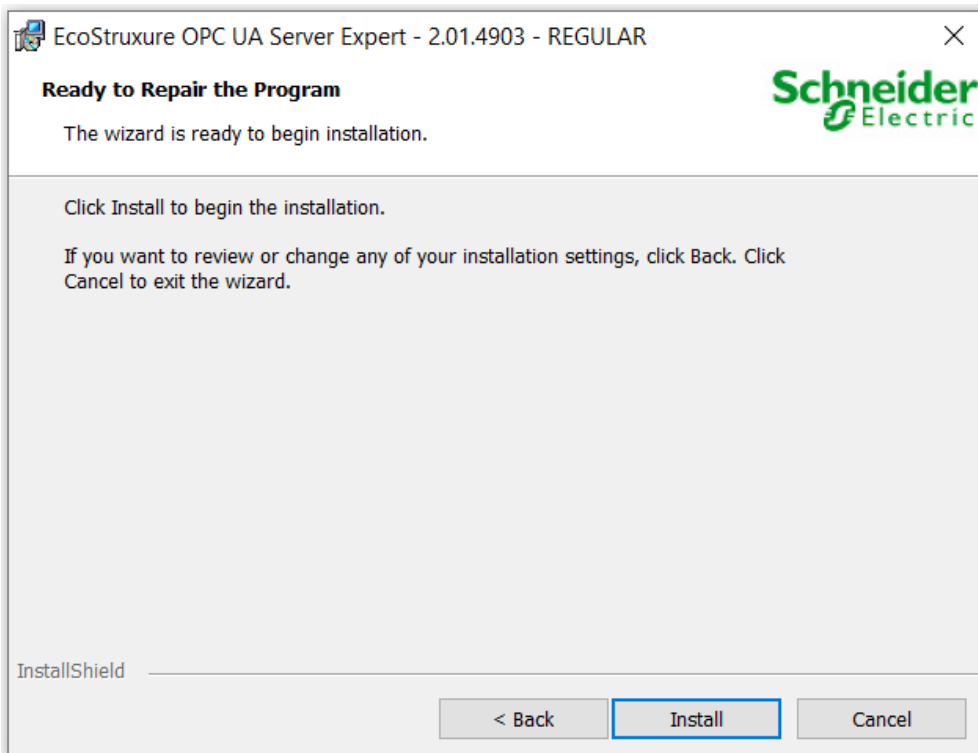


Figura 7.25 Reparación OPC UA Server Expert III

Finalmente, nos dará la opción de actualizar los certificados:

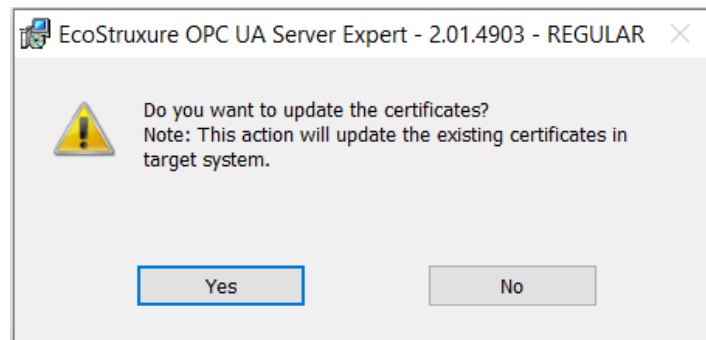


Figura 7.26 Reparación OPC UA Server Expert IV

7.3.2 Dirección URL de servidor OPC UA

El siguiente paso será copiar la dirección URL del servidor OPC UA. Esto nos será útil para indicarle al SCADA donde se encontrará la información del servidor.

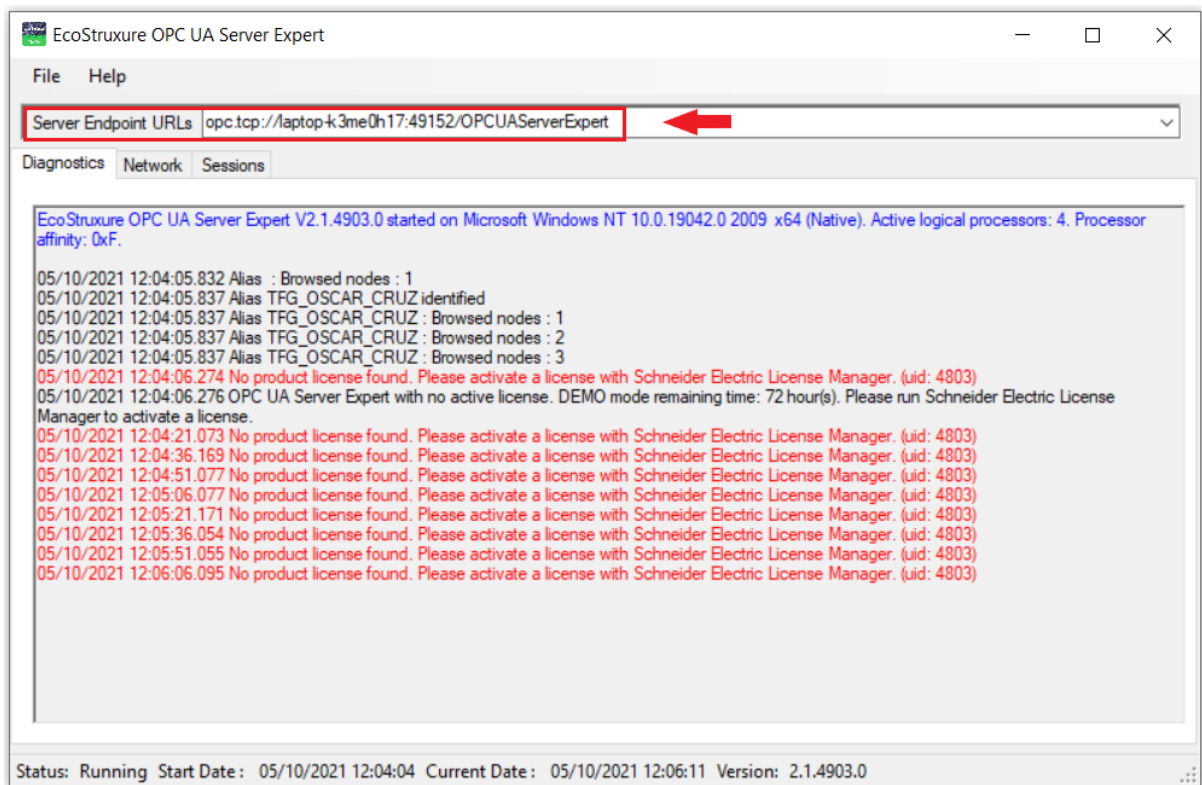


Figura 7.27 URL OPC UA

7.4 Citect Studio 2018 R2

Como ya se ha explicado anteriormente, el programa que usaremos para el diseño del SCADA será ‘Citect Studio 2018 R2’. Es importante utilizar esta versión, ya que por ejemplo la versión 2018 no incluye los drivers de comunicación con OPC UA.

7.4.1 Creación de un nuevo proyecto

Para crear un nuevo proyecto, le damos a Agregar -> Proyecto nuevo...

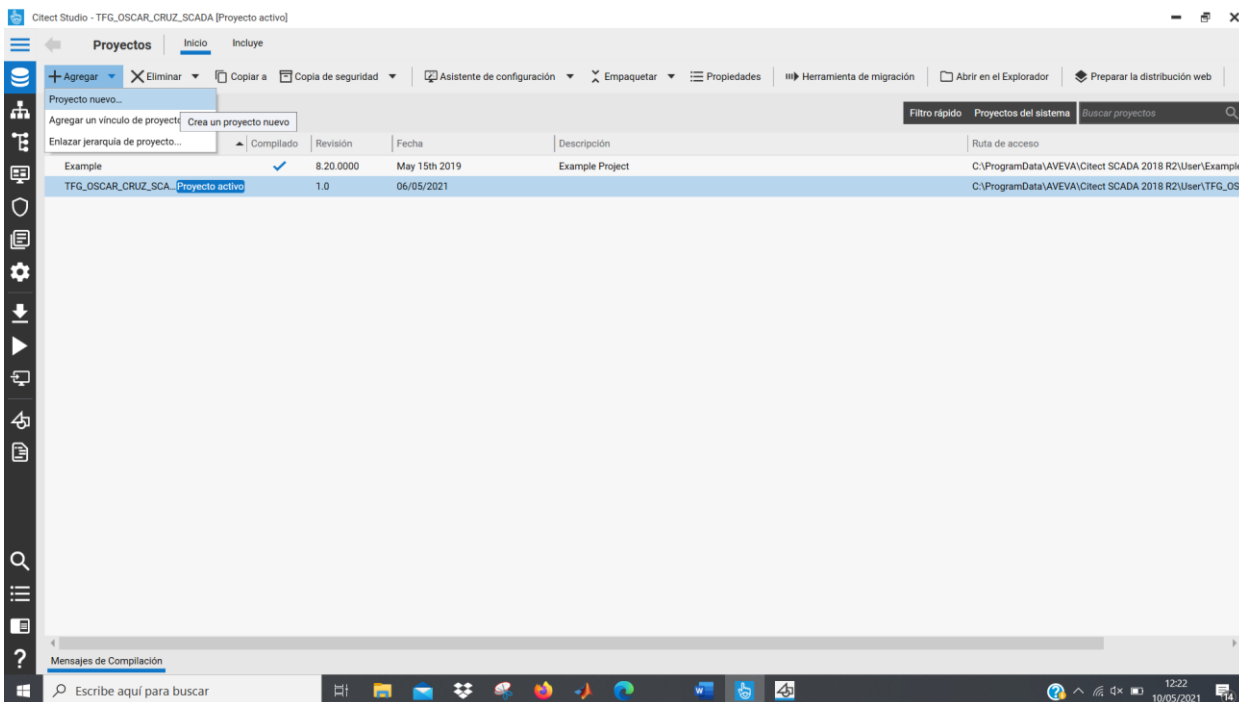


Figura 7.28 Creación nuevo proyecto SCADA

Nos saldrá una ventana como la siguiente, donde le daremos un nombre a nuestro proyecto.

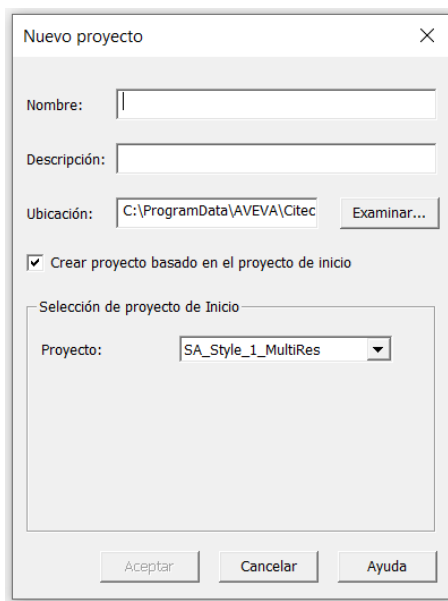


Figura 7.29 Creación nuevo proyecto SCADA II

Para terminar, activaremos nuestro proyecto:

Proyectos	Compilado	Revisión	Fecha	Descripción	
Example	Proyecto activo	✓	8.20.0000	May 15th 2019	Example Project
TFG_OSCAR_CRUZ_SCADA	ACTIVAR	1.0	06/05/2021		

Figura 7.30 Creación nuevo proyecto SCADA III

7.4.2 Configuración dispositivo E/S

En la pestaña de ‘Topología’, hacemos clic en ‘Dispositivos E/S’:

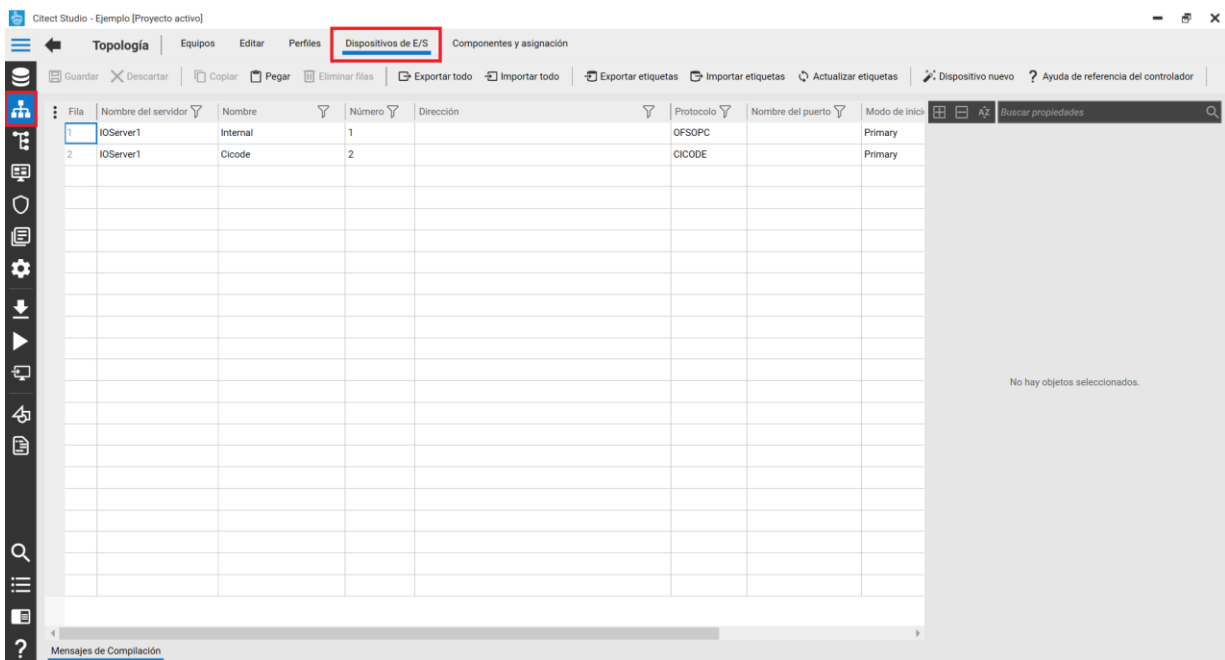


Figura 7.31 Topología SCADA

Una vez ahí, le damos a ‘Dispositivo nuevo’. Nos saldrá una ventana como la siguiente, donde seleccionaremos el proyecto donde queremos crear un nuevo dispositivo E/S:

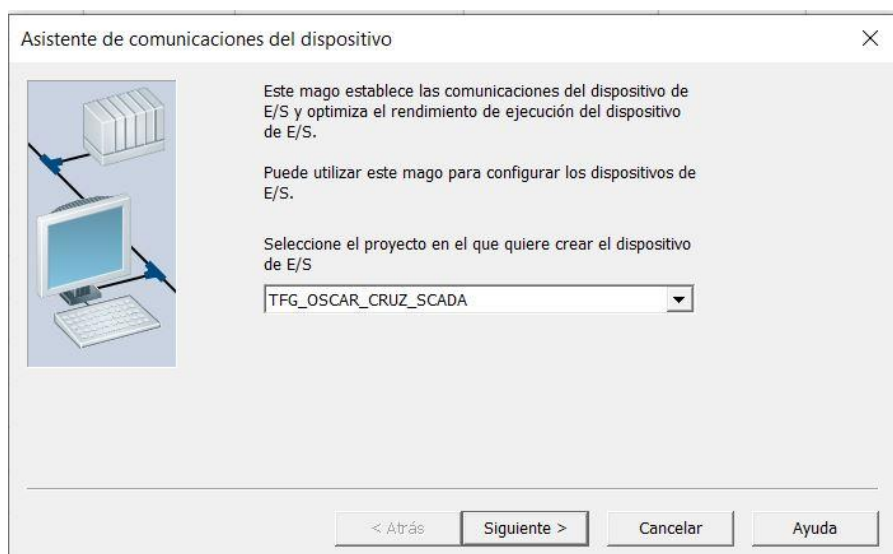


Figura 7.32 Configuración dispositivo E/S

Usaremos un servidor E/S ya existente:

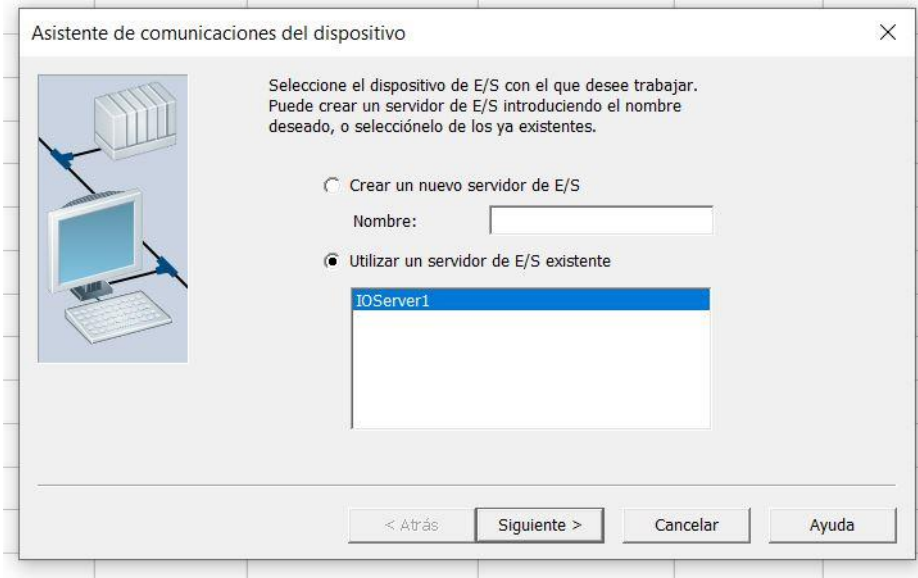


Figura 7.33 Configuración dispositivo E/S II

Creamos un nuevo dispositivo E/S:

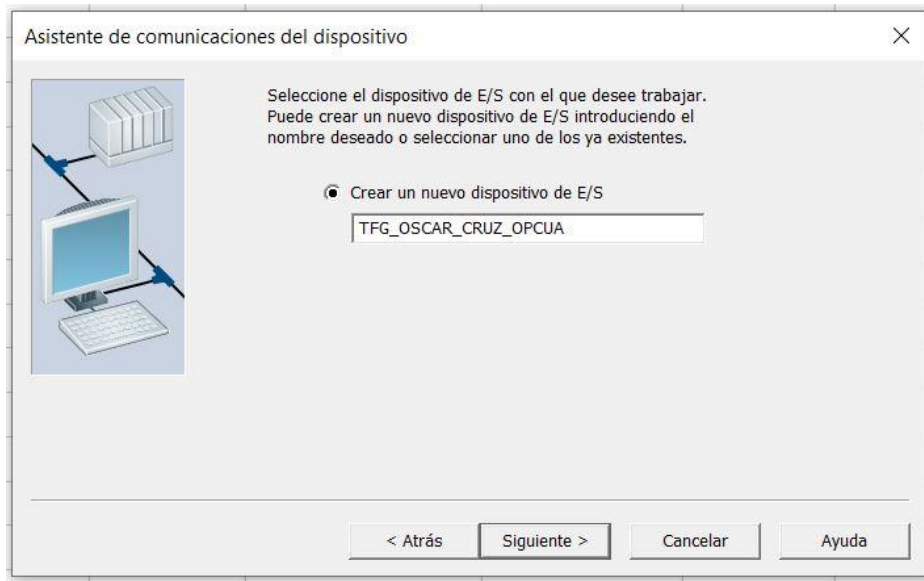


Figura 7.34 Configuración dispositivo E/S III

Que será de tipo externo:

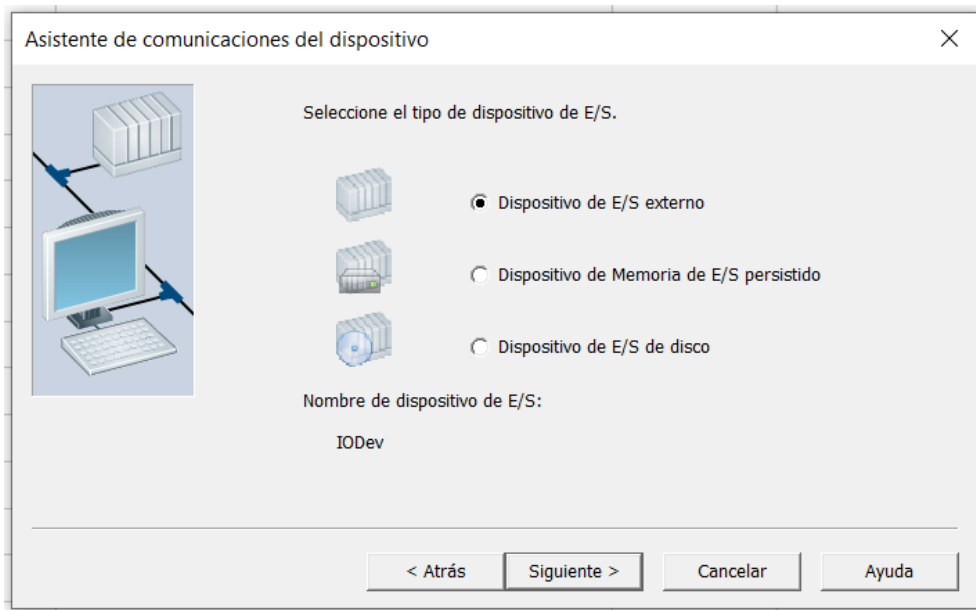


Figura 7.35 Configuración dispositivo E/S IV

El método de comunicación será OPC Foundation -> OPC UA Client. En este paso, si no tenemos la versión R2, nos aparecerá exclusivamente OPC DA Client:

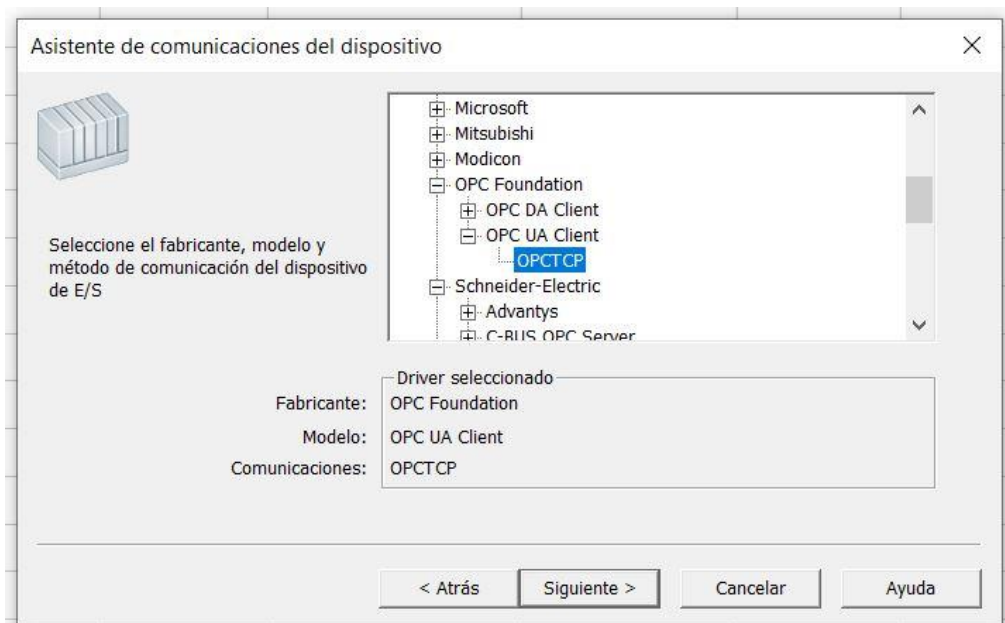


Figura 7.36 Configuración dispositivo E/S V

En la siguiente pantalla pegaremos la dirección del servidor OPC UA, para que nuestro SCADA pueda conectarse a él.

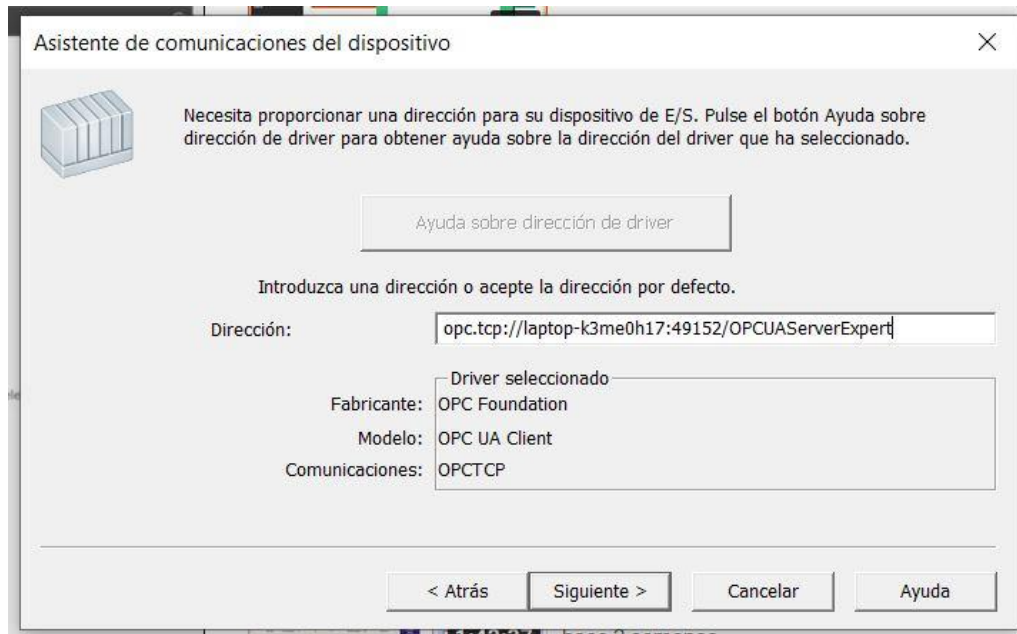


Figura 7.37 Configuración dispositivo E/S VI

En la siguiente ventana configuraremos la base de datos externa. Para ello, activaremos la opción ‘Vincular dispositivos E/S con base de datos de etiquetas ext.’ e indicaremos que el tipo de base de datos será una proveniente de un servidor OPC UA.

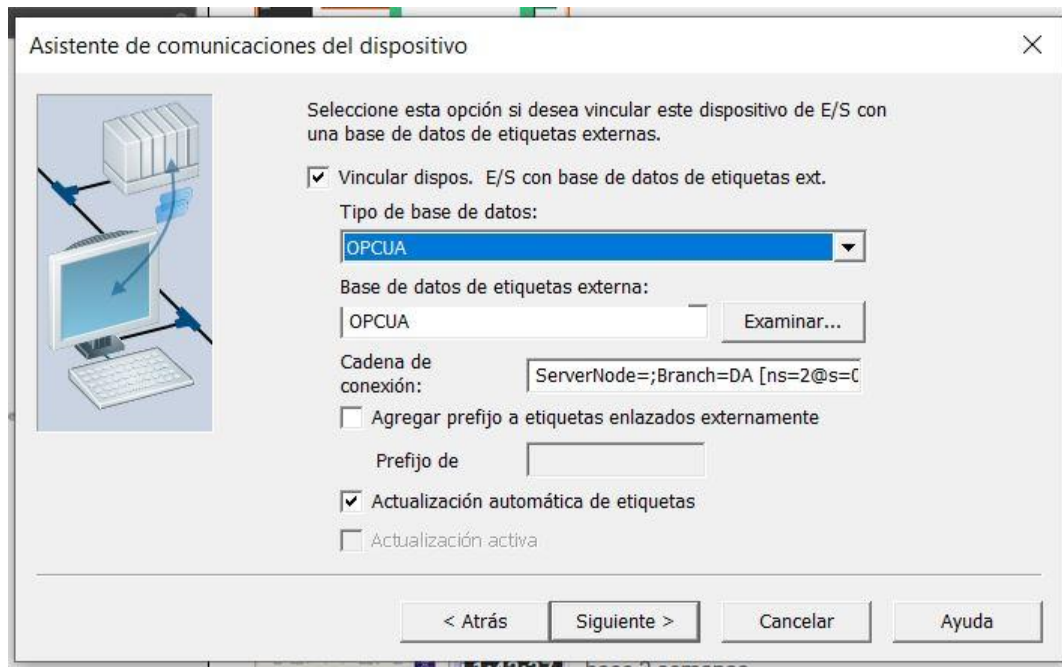


Figura 7.38 Configuración dispositivo E/S VII

En el apartado 'Base de datos de etiquetas externas', le daremos a examinar. Nos saldrá el enlace a nuestro servidor y buscaremos la opción DA.

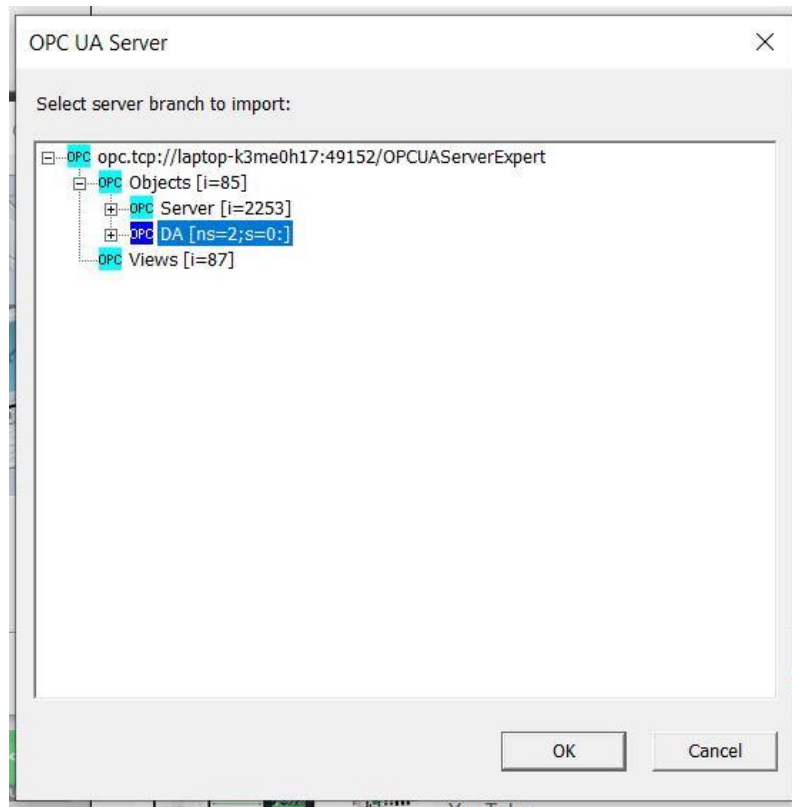


Figura 7.39 Configuración dispositivo E/S VIII

Finalmente, en la última ventana, se nos enseñará un resumen y finalizaremos la configuración.

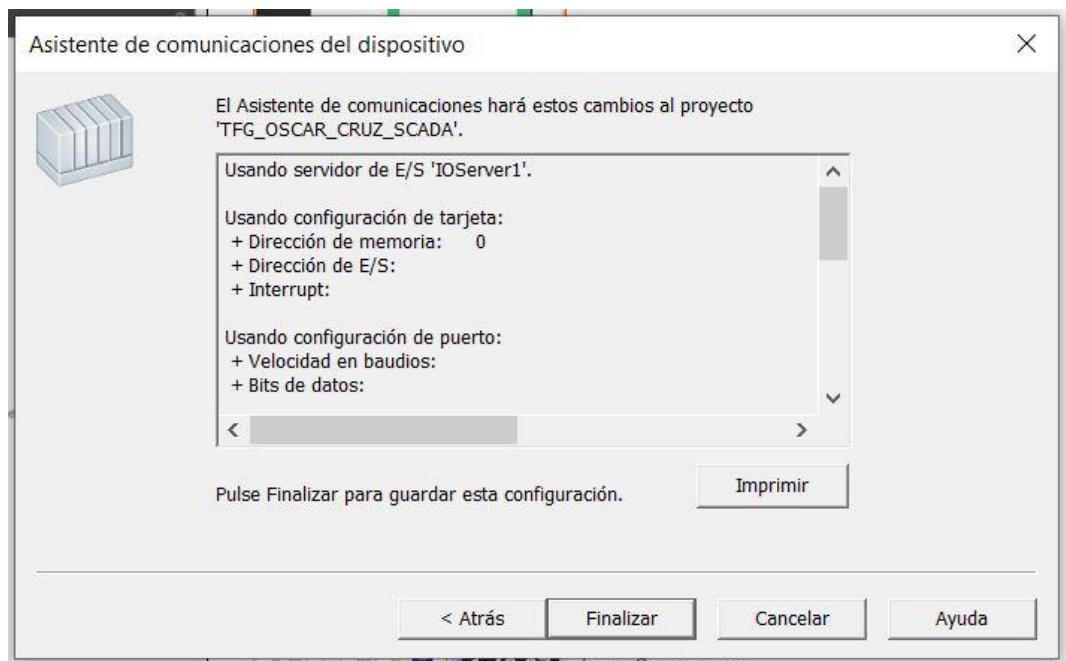
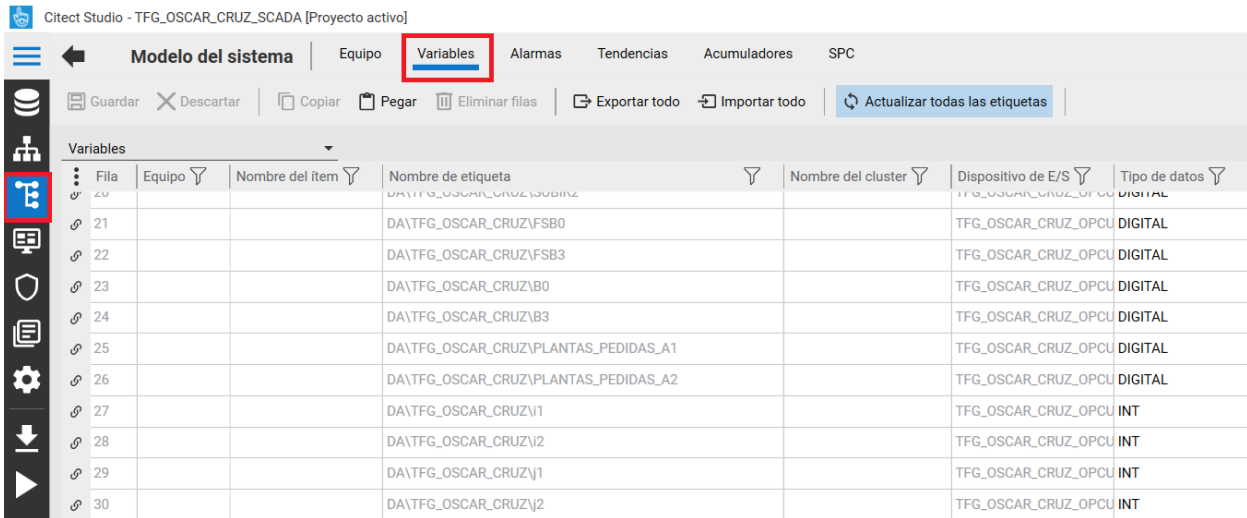


Figura 7.40 Configuración dispositivo E/S IIX

Para comprobar todo lo anterior, nos iremos a la pestaña ‘Modelo del sistema’. En la sección ‘Variables’ podremos encontrarnos con las variables de nuestro proyecto.



Fila	Equipo	Nombre del ítem	Nombre de etiqueta	Nombre del cluster	Dispositivo de E/S	Tipo de datos
20			DA\TFG_OSCAR_CRUZ\FSB0		TFG_OSCAR_CRUZ_OPCL	DIGITAL
21			DA\TFG_OSCAR_CRUZ\FSB0		TFG_OSCAR_CRUZ_OPCL	DIGITAL
22			DA\TFG_OSCAR_CRUZ\FSB3		TFG_OSCAR_CRUZ_OPCL	DIGITAL
23			DA\TFG_OSCAR_CRUZ\B0		TFG_OSCAR_CRUZ_OPCL	DIGITAL
24			DA\TFG_OSCAR_CRUZ\B3		TFG_OSCAR_CRUZ_OPCL	DIGITAL
25			DA\TFG_OSCAR_CRUZ\PLANTAS_PEDIDAS_A1		TFG_OSCAR_CRUZ_OPCL	DIGITAL
26			DA\TFG_OSCAR_CRUZ\PLANTAS_PEDIDAS_A2		TFG_OSCAR_CRUZ_OPCL	DIGITAL
27			DA\TFG_OSCAR_CRUZ\I1		TFG_OSCAR_CRUZ_OPCL	INT
28			DA\TFG_OSCAR_CRUZ\I2		TFG_OSCAR_CRUZ_OPCL	INT
29			DA\TFG_OSCAR_CRUZ\J1		TFG_OSCAR_CRUZ_OPCL	INT
30			DA\TFG_OSCAR_CRUZ\J2		TFG_OSCAR_CRUZ_OPCL	INT

Figura 7.41 Variables importadas del proyecto

7.4.3 Configuración SCADA – OPC UA

Para configurar el OPC UA como fuente de información para ‘Citect Studio’, se tendrá que acceder a su configurador, que puede encontrarse en el buscador de Windows.

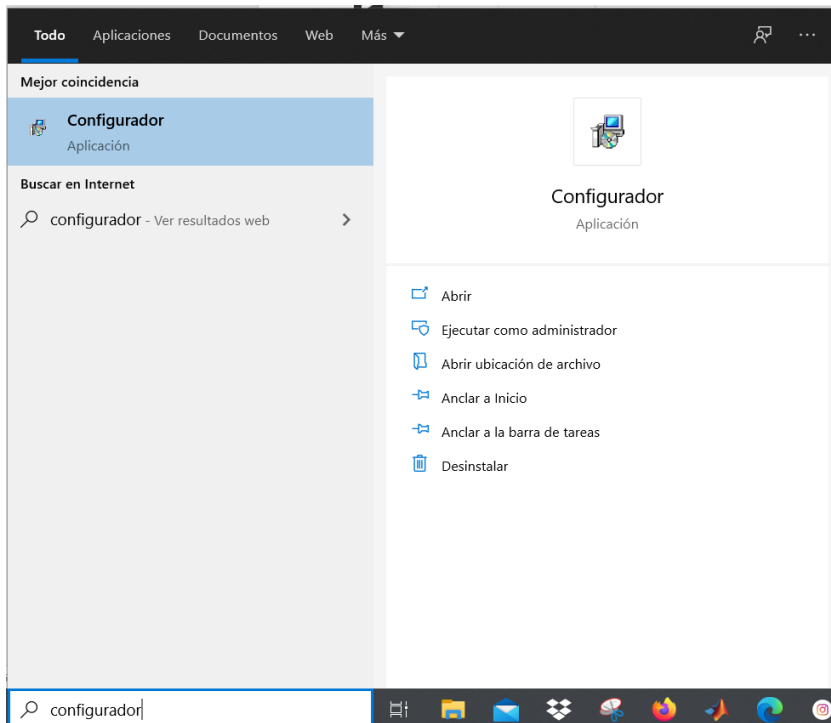


Figura 7.42 Configurador OPC UA

Cuando se ejecute, aparecerá una pantalla como la siguiente. Ahí, se podrá configurar una nueva configuración si es la primera vez o una ya existente, si es que, o bien ya aparece, o se ha configurado con anterioridad.

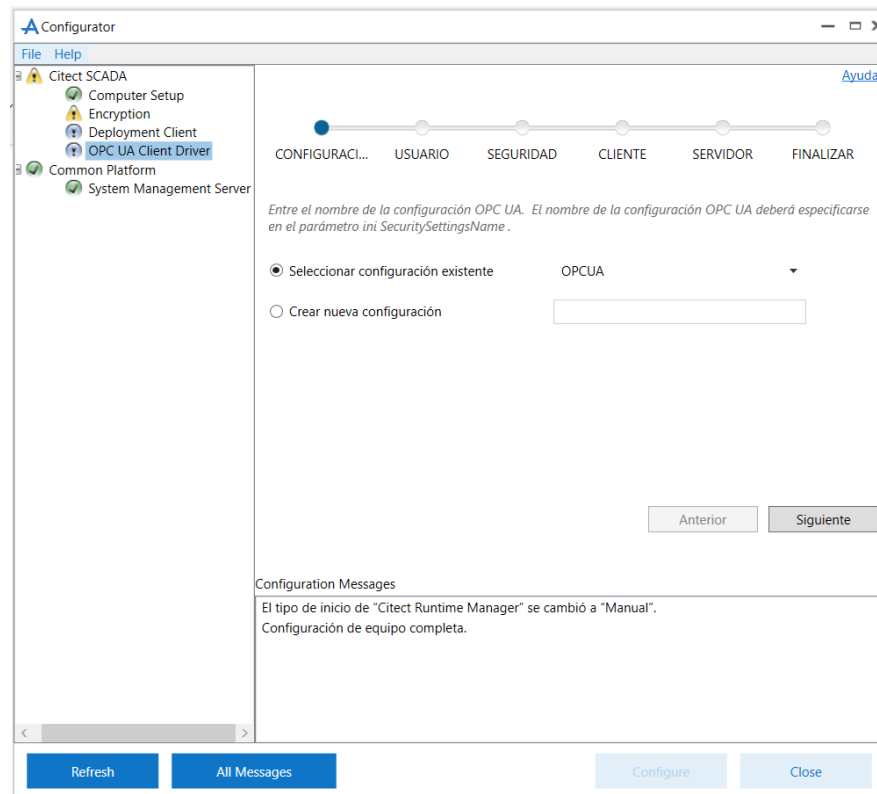


Figura 7.43 Configurator OPC UA II

Para este proyecto no se ha configurado ningún usuario.

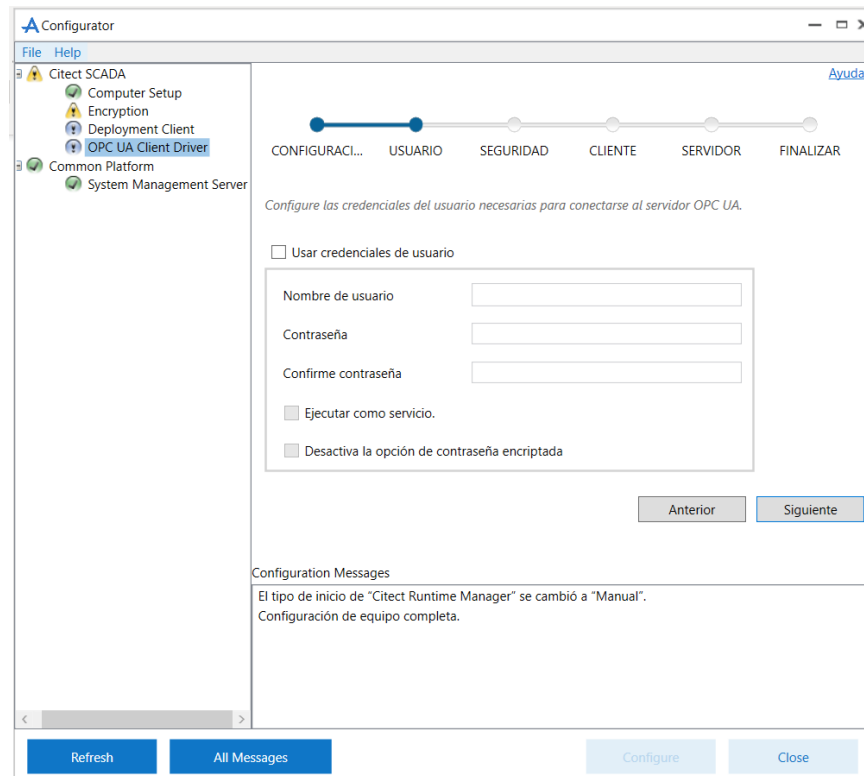


Figura 7.44 Configurator OPC UA III

Ni ningún tipo de seguridad.

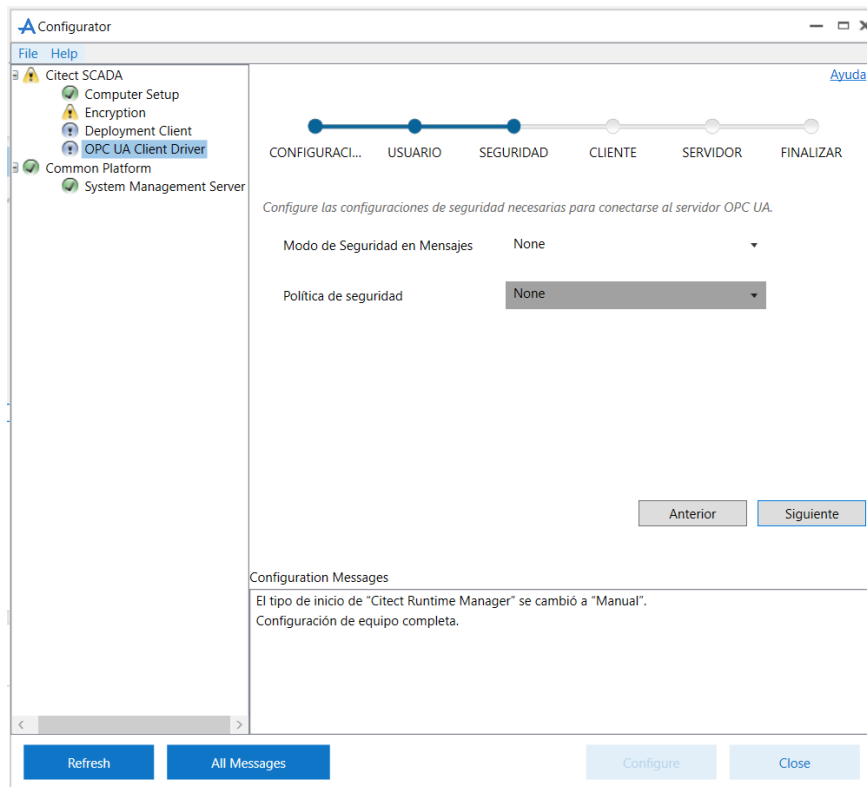


Figura 7.45 Configurador OPC UA IV

Finalmente, se podrá probar la conexión al servidor OPC UA. Para ello, es muy importante darle antes a ‘Configure’. Una vez se le dé, el configurador permitirá introducir la dirección URL del servidor y enviará un mensaje de éxito, si es que está todo bien.

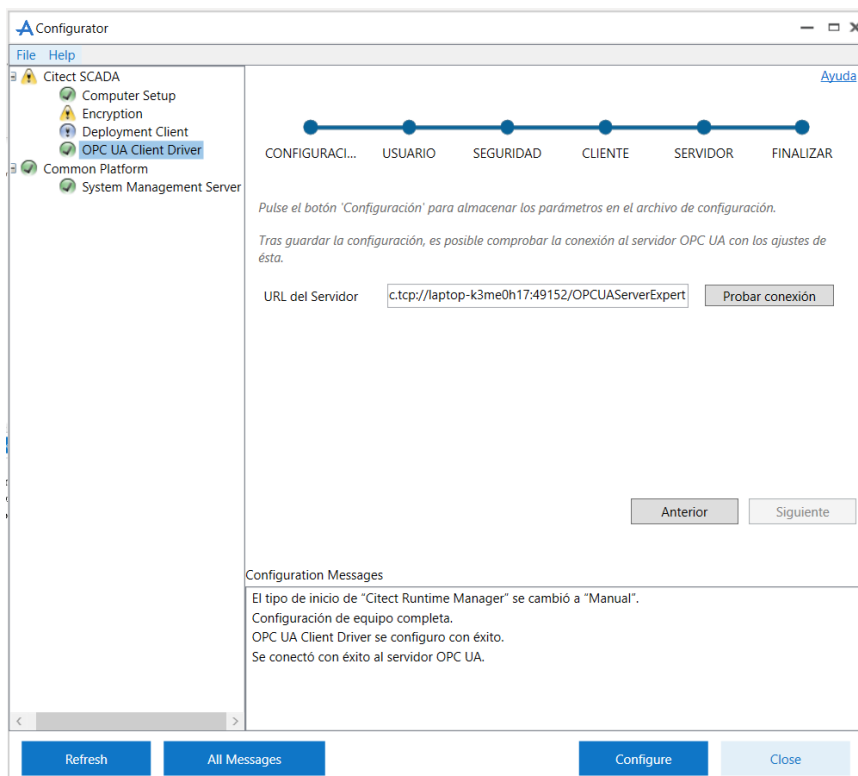


Figura 7.46 Configurador OPC UA V

7.4.4 Asistente de configuración

En la ventana 'Proyectos' se encuentra el asistente de configuración, usado para configurar el proyecto que en ese momento se encuentre activo.

Se usará la configuración expés:

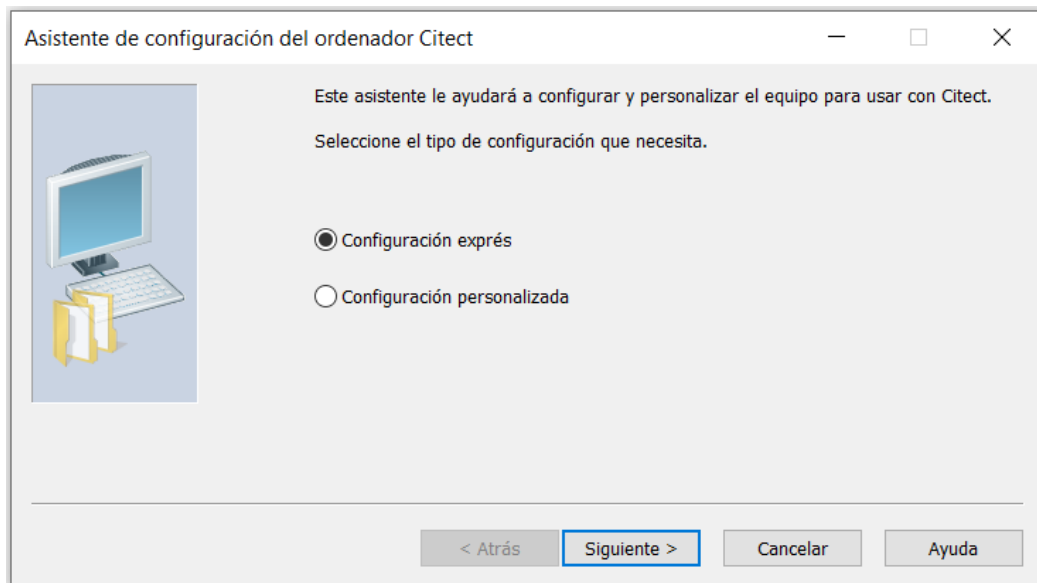


Figura 7.47 Asistente de configuración Citect

Con un entorno de ingeniería local, que quiere decir que toda la información de la planta y el PLC están recogidos en el mismo ordenador, y no en varios.

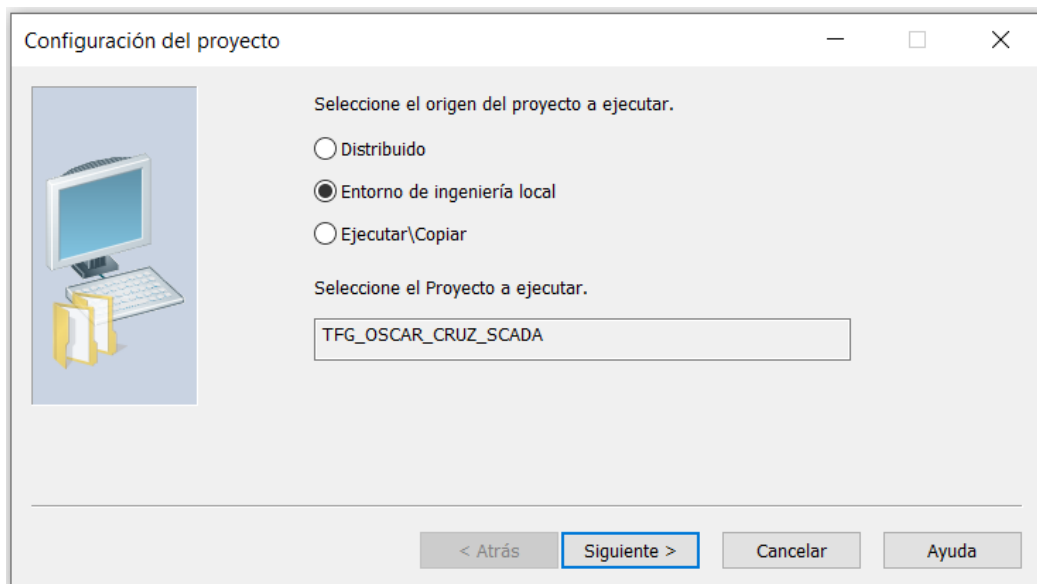


Figura 7.48 Asistente de configuración Citect II

Como ya se ha dicho, el proyecto se selecciona automáticamente, siendo este el que se encuentre activo en el momento de la configuración.

La siguiente pantalla:

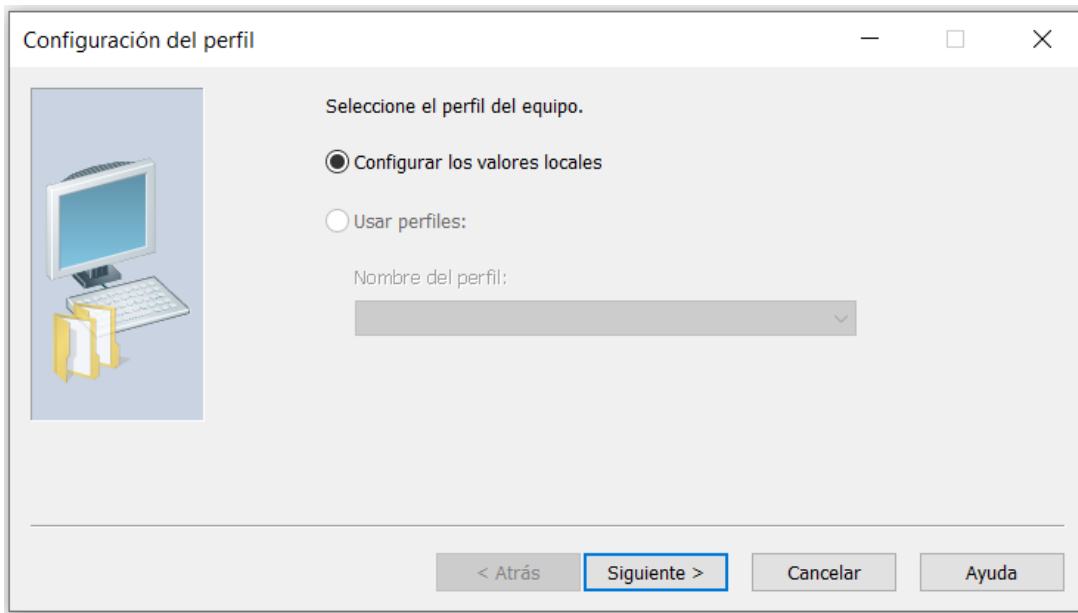


Figura 7.49 Asistente de configuración Citect III

Se tendrá que elegir la página de inicio, teniendo que escoger entre HD1080 o 4K.

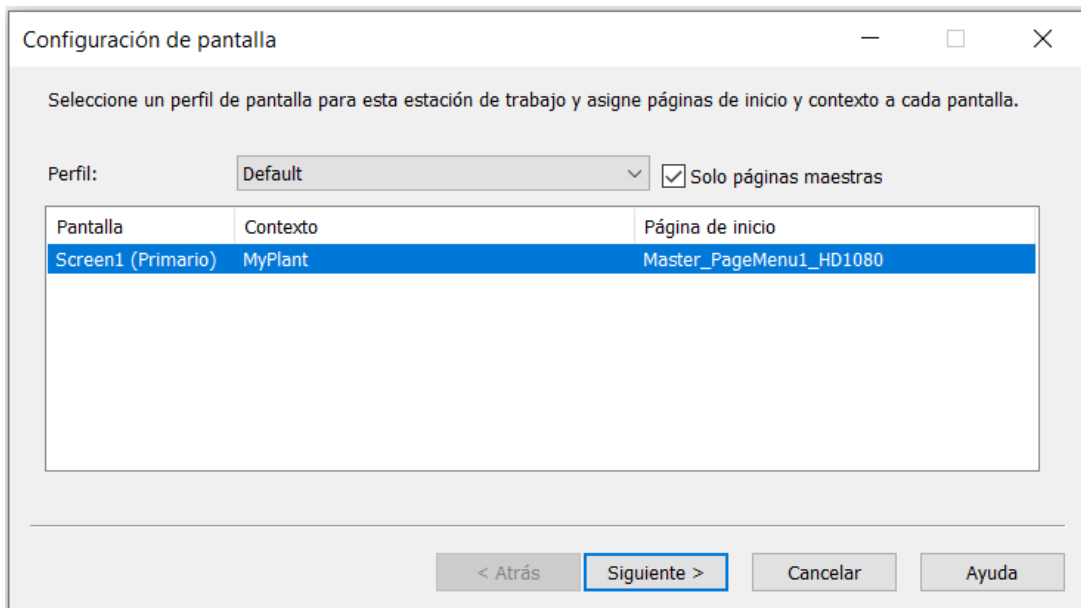


Figura 7.50 Asistente de configuración Citect IV

También se deberá especificar el papel del ordenador que se está configurando dentro del sistema. En este caso, como ya se ha comentado, toda la información estará en el mismo ordenador. De hecho, tanto el servidor (OPC UA) como el cliente (Citect Studio), se encuentran alojados en él.

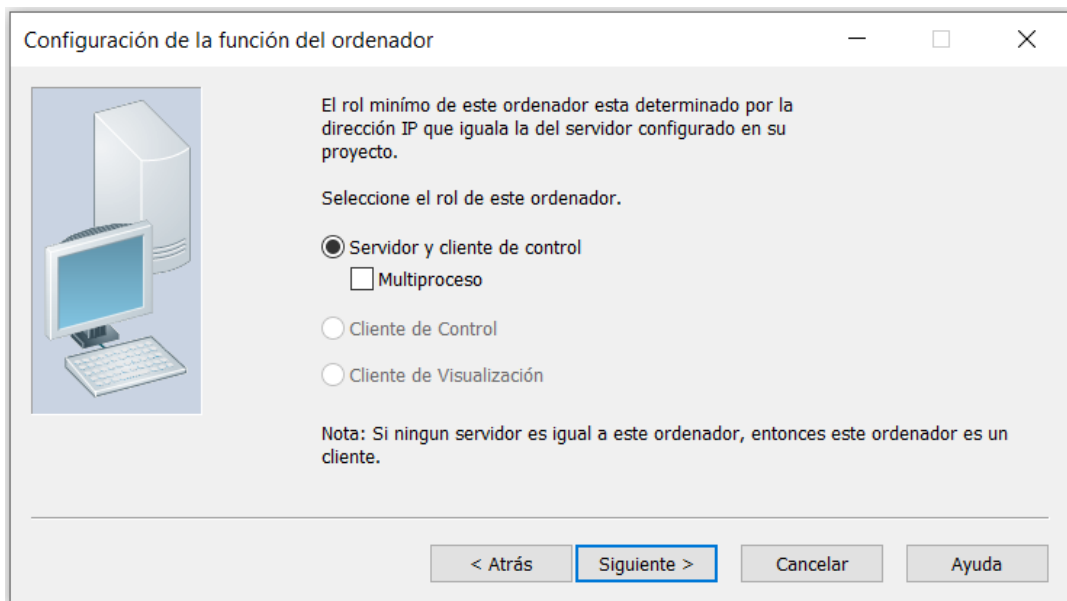


Figura 7.51 Asistente de configuración Citect V

Y, por tanto, este ordenador no se conectará a ningún otro para crear una red distribuida. Cabe destacar que esto es muy poco común en proyectos de ámbito empresarial, ya que precisamente el objetivo del SCADA es crear una red de información rápida y precisa.

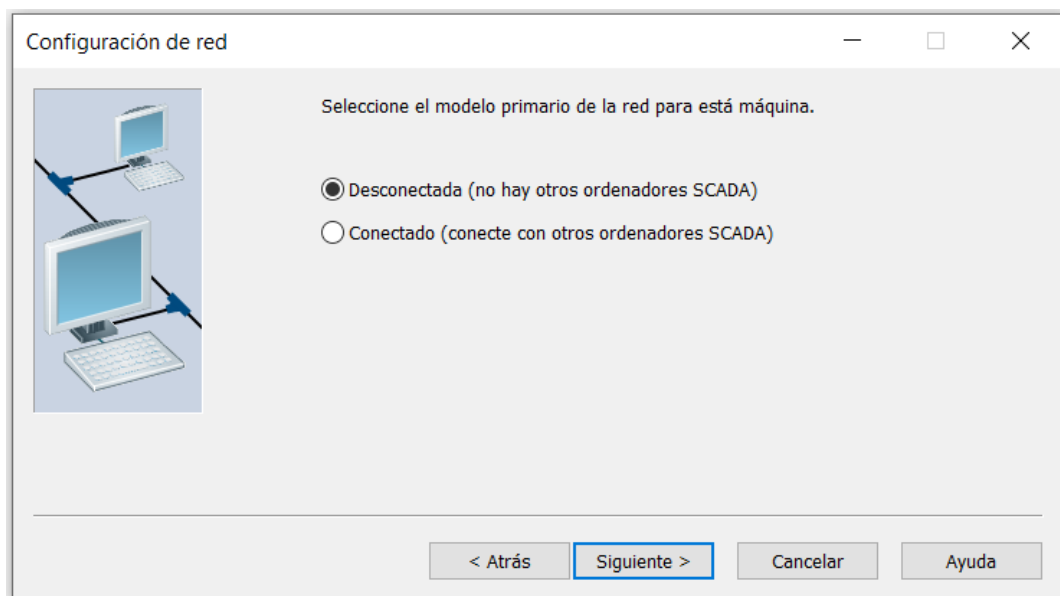


Figura 7.52 Asistente de configuración Citect VI

En la siguiente ventana se pedirá la contraseña del administrador. En este caso, al no haberse configurado ninguna contraseña, se le puede dar a siguiente sin hacer ningún cambio.

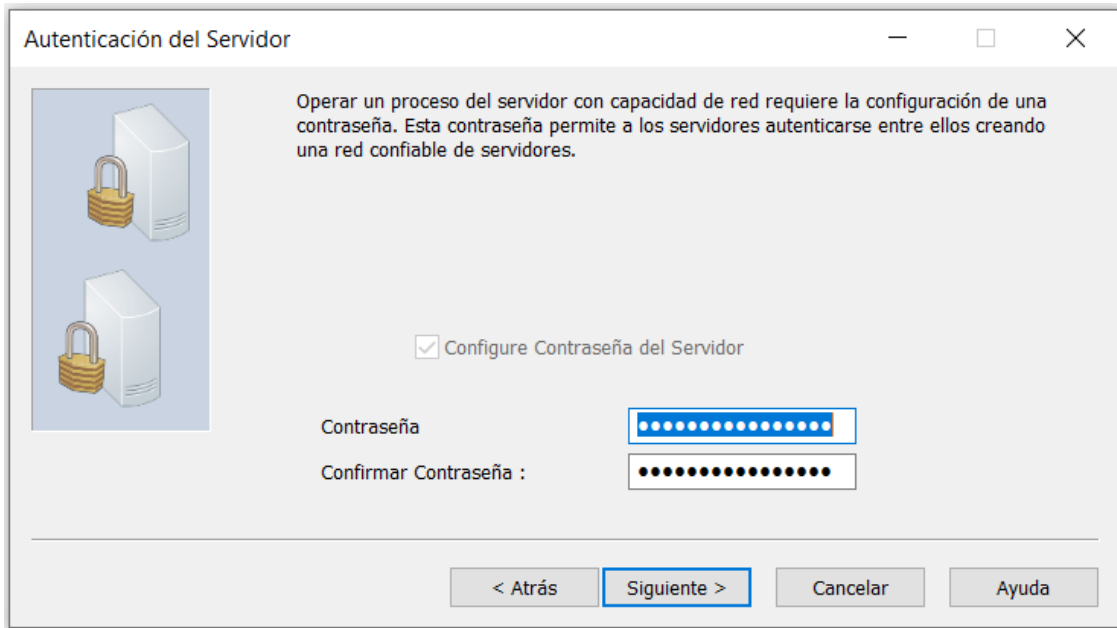


Figura 7.53 Asistente de configuración Citect VII

Y así se llega al final del proceso de configuración:

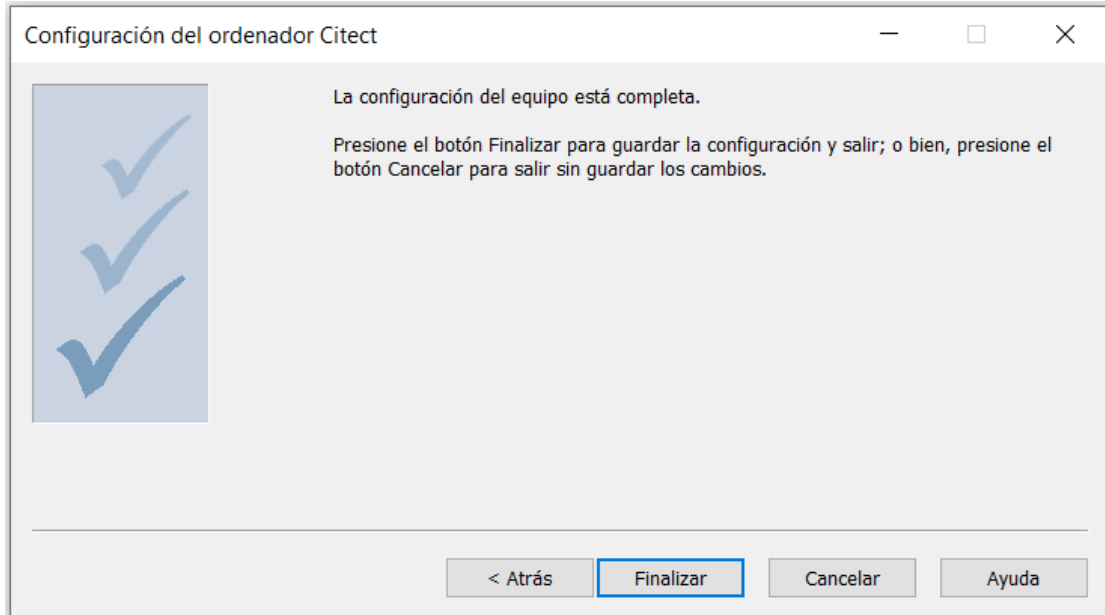


Figura 7.54 Asistente de configuración Citect VIII

8 DISEÑO VISUALIZACIÓN SCADA

La ventana ‘Constructor de gráficos Citect’ ofrece la posibilidad, de una forma intuitiva, de crear una visualización que permitirá al usuario ver el estado de su sistema a tiempo real. Esto se realizará mediante el dibujo de distintas formas, algunas estáticas y otras animadas, de forma que, visualmente, aporte información. Además de esto, hay un sinnúmero de posibilidades, como la de mostrar valores de variables por pantalla, indicadores y crear reports y logs.

8.1 Funciones del SCADA

En este Proyecto se diseñará un SCADA con las siguientes funcionalidades:

1. Visualización en tiempo real de la planta.
2. Gestión de modos.
3. Modo manual.
4. En modo automático, será posible ver aquellas plantas ya pedidas a las que irá el ascensor.
5. Modo semiautomático, incluido dentro del modo automático, en el que se podrá, manualmente, asignar a cada ascensor plantas a las que ir.
6. Creación de logs en formato .txt, que harán un seguimiento de aquellas acciones que se han ejercido sobre la planta desde el SCADA.
7. Creación de informes en formato .rft, que escribirá la planta en la que se encuentra cada ascensor cada 10 segundos.

A pesar de tener tantas funcionalidades, en cada momento, dependiendo del modo en el que se encuentre el autómatas, solo se mostrará la información necesaria, haciendo aparecer y desaparecer cada elemento de forma conveniente.

Por ejemplo, en modo automático:



Figura 8.1 SCADA Modo automático

En modo manual:



Figura 8.2 SCADA Modo manual

En modo reinicio:



Figura 8.3 SCADA Modo Reinicio

Así, se conseguirá una mayor comodidad para el usuario, además de una visión más clara de aquellas funcionalidades que tiene cada modo.

A continuación, se explicará aquellas herramientas que han sido necesarias para el diseño del SCADA.

8.2.1 Elementos estáticos

Los elementos estáticos son aquellos que no reaccionan o modifican ninguna variable. Estos son aquellos que forman la estructura del dibujo y, en este caso, se encuentran representados por:

1. El fondo.
2. La estructura del ascensor.
3. El fondo de los paneles de control.
4. Los títulos de los paneles de control.

Los tres primeros están realizados con el mismo elemento, rectángulos rellenos, mientras que los títulos están hechos con la función texto.

8.2.1.1 Rectángulos

El segundo botón de la primera columna permite dibujar rectángulos. Una vez dibujado, saldrá una ventana de configuración automáticamente. En esa ventana, se podrán configurar muchísimas características, como si este se encuentra relleno o no, el color tanto del interior como del perímetro o incluso crear un fondo con un gradiente de color.

Aunque no se utilizará en este proyecto, también es posible, en la pestaña 'Relleno', utilizar un rectángulo como medidor de nivel o como control deslizante en la pestaña con dicho nombre.

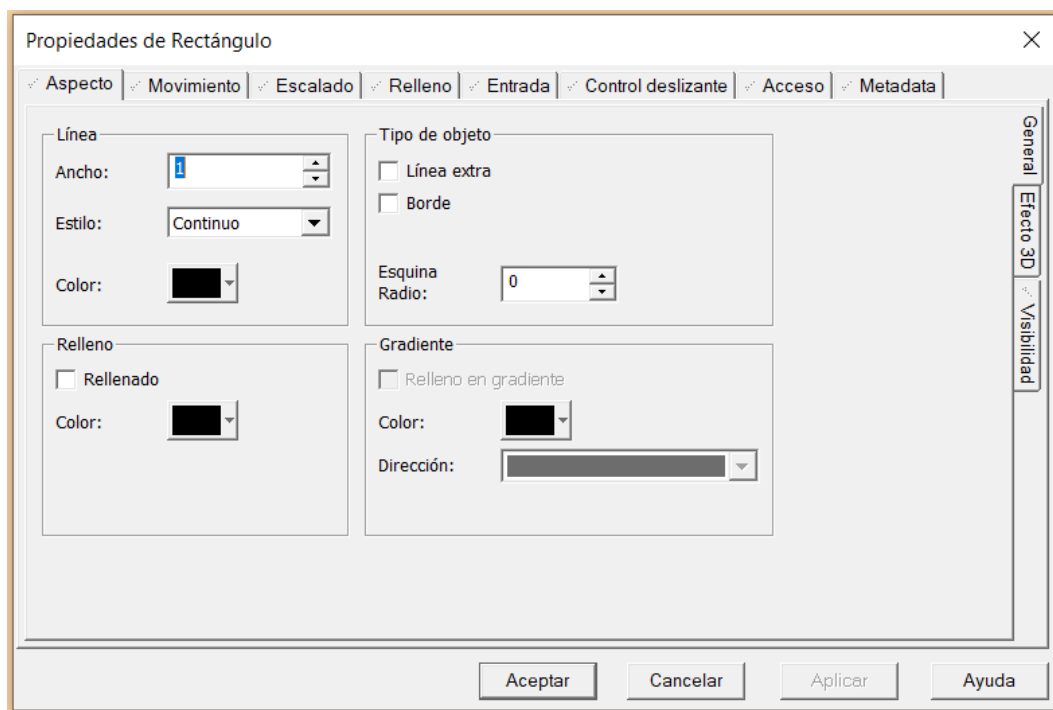


Figura 8.6 Propiedades de Rectángulo

8.2.1.2 Títulos

El cuarto botón de la primera columna tiene la función de crear texto. Una vez pulsado, se escribirá el texto y se pulsará en el lugar en el que se quiera poner. Entonces, aparecerá un cuadro de configuración con las características típicas de cualquier editor.

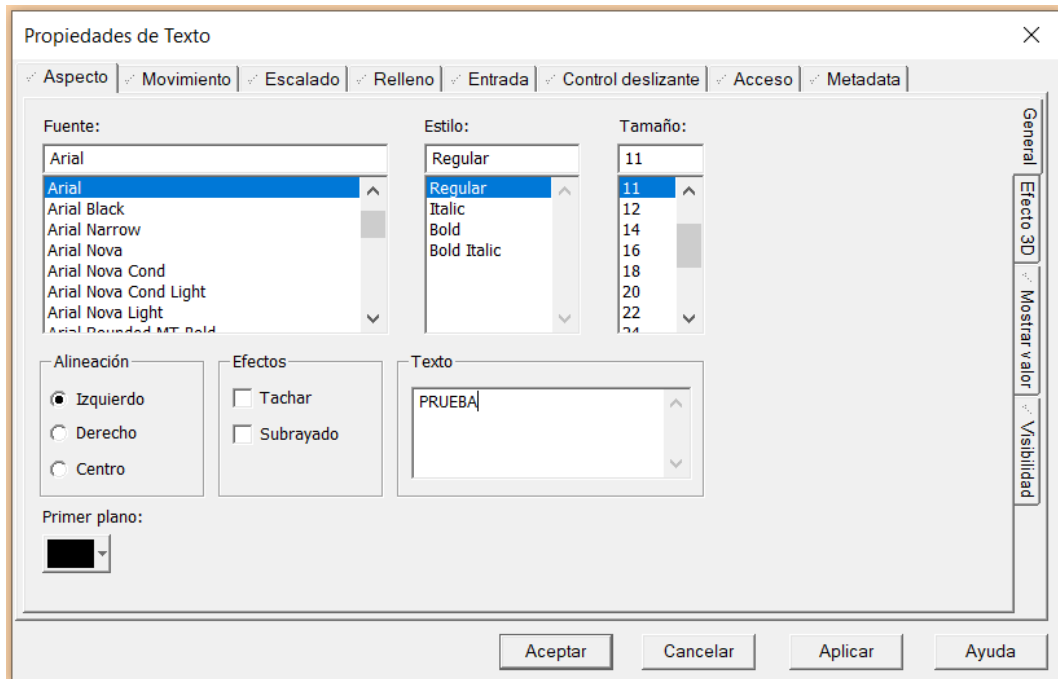


Figura 8.7 Propiedades de Texto

8.2.2 Elementos dinámicos

Son aquellos elementos que modifican o reaccionan ante alguna variable. En este proyecto, algunos de estos elementos son:

1. Los motores.
2. El indicador de subida/bajada/parada justo al lado de cada motor.
3. La cabina del ascensor.
4. El cable de la cabina del ascensor.
5. Leds de llamada a ascensor o leds indicadores de modo.
6. Indicadores numéricos de buffer.
7. Botones.

A continuación, se hará un análisis de cada uno de los elementos.

8.2.2.1 Motores

El quinto botón de la segunda columna se titula ‘Grupo de símbolos’ y ofrece una amplia variedad de representaciones de sistemas. Uno de ellos son los motores.

Una vez se clicla en el botón y posteriormente en la visualización aparece la siguiente pantalla:

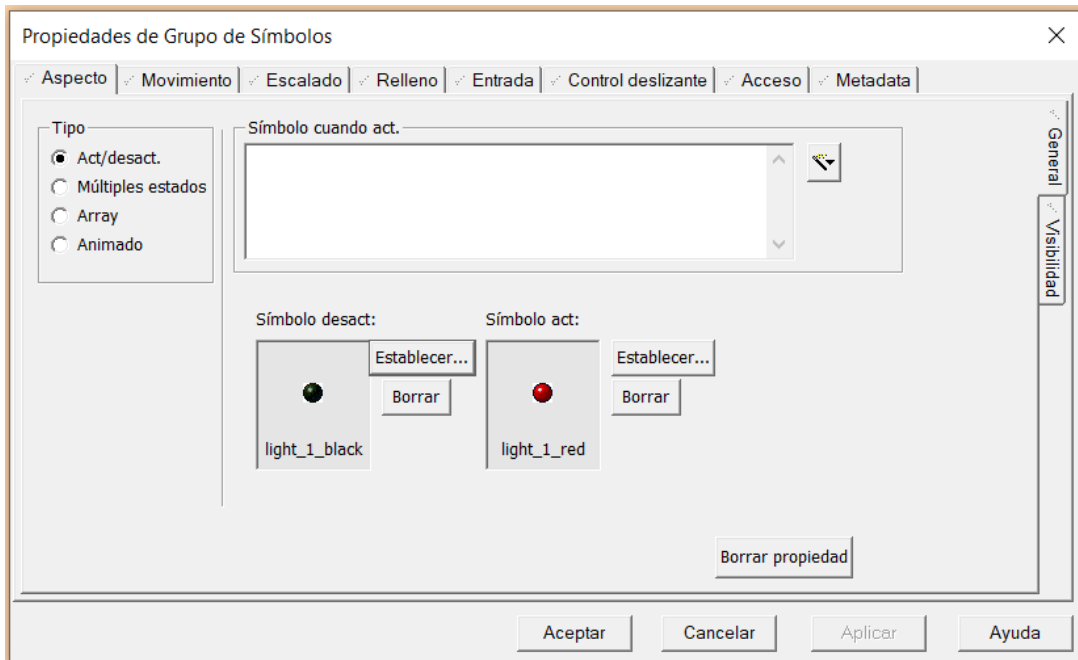


Figura 8.8 Propiedades de Grupo de Símbolos

Como se observa, se puede establecer un símbolo para aquellos momentos en los que una variable esté desactivada y otro para aquellos en los que la variable esté activada. Dicha variable podrá introducirse en el recuadro superior.

En este caso es importante destacar que el nombre de la variable podrá encontrarse en la ventana ‘Modelo del sistema’, pestaña ‘Variables’ y columna ‘Nombre de etiqueta’.

Por defecto, aparecen leds en color negro y rojo para los estados de desactivado y activado, respectivamente. Sin embargo, existen una gran variedad de símbolos que pueden escogerse clicando en el botón ‘Establecer...’.

En el caso del motor usado en este proyecto, se encontrará en la pestaña ‘motor_base’.

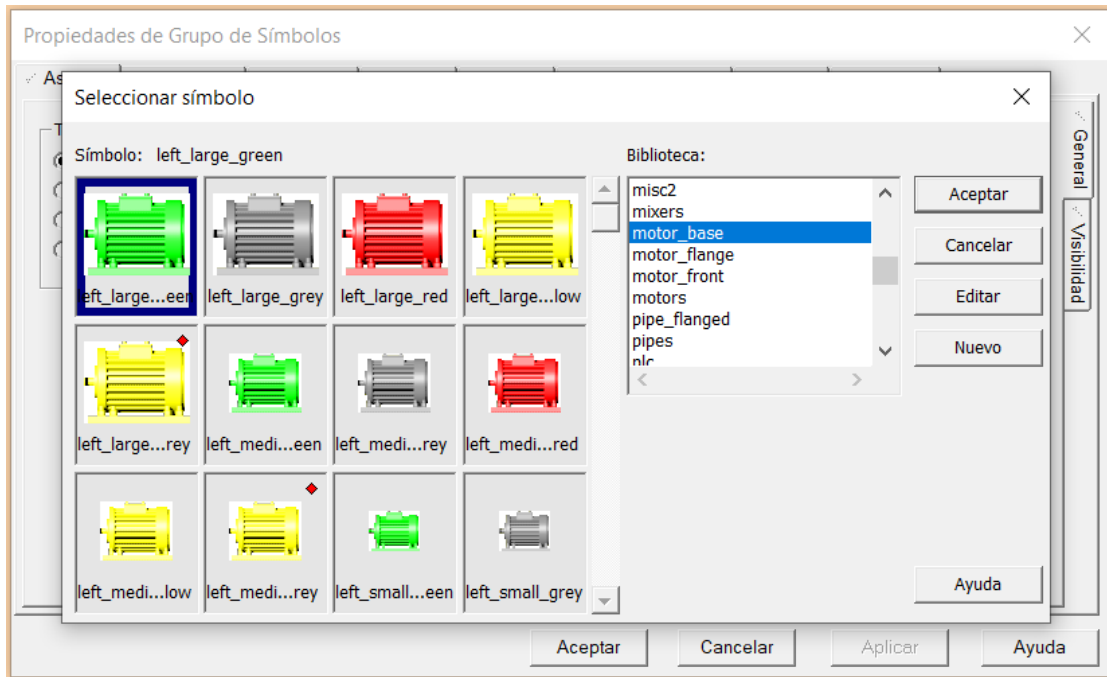


Figura 8.9 Propiedades de Grupo de Símbolos II

También pueden encontrarse otros tipos de motores, tuberías, los propios autómatas y un sinfín de sistemas. El programa también ofrece al usuario la posibilidad de diseñar sus propios sistemas.

8.2.2.2 Indicadores de subida/bajada/parada

Justo al lado de los motores, se encuentran unos indicadores que aportan información sobre el movimiento del ascensor. Estos son flechas verdes de subida/bajada o una línea horizontal gris para indicar la parada.

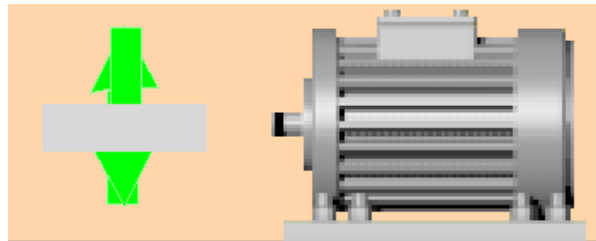


Figura 8.10 Motor con indicador de estado

Al no tener la visualización en modo ejecución, aparecen los tres a la vez. Sin embargo, cada uno de ellos está configurado para aparecer solo y exclusivamente cuando su variable esté activada. Por tanto, en ejecución, al no poder subir/bajar a la vez el ascensor, solo aparecerá uno de ellos.

Un ejemplo configuración de la flecha de subida del ascensor 1, donde se ve claramente el nombre de la variable tal y como se ha explicado en el apartado 9.2.1 podrá verse en la siguiente página.

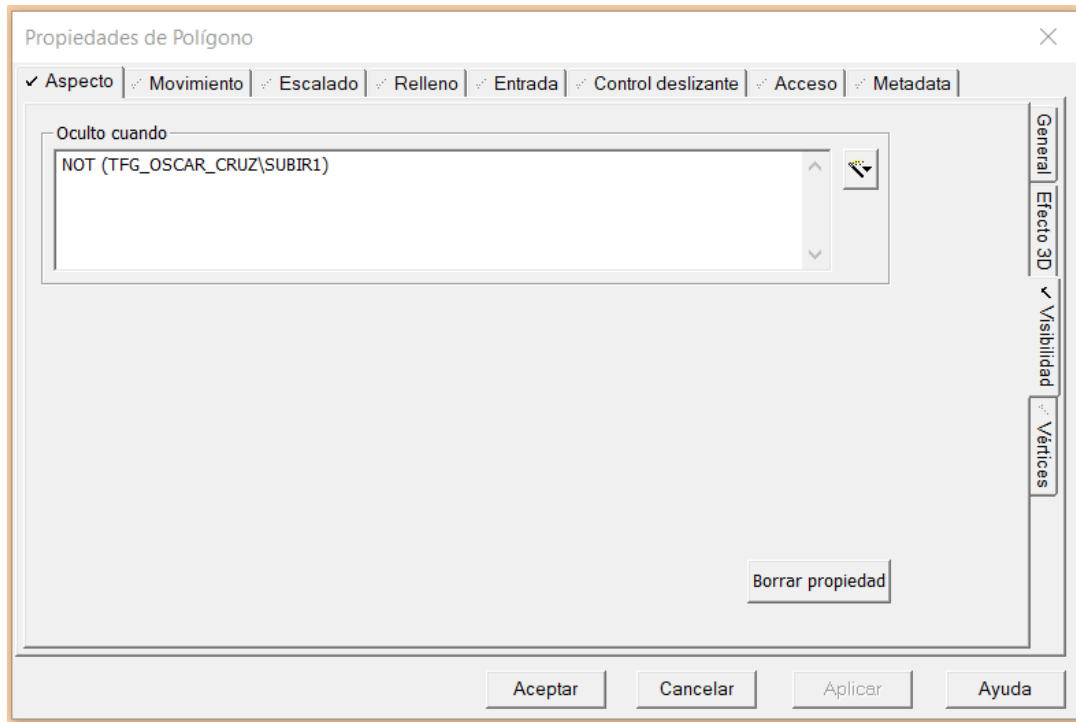


Figura 8.11 Propiedades de Polígono

Como se puede ver, en este caso se configura cuando este elemento estará oculto. En este caso, se dirá que la flecha debe estar oculta cuando el ascensor no esté subiendo (Por tanto, estará visible cuando el ascensor suba).

8.2.2.3 Cabina del ascensor

La cabina del ascensor sigue el mismo patrón que los indicadores de movimiento del ascensor. En este caso también se jugará con los modos de parada/ejecución, ya que, en realidad, en el modo de no ejecución, estará pintada la cabina (rectángulo naranja) en todas y cada una de las plantas. En el modo de ejecución, estos rectángulos se ocultarán o serán visibles en función de las variables PLANTAA1 o PLANTAA2.

Las dos primeras plantas:

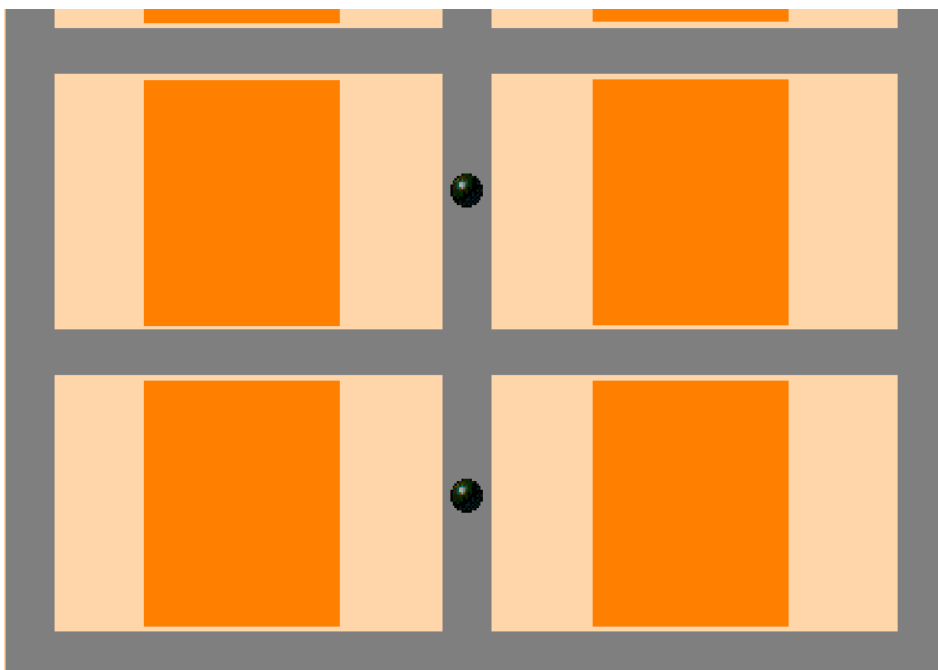


Figura 8.12 Cabinas de ascensores

Un ejemplo del ascensor 1 sería:

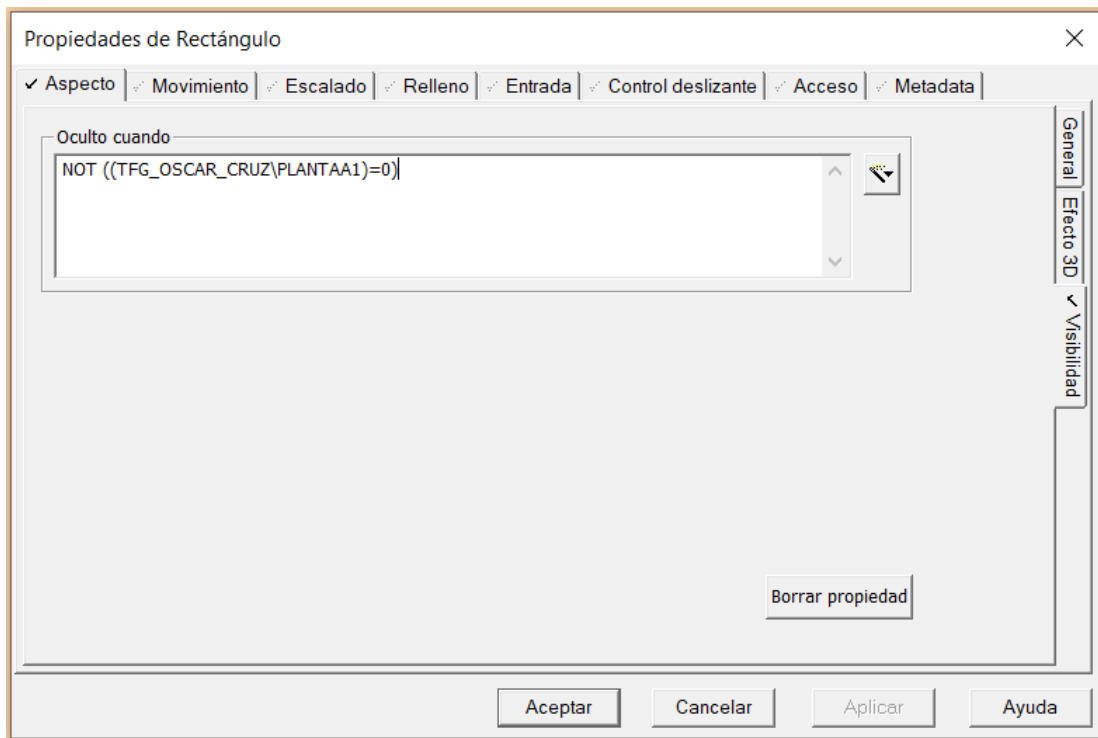


Figura 8.13 Propiedades de Rectángulo – Visibilidad

Este ascensor estará oculto cuando la variable PLANTAA1 no sea cero (estará visible cuando PLANTAA1 sea cero). El resto de las cabinas sigue un patrón similar.

8.2.2.4 Leds de llamada de ascensor o indicadores de modo

Los leds siguen el mismo patrón expuesto en el punto 9.2.2.1, con las variables correspondientes, en este caso las variables de modo o las variables de planta llamada.

8.2.2.5 Indicadores numéricos de buffer

El menú ofrece la posibilidad de insertar valores numéricos en el cuarto botón de la segunda columna. Su funcionamiento es sencillo; solo debe introducirse la variable a mostrar, además del formato de dicho número.

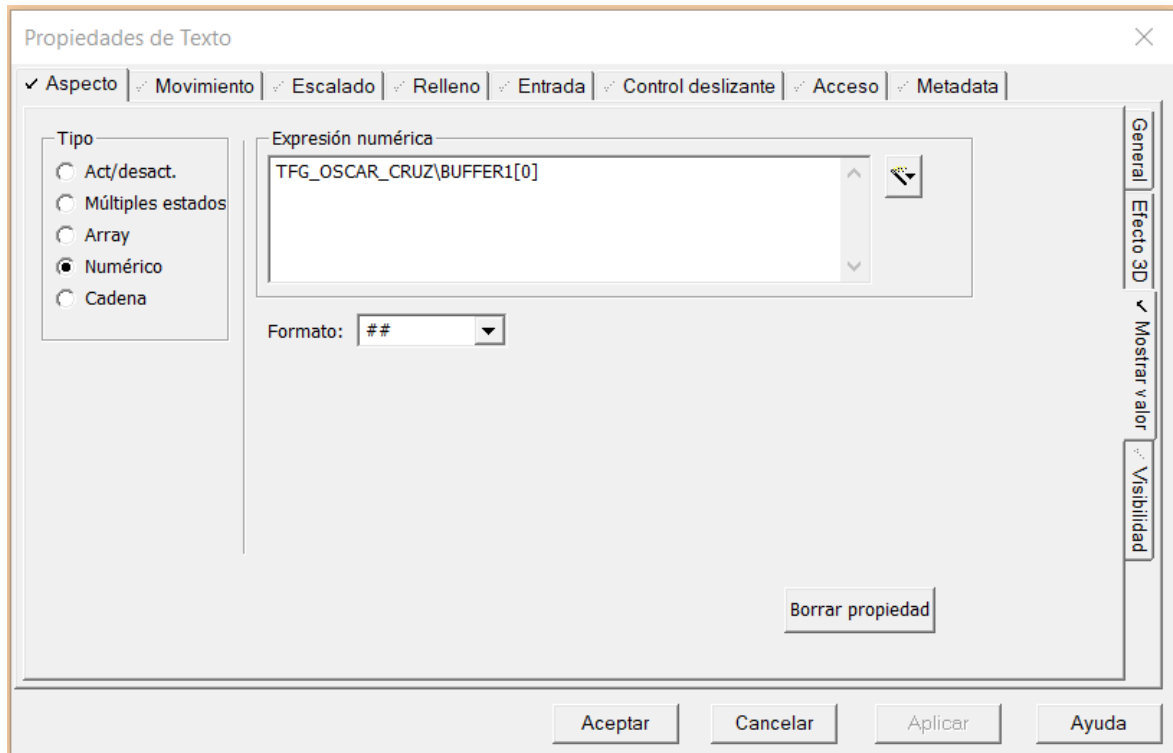


Figura 8.14 Propiedades de Texto - Expresión numérica

8.2.2.6 Botones

En la visualización existen una gran cantidad de botones, ya que, además de visualizar, se pretende hacer una pantalla de operador, que permita al usuario controlar la planta desde el mismo SCADA.

La configuración de botones puede ser algo compleja. El primer paso, el más sencillo, es establecer la apariencia física del mismo.

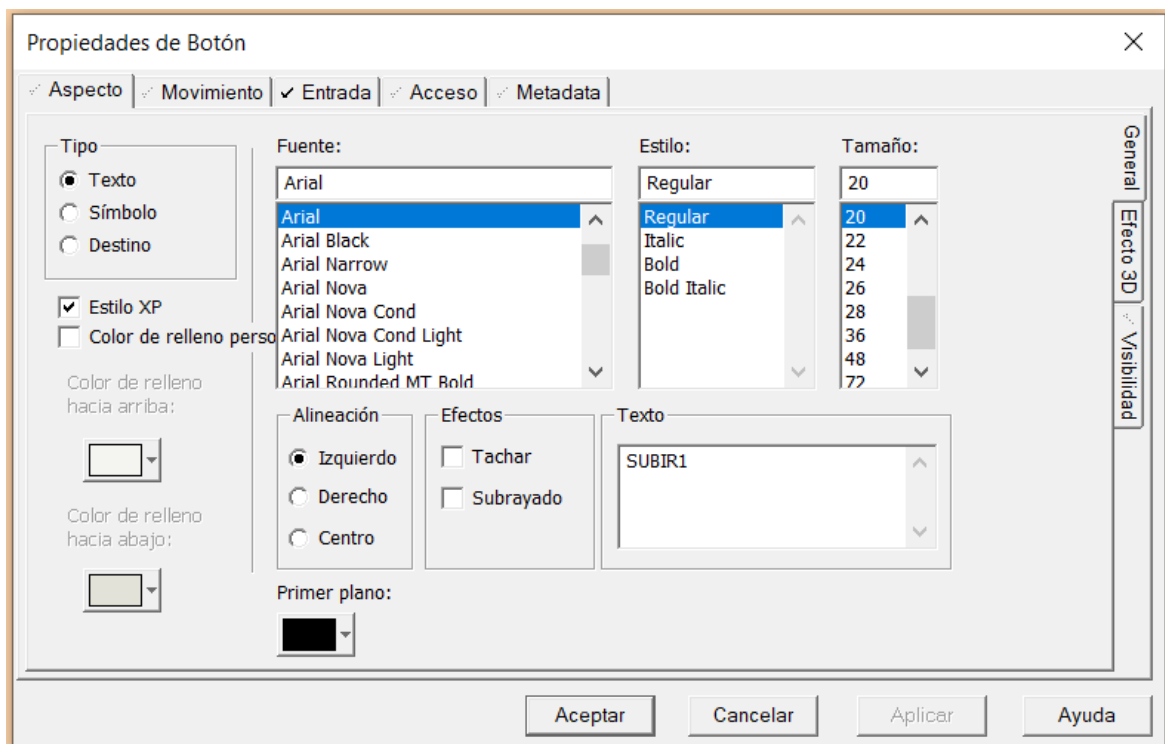


Figura 8.15 Propiedades de Botón

Posteriormente, en el apartado de 'Entrada', se muestran tres tipos de acciones, aunque en este proyecto solo se utilizarán dos:

1. Arriba: se pulsará el botón, pero este no realizará la acción pedida hasta que no deje de ser pulsado.
2. Abajo: el botón activará la variable conveniente durante el tiempo en el que el botón esté pulsado.

Por ello, por ejemplo, para hacer subir manualmente el ascensor 1 se realizarán las siguientes configuraciones: en el apartado 'Abajo' se activará la variable manual de subida, mientras que en apartado 'Arriba' se desactivará. Así, se conseguirá que el ascensor suba mientras esté el botón pulsado y se pare una vez que este deje de pulsarse.

Abajo:

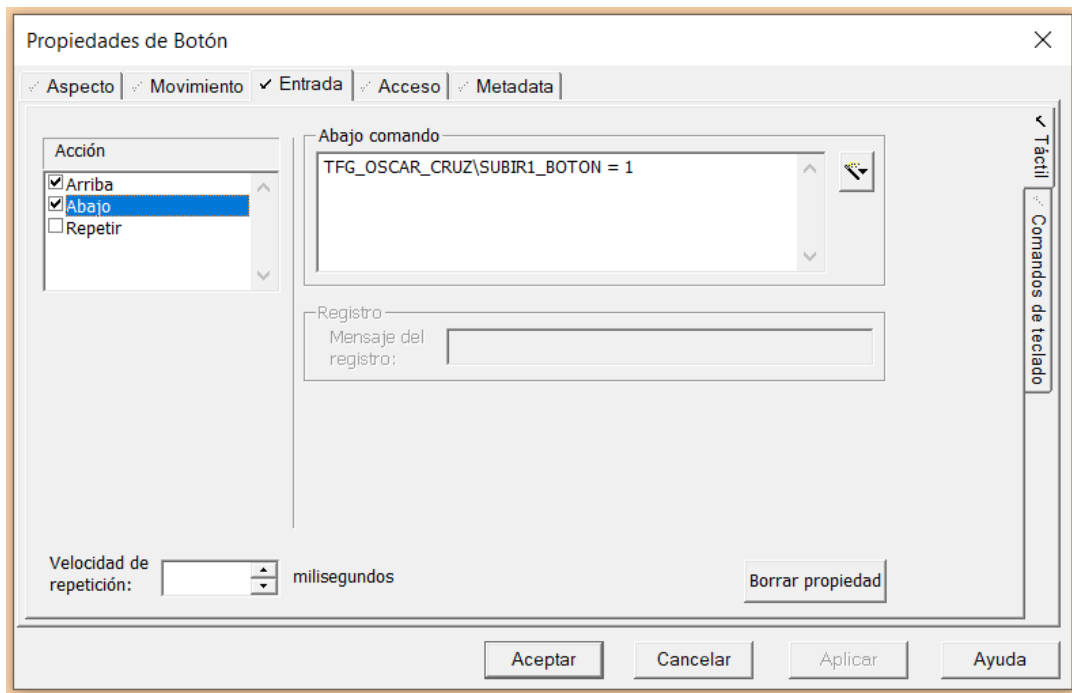


Figura 8.16 Propiedades de Botón - Abajo

Arriba:

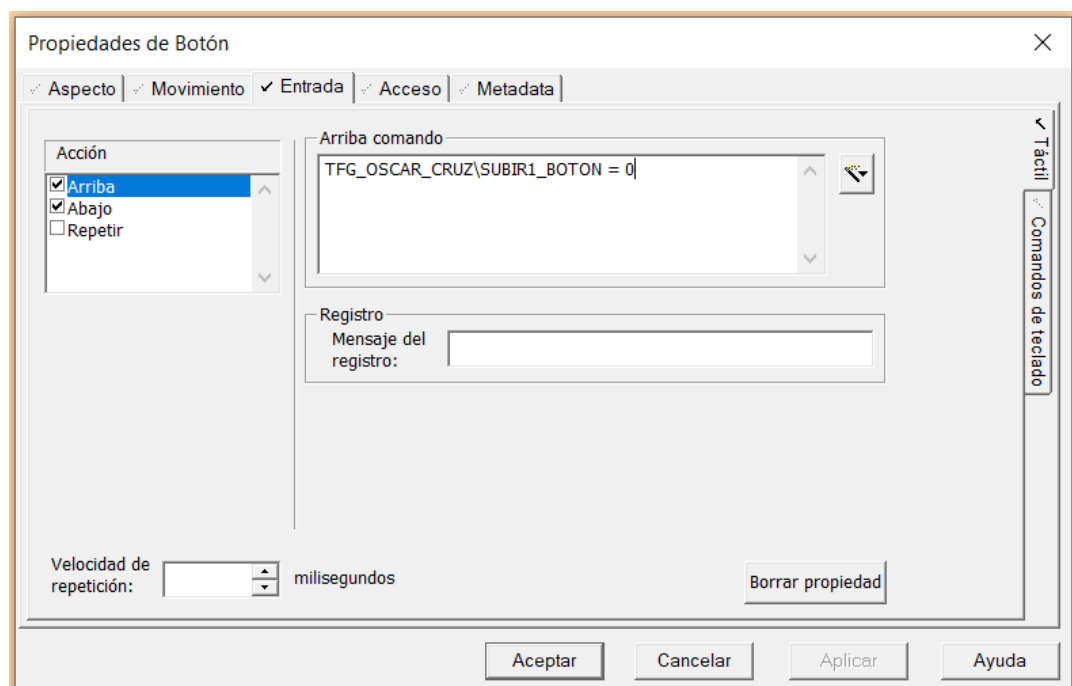


Figura 8.17 Propiedades de Botón - Arriba

8.3 Implementación de 'logs' e informes

Tanto los logs como los informes tienen como objetivo crear un historial, aunque con un punto de vista distinto:

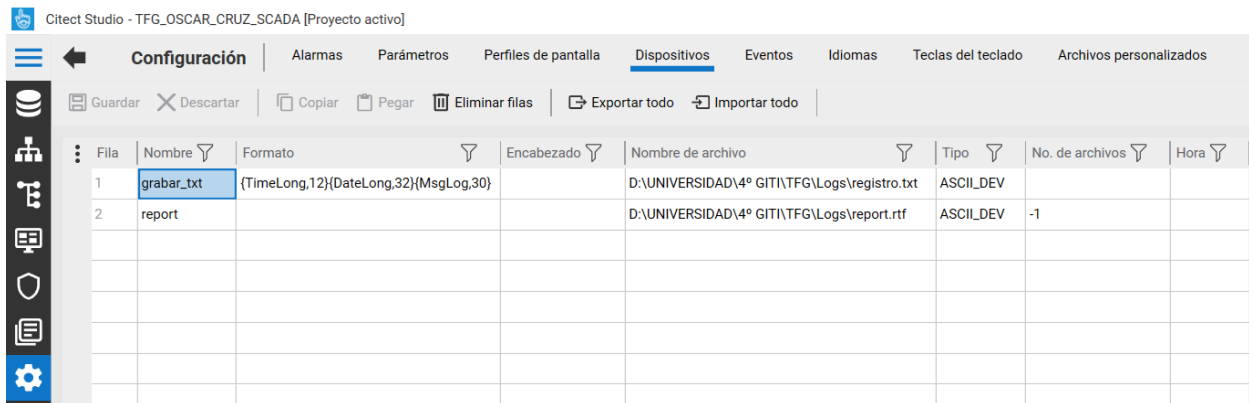
1. Los logs buscan llevar un control de los distintos usuarios que tienen acceso a la planta. Por ello, estará orientado a registrar que botones se han pulsado y que usuario lo ha hecho. En este proyecto, al no haber usuario, pues solo tiene acceso a la planta el administrador, no se mostrará dicha información, sino solo el pulsado de botones, junto a la fecha y la hora.
2. Los informes tienen un alcance mucho mayor, usando lenguaje Cicode, el propio de Citect, el programa podrá, periódicamente, escribir en un fichero lo que el programador haya decidido que se escriba. En este caso, se ha optado por, cada 10 segundos, hacer un registro de fecha, hora y planta en la que se encuentra cada uno de los ascensores.

Naturalmente, se deben hacer unas configuraciones adicionales para crear estos registros. Estas configuraciones se explicarán a continuación.

8.3.1 Logs

8.3.1.1 Creación de dispositivo

Para crear un Log, lo primero que se debe hacer es crear un 'dispositivo', opción se encuentra en la ventana 'Configuración'



Citect Studio - TFG_OSCAR_CRUZ_SCADA [Proyecto activo]

Configuración | Alarmas | Parámetros | Perfiles de pantalla | **Dispositivos** | Eventos | Idiomas | Teclas del teclado | Archivos personalizados

Guardar | Descartar | Copiar | Pegar | Eliminar filas | Exportar todo | Importar todo

Fila	Nombre	Formato	Encabezado	Nombre de archivo	Tipo	No. de archivos	Hora
1	grabar_txt	{TimeLong,12}{DateLong,32}{MsgLog,30}		D:\UNIVERSIDAD\4º GITI\TFG\Logs\registro.txt	ASCII_DEV		
2	report			D:\UNIVERSIDAD\4º GITI\TFG\Logs\report.rtf	ASCII_DEV	-1	

Figura 8.18 Creación de dispositivo

Los campos obligatorios son:

1. Nombre: en pasos siguientes se utilizará para indicar a los botones en que dispositivo deben grabar.
2. Formato: definirá que datos se incluirán en el archivo y cuánto puede ocupar, como máximo, cada apartado. En este caso se ha incluido fecha, hora y el mensaje que enviará el botón al ser pulsado.
3. Nombre de archivo: dirección en la que se quiere que se guarde el archivo.
4. Tipo: ASCII_DEV para formato tipo texto.

8.3.1.2 Configuración de botones

En la pestaña 'acceso' se le indicará a cada botón en que dispositivo debe escribir.

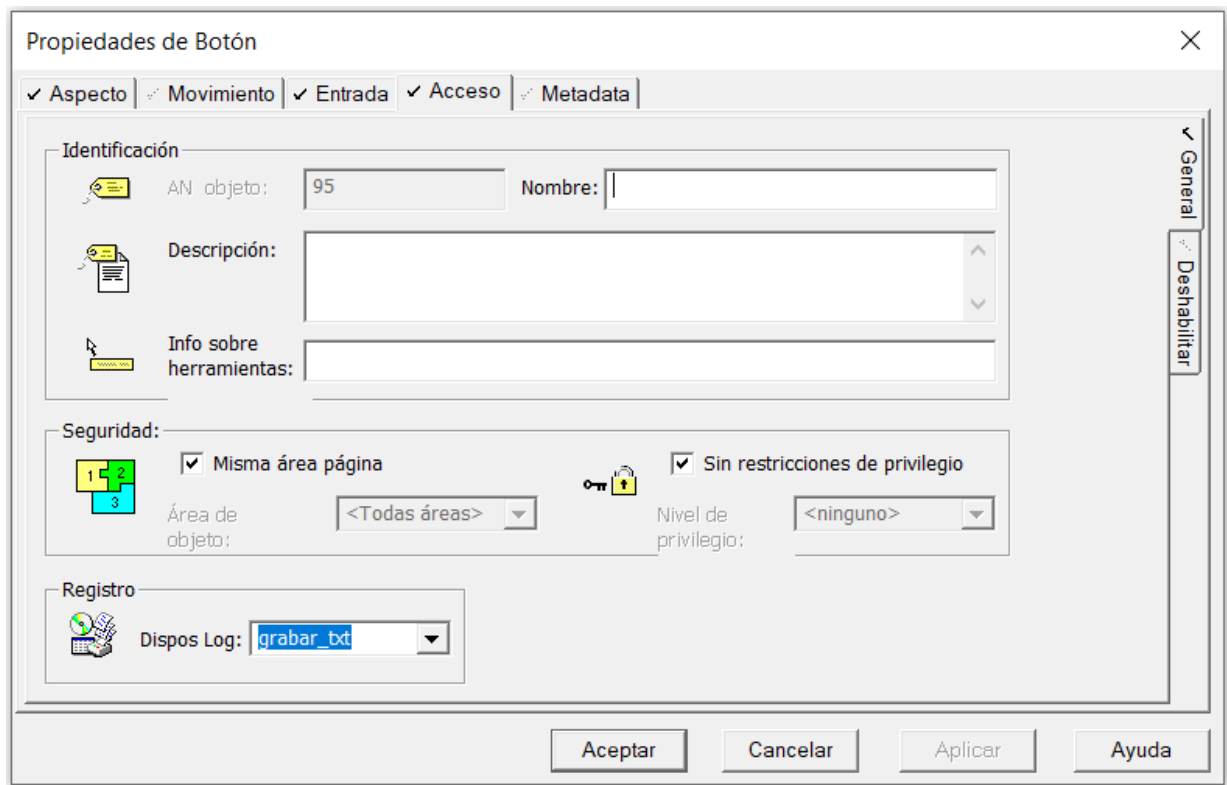


Figura 8.19 Propiedades de Botón - Configuración Log

Y en la pestaña 'entrada' se le indicará que mensaje escribir:

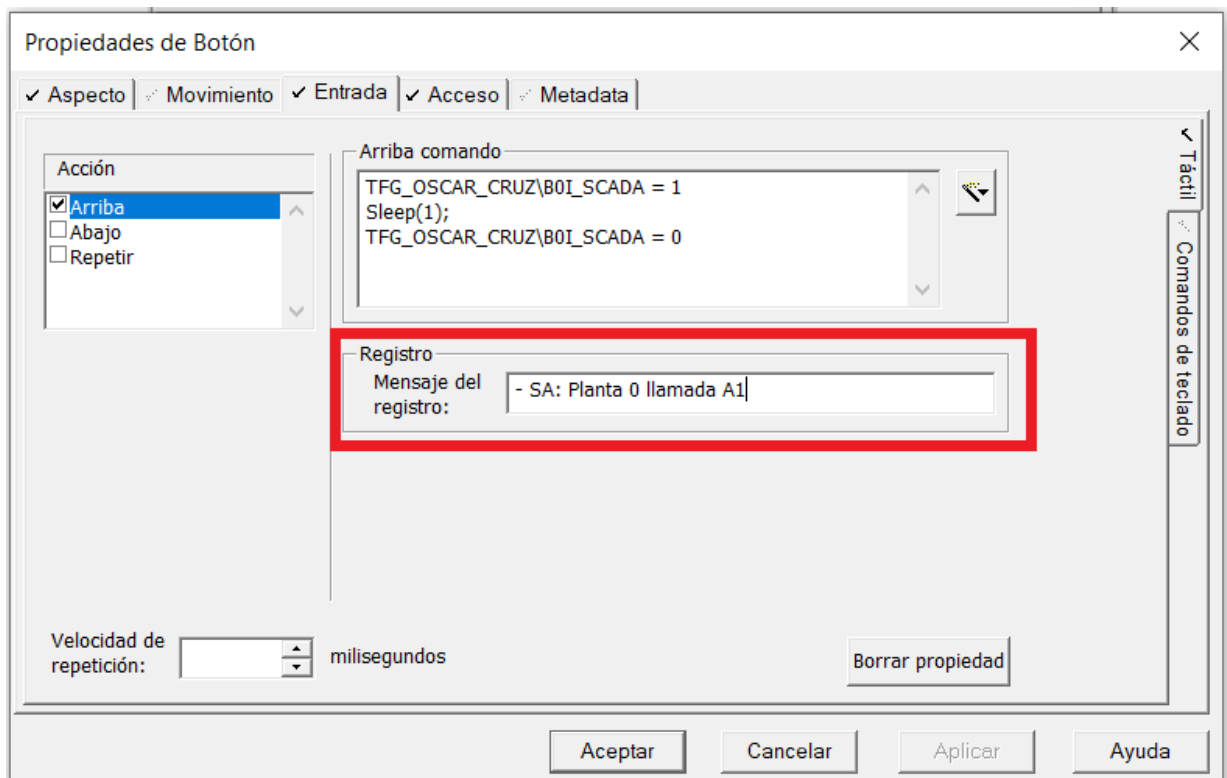


Figura 8.20 Propiedades de Botón - Configuración Log II

8.3.1.3 Lectura de registro

Finalmente, en la dirección especificada en el campo ‘Nombre de archivo’ aparecerá el registro. Es muy importante tener en cuenta que no se debe abrir ese archivo, ya que se cortaría su escritura. En su lugar, se debe copiar y abrir la copia. De este modo, puede leerse el contenido hasta ese momento, pero continuar con su escritura.

Un ejemplo de log sería:

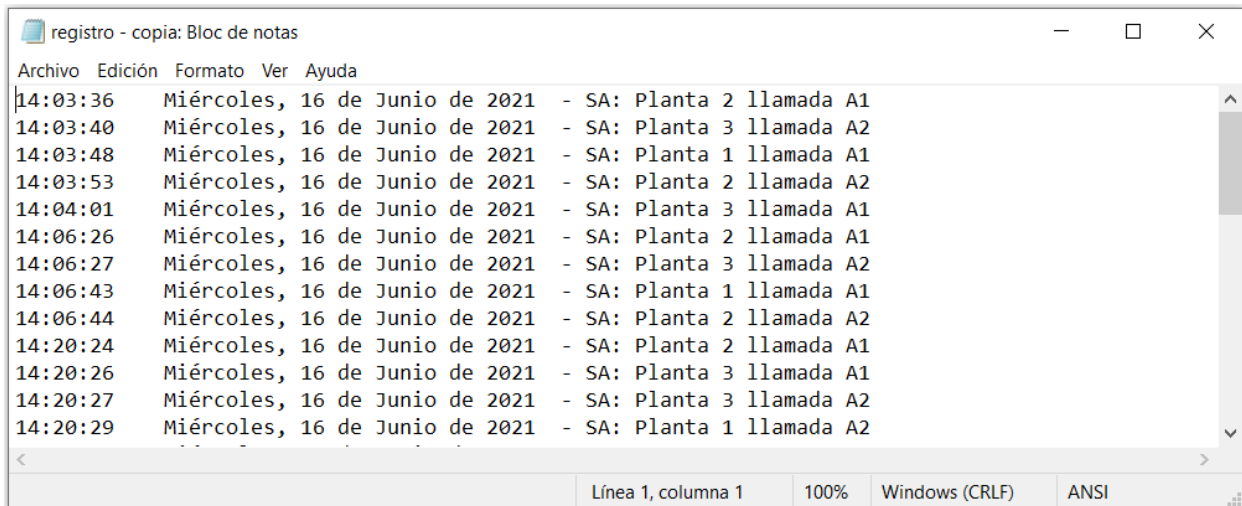


Figura 8.21 Ejemplo Log

8.3.2 Informes

8.3.2.1 Creación de dispositivo

El dispositivo para un informe se crea de forma prácticamente idéntica al dispositivo de un log, tal y como se observa en la Figura 8.12. Sin embargo, hay algunas diferencias.

1. No se añadirá un formato, ya que este se configurará con lenguaje Cicode.
2. En el campo No. de archivos se escribirá -1, ya que si no podría crearse más de un informe.

8.3.2.2 Creación de informe

En la ventana ‘Visualización’ se encuentra la pestaña ‘Informes’, que permite crearlos de forma muy sencilla:

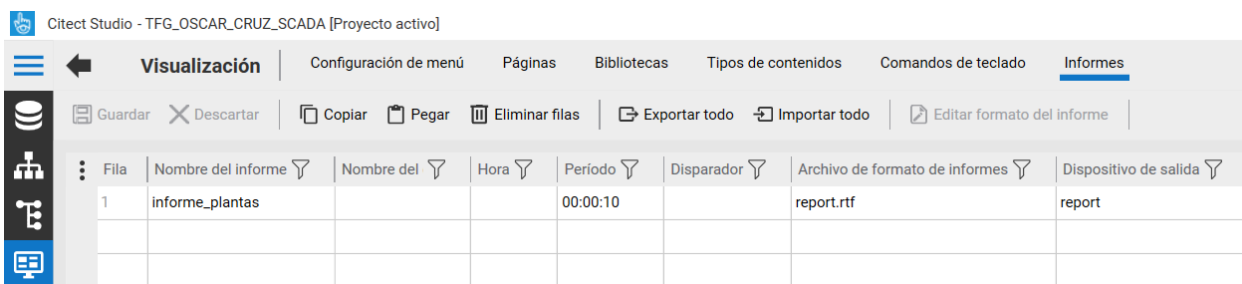


Figura 8.22 Creación de informe

Los campos para rellenar serán:

1. Nombre del informe.
2. Periodo de actualización del informe.
3. Archivo de formato que contendrá, programada en Cicode, la estructura del informe.
4. Dispositivo de salida, que corresponderá con el dispositivo creado anteriormente.

8.3.2.3 Edición del formato del informe

En esa misma pestaña, entre las opciones se encuentra 'Editar formato del informe'. Si se clica en dicha opción aparecerá una ventana de WordPad donde se podrá configurar el formato:

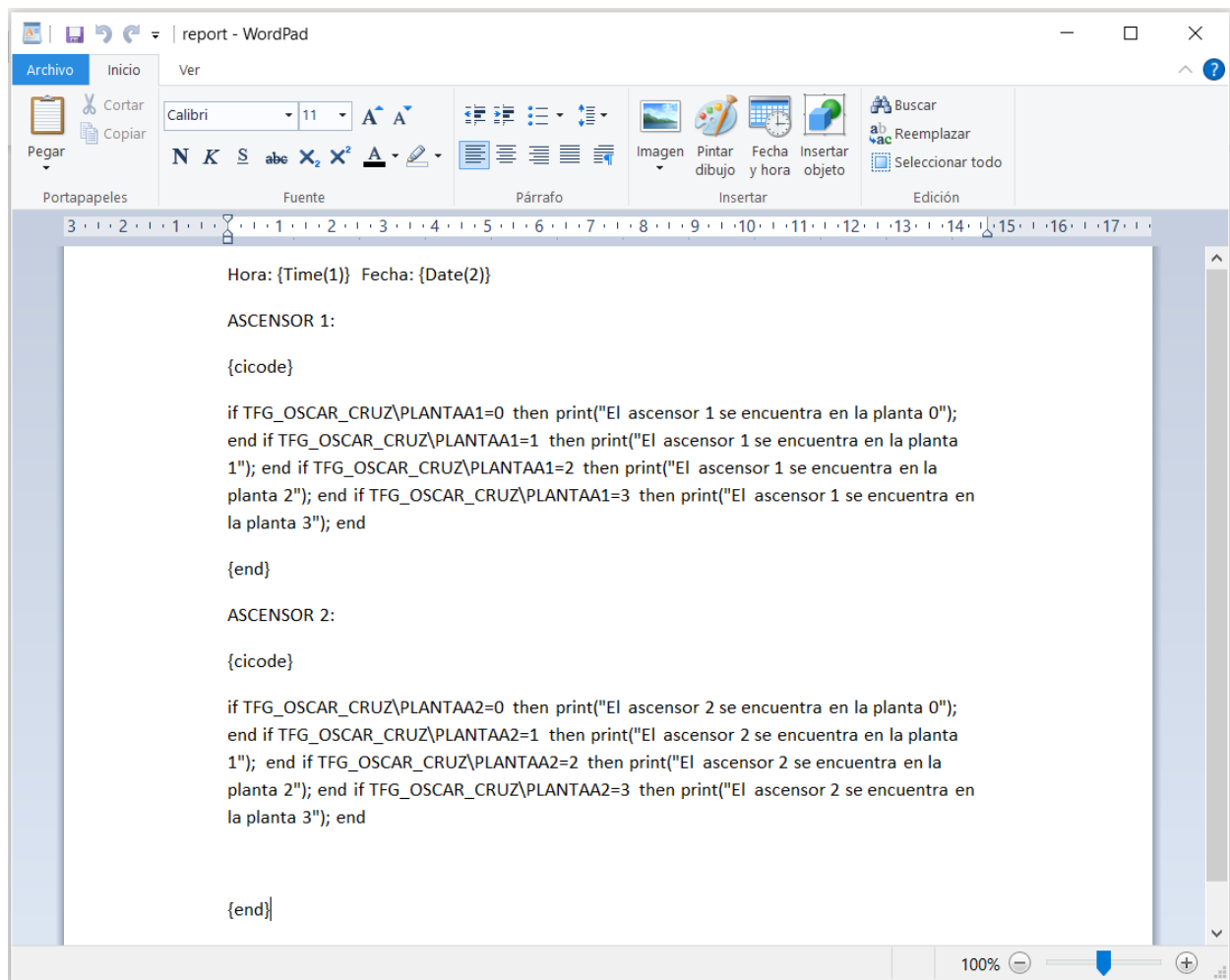


Figura 8.23 Programación de formato de informe

Es importante tener en cuenta que Cicode y WordPad son muy sensibles a la línea en la que se encuentre cada sentencia del código. Es por ello por lo que, si se quiere que dependiendo de una variable se escriba un mensaje u otro, todos los if deben ir seguidos, sin salto de línea.

Otro detalle es que se le debe indicar a WordPad cuando comienza y acaba el lenguaje Cicode. Por ejemplo, en este caso, los títulos ASCENSOR 1 y ASCENSOR 2 se encuentran fuera, mientras que las sentencias de código se encuentran entre corchetes que indican cuando comienza {cicode} y cuando acaba {end} la parte programada.

8.3.2.4 Lectura del informe

Al igual que con los ‘logs’, no se debe abrir directamente el informe, sino que debe copiarse y pegarse para no interrumpir su escritura.

Un ejemplo de informe:

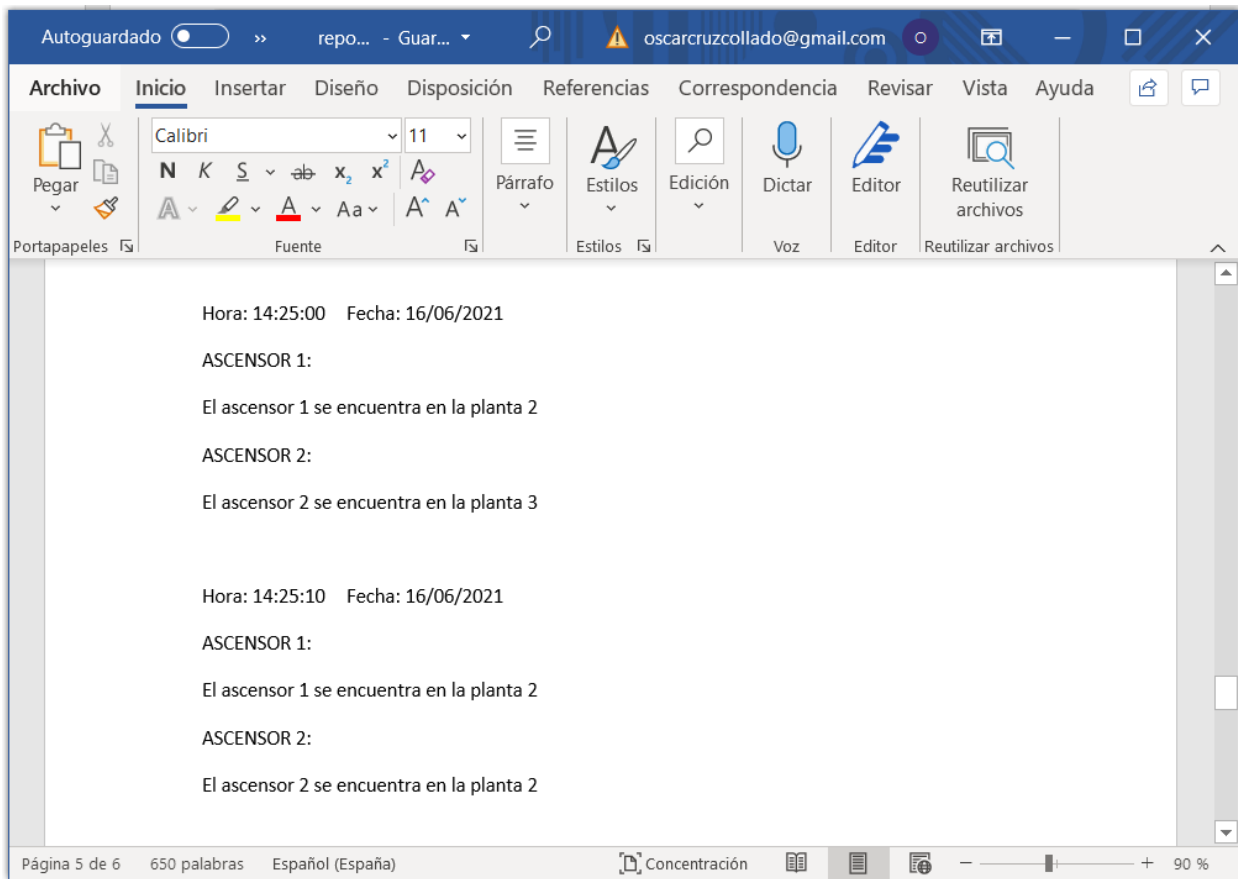


Figura 8.24 Ejemplo informe

9 CONCLUSIONES

Una vez expuesto el trabajo realizado durante el proyecto, junto a una guía de utilización de los distintos programas utilizados, se procede a hacer una valoración de los resultados, así como posibles mejoras y ampliaciones que se podrían realizar.

9.1 Conclusiones

Desde el principio tuve claro que quería que mi Trabajo de Fin de Grado estuviera basado en la automatización. En su momento la asignatura ‘Automatización Industrial’ me pareció una asignatura fundamental ya que, como se explica en la introducción de esta memoria, vivimos en un mundo en constante desarrollo en el que la automatización de procesos juega un papel fundamental en el avance de la tecnología.

Por ello, cuando se me propuso la posibilidad de indagar en el mundo de los sistemas SCADA, con la idea de que mis conocimientos fueran transferidos a otros alumnos en el futuro, me pareció un proyecto motivador.

Afortunadamente, el trabajo se ha visto completado con éxito, como a continuación se expondrá, desarrollando cada una de las partes que lo han conformado.

9.1.1 Programación de la planta

- El trabajo con una planta real conlleva un esfuerzo extra comparado con el esfuerzo necesario en una simulación. Pueden surgir problemas para los que hay que estar preparado, como fallos mecánicos.
- Es conveniente realizar un trabajo lo más general posible, de modo que el algoritmo pueda, con adaptaciones, ser implementados, por ejemplo, en este caso, en distintos ascensores con un número de plantas distinto. Es por ello por lo que, para programar el movimiento del ascensor, se ha utilizado un vector que le indica al algoritmo a que planta ir, en vez de crear un diagrama SFC fijo, que tome decisiones en función de unas ramas establecidas.

9.1.2 Conexión PLC – OPC UA – SCADA

- Para un ingeniero de sistemas, es importante conocer las distintas posibilidades de comunicación entre dispositivos dentro de su ámbito, ya que, por la cantidad de ellos de los que dispone una empresa, se hace necesario centralizar la información, por lo que serán necesarias dichas conexiones.
- El sistema OPC UA ha funcionado de forma más que correcta, sirviendo de puente entre el autómatas y el sistema de supervisión. El router ha hecho perfectamente su labor como asignador de direcciones IP necesarias para que este sistema funcionase.
- La configuración OPC UA – Citect Studio se ha hecho de forma rápida e intuitiva, gracias a la cantidad de material que proporciona la empresa Schneider Electronics tanto en formato escrito como vídeo, además de los vídeos explicativos de usuarios particulares.
- El rendimiento de las conexiones del sistema ha sido bueno, ya que tanto la transmisión de información PLC – SCADA, pasando por el OPC UA, era prácticamente instantánea.

9.1.3 Diseño y funciones SCADA

- La visualización se ha hecho de forma intuitiva, gracias también a la cantidad de información que pone la empresa en manos del usuario.
- La configuración tanto del log como de los informes ha sido rápida, y su funcionamiento ha sido bueno.

9.2 Líneas de mejora

Para posibles proyectos futuros tanto con la planta como con comunicaciones entre PLC y SCADA y diseño de este, se proponen las siguientes mejoras.

9.2.1 Planta de ascensores

- Cambio en los motores, ya que los que se incluyen actualmente son muy ruidosos, dificultando así el poder trabajar con ella en ambientes silenciosos.
- La inclusión de un nuevo módulo de salidas en el PLC, ya que la maqueta tiene dos salidas más de las que le proporciona el autómata. Por ejemplo, un BMX DDM16022 aportaría 8 entradas más, aunque, por no desaprovechar, convendría buscar un módulo con menos salidas.

9.2.2 Conexiones

- Sería conveniente conectar el autómata de la planta al router general del laboratorio de control de la Escuela. Así, se podría acceder desde cualquier punto, y no se tendría que hacer uso de un router exclusivo para ella.
- El servidor OPC UA permite encriptar la información, dotando a los datos de una gran seguridad. Sería interesante indagar en ese campo.

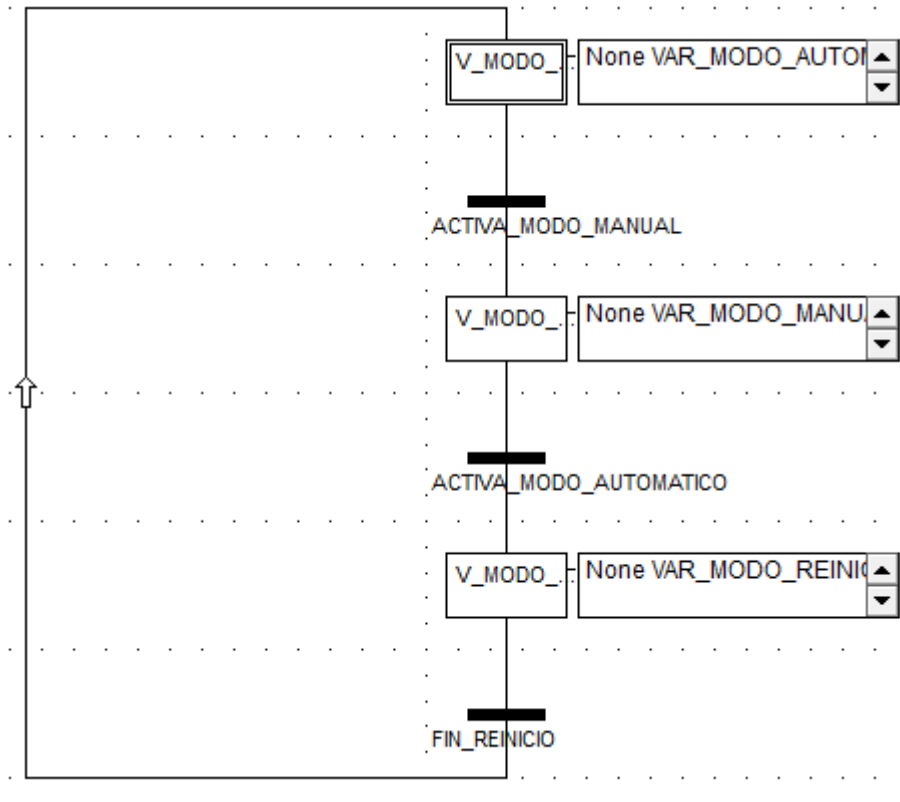
9.2.3 SCADA

El programa Citect Studio, como se ha explicado a lo largo de la presente memoria, tiene muchísimas posibilidades, la mayoría de las cuales no han sido implementadas en este proyecto. Algunas de ellas son:

- Gestión de alarmas.
- Gestión de seguridad.
- Gestión de usuarios.
- Creación de bases de datos SQL.
- Creación de 'Genies', que son objetos gráficos personalizados y parametrizables.
- Estudio de tendencias.
- Posibilidad de arranque automático.

ANEXO A – CÓDIGO DE PROGRAMACIÓN

MODO_NORMAL (SFC)



Control (ST)

```
(*ESCRITURA DE PLANTAS EN PLANTAA1 y PLANTAA2*)
(*Cuando se active alguno de los sensores de planta,
estas variables cambiarán de valor*)
IF (P0A1=TRUE) THEN
    PLANTAA1:=0;

ELSIF (P1A1=TRUE) THEN
    PLANTAA1:=1;

ELSIF (P2A1=TRUE) THEN
    PLANTAA1:=2;

ELSIF (P3A1=TRUE) THEN
    PLANTAA1:=3;
END_IF;

IF (P0A2=TRUE) THEN
    PLANTAA2:=0;

ELSIF (P1A2=TRUE) THEN
    PLANTAA2:=1;

ELSIF (P2A2=TRUE) THEN
    PLANTAA2:=2;

ELSIF (P3A2=TRUE) THEN
    PLANTAA2:=3;
END_IF;

(*ESTADOS DE ASCENSORES*)
IF (SUBIR1 OR BAJAR1 OR PUERTAS1) THEN
    IZQUIERDO_OCUPADO:=TRUE;
ELSE IZQUIERDO_OCUPADO:=FALSE;
END_IF;

IF (SUBIR2 OR BAJAR2 OR PUERTAS2) THEN
    DERECHO_OCUPADO:=TRUE;
ELSE DERECHO_OCUPADO:=FALSE;
END_IF;

IF (DERECHO_OCUPADO AND IZQUIERDO_OCUPADO) THEN
    AMBOS_OCUPADOS:=TRUE;
ELSE AMBOS_OCUPADOS:=FALSE;
END_IF;
```



```
(*PETICIÓN DE PLANTA*)
(*Se utilizan flancos de subida para forzar una
sola pedida por pulsación*)
IF (FSB0) THEN
    PLANTA_PEDIDA:=0;
    P0_PEDIDA:=TRUE;
    NUEVA_LLAMADA:=TRUE;
ELSIF (FSB1S OR FSB1B) THEN
    PLANTA_PEDIDA:=1;
    P1_PEDIDA:=TRUE;
    NUEVA_LLAMADA:=TRUE;
ELSIF (FSB2S OR FSB2B) THEN
    PLANTA_PEDIDA:=2;
    P2_PEDIDA:=TRUE;
    NUEVA_LLAMADA:=TRUE;
ELSIF (FSB3) THEN
    PLANTA_PEDIDA:=3;
    P3_PEDIDA:=TRUE;
    NUEVA_LLAMADA:=TRUE;
END_IF;

(*SUBIDA O BAJADA EN PLANTAS INTERMEDIAS*)
(*Se usará la variable SUBEBAJA_PX para distinguir
si se ha pedido subir o bajar en plantas intermedias*)
IF (FSB1S) THEN SUBEBAJA_P1:=1;
ELSIF (FSB1B) THEN SUBEBAJA_P1:=0;
END_IF;

IF (FSB2S) THEN SUBEBAJA_P2:=1;
ELSIF (FSB2B) THEN SUBEBAJA_P2:=0;
END_IF;
```

```
(*PULSACIÓN DE BOTONES INTERIORES*)
(*Cuando se pulse un botón interior, se apuntará
en el buffer correspondiente, se incrementará el índice
y se activará la variable de planta del ascensor*)
IF (FSB0I) THEN
    BUFFER1[i1]:=0;
    i1:=i1+1;
    P0_PEDIDA_A1:=TRUE;
ELSIF (FSB1I) THEN
    BUFFER1[i1]:=1;
    i1:=i1+1;
    P1_PEDIDA_A1:=TRUE;
ELSIF (FSB2I) THEN
    BUFFER1[i1]:=2;
    i1:=i1+1;
    P2_PEDIDA_A1:=TRUE;
ELSIF (FSB3I) THEN
    BUFFER1[i1]:=3;
    i1:=i1+1;
    P3_PEDIDA_A1:=TRUE;
END_IF;

IF (FSB0D) THEN
    BUFFER2[i2]:=0;
    i2:=i2+1;
    P0_PEDIDA_A2:=TRUE;
ELSIF (FSB1D) THEN
    BUFFER2[i2]:=1;
    i2:=i2+1;
    P1_PEDIDA_A2:=TRUE;
ELSIF (FSB2D) THEN
    BUFFER2[i2]:=2;
    i2:=i2+1;
    P2_PEDIDA_A2:=TRUE;
ELSIF (FSB3D) THEN
    BUFFER2[i2]:=3;
    i2:=i2+1;
    P3_PEDIDA_A2:=TRUE;
END_IF;
```

```

(*PULSACIÓN BOTONES EXTERIORES*)
(*Si los dos están ocupados, acudirá alguno si le viene bien la trayectoria*)
IF (AMBOS_OCUPADOS AND NUEVA_LLAMADA) THEN
  IF (SUBIR1 AND PLANTA_PEDIDA>PLANTAA1) THEN
    IF ((PLANTA_PEDIDA=1 AND SUBEBAJA_P1=1) OR (PLANTA_PEDIDA=2 AND SUBEBAJA_P2=1)) THEN
      BUFFER1[i1]:=PLANTA_PEDIDA;
      i1:=i1+1;

      IF (PLANTA_PEDIDA=1) THEN
        P1_PEDIDA_A1:=TRUE;
      ELSIF (PLANTA_PEDIDA=2) THEN
        P2_PEDIDA_A1:=TRUE;
      END_IF;
    ELSIF (PLANTA_PEDIDA=3) THEN
      BUFFER1[i1]:=PLANTA_PEDIDA;
      i1:=i1+1;
      P3_PEDIDA_A1:=TRUE;
    END_IF;

  ELSIF (SUBIR2 AND PLANTA_PEDIDA>PLANTAA2) THEN
    IF ((PLANTA_PEDIDA=1 AND SUBEBAJA_P1=1) OR (PLANTA_PEDIDA=2 AND SUBEBAJA_P2=1)) THEN
      BUFFER2[i2]:=PLANTA_PEDIDA;
      i2:=i2+1;

      IF (PLANTA_PEDIDA=1) THEN
        P1_PEDIDA_A2:=TRUE;
      ELSIF (PLANTA_PEDIDA=2) THEN
        P2_PEDIDA_A2:=TRUE;
      END_IF;
    ELSIF (PLANTA_PEDIDA=3) THEN
      BUFFER2[i2]:=PLANTA_PEDIDA;
      i2:=i2+1;
      P3_PEDIDA_A2:=TRUE;
    END_IF;

  ELSIF (BAJAR1 AND PLANTA_PEDIDA<PLANTAA1) THEN
    IF ((PLANTA_PEDIDA=1 AND SUBEBAJA_P1=0) OR (PLANTA_PEDIDA=2 AND SUBEBAJA_P2=0)) THEN
      BUFFER1[i1]:=PLANTA_PEDIDA;
      i1:=i1+1;

      IF (PLANTA_PEDIDA=1) THEN
        P1_PEDIDA_A1:=TRUE;
      ELSIF (PLANTA_PEDIDA=2) THEN
        P2_PEDIDA_A1:=TRUE;
      END_IF;
    ELSIF (PLANTA_PEDIDA=0) THEN
      BUFFER1[i1]:=PLANTA_PEDIDA;
      i1:=i1+1;
      P0_PEDIDA_A1:=TRUE;
    END_IF;

  ELSIF (BAJAR2 AND PLANTA_PEDIDA<PLANTAA2) THEN
    IF ((PLANTA_PEDIDA=1 AND SUBEBAJA_P1=0) OR (PLANTA_PEDIDA=2 AND SUBEBAJA_P2=0)) THEN
      BUFFER2[i2]:=PLANTA_PEDIDA;
      i2:=i2+1;

      IF (PLANTA_PEDIDA=1) THEN
        P1_PEDIDA_A2:=TRUE;
      ELSIF (PLANTA_PEDIDA=2) THEN
        P2_PEDIDA_A2:=TRUE;
      END_IF;
    ELSIF (PLANTA_PEDIDA=0) THEN
      BUFFER2[i2]:=PLANTA_PEDIDA;
      i2:=i2+1;
      P0_PEDIDA_A2:=TRUE;
    END_IF;
  END_IF;
END_IF;

```

```

(*Si uno solo está ocupado, irá el otro, a menos que le venga bien la trayectoria*)
ELSIF (DERECHO_OCUPADO AND NUEVA_LLAMADA) THEN
  IF (SUBIR2) THEN
    IF (PLANTA_PEDIDA > PLANTAA2) THEN
      IF (PLANTA_PEDIDA = 1 AND SUBEBAJA_P1 = 1) THEN
        BUFFER2[i2] := PLANTA_PEDIDA;
        i2 := i2 + 1;
        P1_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 1 AND SUBEBAJA_P1 = 0) THEN
        BUFFER1[i1] := PLANTA_PEDIDA;
        i1 := i1 + 1;
        P1_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 2 AND SUBEBAJA_P2 = 1) THEN
        BUFFER2[i2] := PLANTA_PEDIDA;
        i2 := i2 + 1;
        P2_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 2 AND SUBEBAJA_P2 = 0) THEN
        BUFFER1[i1] := PLANTA_PEDIDA;
        i1 := i1 + 1;
        P2_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 3) THEN
        BUFFER2[i2] := PLANTA_PEDIDA;
        i2 := i2 + 1;
        P3_PEDIDA_A2 := TRUE;
      END_IF;
    ELSE
      BUFFER1[i1] := PLANTA_PEDIDA;
      i1 := i1 + 1;

      IF (PLANTA_PEDIDA = 0) THEN
        P0_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 1) THEN
        P1_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 2) THEN
        P2_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 3) THEN
        P3_PEDIDA_A1 := TRUE;
      END_IF;
    END_IF;
  ELSIF (BAJAR2) THEN
    IF (PLANTA_PEDIDA < PLANTAA2) THEN
      IF (PLANTA_PEDIDA = 1 AND SUBEBAJA_P1 = 0) THEN
        BUFFER2[i2] := PLANTA_PEDIDA;
        i2 := i2 + 1;
        P1_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 1 AND SUBEBAJA_P1 = 1) THEN
        BUFFER1[i1] := PLANTA_PEDIDA;
        i1 := i1 + 1;
        P1_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 2 AND SUBEBAJA_P2 = 0) THEN
        BUFFER2[i2] := PLANTA_PEDIDA;
        i2 := i2 + 1;
        P2_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 2 AND SUBEBAJA_P2 = 1) THEN
        BUFFER1[i1] := PLANTA_PEDIDA;
        i1 := i1 + 1;
        P2_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 0) THEN
        BUFFER2[i2] := PLANTA_PEDIDA;
        i2 := i2 + 1;
        P0_PEDIDA_A2 := TRUE;
      END_IF;
    ELSE
      BUFFER1[i1] := PLANTA_PEDIDA;
      i1 := i1 + 1;
    END_IF;
  END_IF;
END_IF;

```

```
ELSE
    BUFFER1[i1]:=PLANTA_PEDIDA;
    i1:=i1+1;

    IF (PLANTA_PEDIDA=0) THEN
        P0_PEDIDA_A1:=TRUE;
    ELSIF (PLANTA_PEDIDA=1) THEN
        P1_PEDIDA_A1:=TRUE;
    ELSIF (PLANTA_PEDIDA=2) THEN
        P2_PEDIDA_A1:=TRUE;
    ELSIF (PLANTA_PEDIDA=3) THEN
        P3_PEDIDA_A1:=TRUE;
    END_IF;
END_IF;
ELSIF (PUERTAS2) THEN
    BUFFER1[i1]:=PLANTA_PEDIDA;
    i1:=i1+1;

    IF (PLANTA_PEDIDA=0) THEN
        P0_PEDIDA_A1:=TRUE;
    ELSIF (PLANTA_PEDIDA=1) THEN
        P1_PEDIDA_A1:=TRUE;
    ELSIF (PLANTA_PEDIDA=2) THEN
        P2_PEDIDA_A1:=TRUE;
    ELSIF (PLANTA_PEDIDA=3) THEN
        P3_PEDIDA_A1:=TRUE;
    END_IF;
END_IF;
```

```

ELSIF (IZQUIERDO_OCUPADO AND NUEVA_LLAMADA) THEN
  IF (SUBIR1) THEN
    IF (PLANTA_PEDIDA > PLANTA_A1) THEN
      IF (PLANTA_PEDIDA = 1 AND SUBEBAJA_P1 = 1) THEN
        BUFFER1[i1] := PLANTA_PEDIDA;
        i1 := i1 + 1;
        P1_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 1 AND SUBEBAJA_P1 = 0) THEN
        BUFFER2[i2] := PLANTA_PEDIDA;
        i2 := i2 + 1;
        P1_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 2 AND SUBEBAJA_P2 = 1) THEN
        BUFFER1[i1] := PLANTA_PEDIDA;
        i1 := i1 + 1;
        P2_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 2 AND SUBEBAJA_P2 = 0) THEN
        BUFFER2[i2] := PLANTA_PEDIDA;
        i2 := i2 + 1;
        P2_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 3) THEN
        BUFFER1[i1] := PLANTA_PEDIDA;
        i1 := i1 + 1;
        P3_PEDIDA_A1 := TRUE;
      END_IF;
    ELSE
      BUFFER2[i2] := PLANTA_PEDIDA;
      i2 := i2 + 1;

      IF (PLANTA_PEDIDA = 0) THEN
        P0_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 1) THEN
        P1_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 2) THEN
        P2_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 3) THEN
        P3_PEDIDA_A2 := TRUE;
      END_IF;
    END_IF;
  ELSIF (BAJAR1) THEN
    IF (PLANTA_PEDIDA < PLANTA_A1) THEN
      IF (PLANTA_PEDIDA = 1 AND SUBEBAJA_P1 = 0) THEN
        BUFFER1[i1] := PLANTA_PEDIDA;
        i1 := i1 + 1;
        P1_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 1 AND SUBEBAJA_P1 = 1) THEN
        BUFFER2[i2] := PLANTA_PEDIDA;
        i2 := i2 + 1;
        P1_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 2 AND SUBEBAJA_P2 = 0) THEN
        BUFFER1[i1] := PLANTA_PEDIDA;
        i1 := i1 + 1;
        P2_PEDIDA_A1 := TRUE;
      ELSIF (PLANTA_PEDIDA = 2 AND SUBEBAJA_P2 = 1) THEN
        BUFFER2[i2] := PLANTA_PEDIDA;
        i2 := i2 + 1;
        P2_PEDIDA_A2 := TRUE;
      ELSIF (PLANTA_PEDIDA = 0) THEN
        BUFFER1[i1] := PLANTA_PEDIDA;
        i1 := i1 + 1;
        P0_PEDIDA_A1 := TRUE;
      END_IF;
    ELSE

```

```
ELSE
    BUFFER2[i2]:=PLANTA_PEDIDA;
    i2:=i2+1;

    IF (PLANTA_PEDIDA=0) THEN
        P0_PEDIDA_A2:=TRUE;
    ELSIF (PLANTA_PEDIDA=1) THEN
        P1_PEDIDA_A2:=TRUE;
    ELSIF (PLANTA_PEDIDA=2) THEN
        P2_PEDIDA_A2:=TRUE;
    ELSIF (PLANTA_PEDIDA=3) THEN
        P3_PEDIDA_A2:=TRUE;
    END_IF;
END_IF;

ELSIF (PUERTAS1) THEN
    BUFFER2[i2]:=PLANTA_PEDIDA;
    i2:=i2+1;

    IF (PLANTA_PEDIDA=0) THEN
        P0_PEDIDA_A2:=TRUE;
    ELSIF (PLANTA_PEDIDA=1) THEN
        P1_PEDIDA_A2:=TRUE;
    ELSIF (PLANTA_PEDIDA=2) THEN
        P2_PEDIDA_A2:=TRUE;
    ELSIF (PLANTA_PEDIDA=3) THEN
        P3_PEDIDA_A2:=TRUE;
    END_IF;
END_IF;
```

```

(*Si están los dos libres, acudirá el más cercano a la planta*)
(*Si ambos están a la misma distancia, irá por defecto el ascensor1, es decir, el izquierdo*)
ELSIF(((NOT IZQUIERDO_OCUPADO) ANE (NOT DERECHO_OCUPADO)) ANE NUEVA_LLAMADA) THEN
    IF (ABS (PLANTA_PEDIDA-PLANTAA1) <= ABS (PLANTA_PEDIDA-PLANTAA2)) THEN
        DECISION:=1;
        BUFFER1[i1]:=PLANTA_PEDIDA;
        i1:=i1+1;

        IF (PLANTA_PEDIDA=0) THEN
            P0_PEDIDA_A1:=TRUE;
        ELSIF (PLANTA_PEDIDA=1) THEN
            P1_PEDIDA_A1:=TRUE;
        ELSIF (PLANTA_PEDIDA=2) THEN
            P2_PEDIDA_A1:=TRUE;
        ELSIF (PLANTA_PEDIDA=3) THEN
            P3_PEDIDA_A1:=TRUE;
        END_IF;
    ELSE
        DECISION:=2;
        BUFFER2[i2]:=PLANTA_PEDIDA;
        i2:=i2+1;

        IF (PLANTA_PEDIDA=0) THEN
            P0_PEDIDA_A2:=TRUE;
        ELSIF (PLANTA_PEDIDA=1) THEN
            P1_PEDIDA_A2:=TRUE;
        ELSIF (PLANTA_PEDIDA=2) THEN
            P2_PEDIDA_A2:=TRUE;
        ELSIF (PLANTA_PEDIDA=3) THEN
            P3_PEDIDA_A2:=TRUE;
        END_IF;
    END_IF;
END_IF;

(*DESACTIVAMOS NUEVA_LLAMADA PARA EVITAR QUE SIEMPRE ENTRE EN ALGÚN IF*)
NUEVA_LLAMADA:=FALSE;

```



```

(*ORDENAR VECTORES CONSTANTEMENTE*)
(*Si un ascensor está subiendo, interesa que su vector se ordene de menor a mayor. En este caso se debe
tener cuidado, ya que los valores -1 del vector serían los mas pequeños y se pondrían al principio, por
lo que se deben ignorar*)
(*Si un ascensor está bajando, se debe ordenar su vector de mayor a menor. En este caso, no hay problema
con los valores -1*)
j1:=0;
k1:=0;
j2:=0;
k2:=0;

IF(SUBIR1) THEN
  FOR j1:=0 TO 4 BY 1 DO
    FOR k1:=0 TO 4 BY 1 DO
      IF ((BUFFER1[j1]<BUFFER1[k1]) AND (BUFFER1[k1]<>-1) AND (BUFFER1[j1]<>-1)) THEN
        AUX:=BUFFER1[j1];
        BUFFER1[j1]:=BUFFER1[k1];
        BUFFER1[k1]:=AUX;
      END_IF;
    END_FOR;
  END_FOR;

ELSIF(BAJAR1) THEN
  FOR j1:=0 TO 4 BY 1 DO
    FOR k1:=0 TO 4 BY 1 DO
      IF (BUFFER1[j1]>BUFFER1[k1]) THEN
        AUX:=BUFFER1[j1];
        BUFFER1[j1]:=BUFFER1[k1];
        BUFFER1[k1]:=AUX;
      END_IF;
    END_FOR;
  END_FOR;
END_IF;

IF(SUBIR2) THEN
  FOR j2:=0 TO 4 BY 1 DO
    FOR k2:=0 TO 4 BY 1 DO
      IF ((BUFFER2[j2]<BUFFER2[k2]) AND (BUFFER2[k2]<>-1) AND (BUFFER2[j2]<>-1)) THEN
        AUX:=BUFFER2[j2];
        BUFFER2[j2]:=BUFFER2[k2];
        BUFFER2[k2]:=AUX;
      END_IF;
    END_FOR;
  END_FOR;

ELSIF(BAJAR2) THEN
  FOR j2:=0 TO 4 BY 1 DO
    FOR k2:=0 TO 4 BY 1 DO
      IF (BUFFER2[j2]>BUFFER2[k2]) THEN
        AUX:=BUFFER2[j2];
        BUFFER2[j2]:=BUFFER2[k2];
        BUFFER2[k2]:=AUX;
      END_IF;
    END_FOR;
  END_FOR;
END_IF;

```

```
(*LLEGADA A PLANTA*)
(*Cuando se llega a una planta, se moverán todos los elementos del vector una casilla
hacia la izquierda y se decrementará el índice. Además, se desactivará la variable de
Planta pedida correspondiente*)
IF (LLEGA_PLANTA_A1) THEN
    FOR j1:=0 TO 3 BY 1 DO
        BUFFER1[j1]:=BUFFER1[j1+1];
    END_FOR;
    i1:=i1-1;

    IF PLANTAA1=0 THEN P0_PEDIDA:=FALSE;
    ELSIF PLANTAA1=1 THEN P1_PEDIDA:=FALSE;
    ELSIF PLANTAA1=2 THEN P2_PEDIDA:=FALSE;
    ELSIF PLANTAA1=3 THEN P3_PEDIDA:=FALSE;
    END_IF;

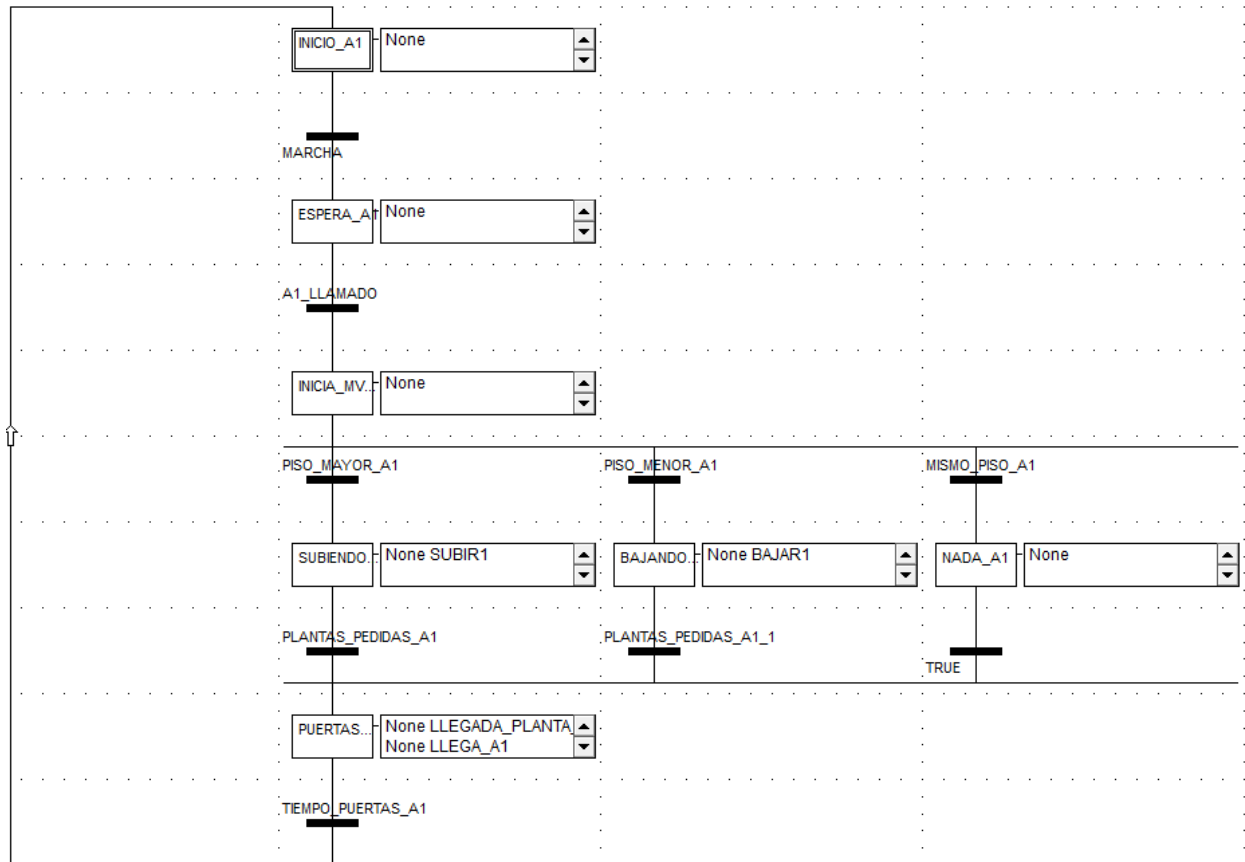
    IF PLANTAA1=0 THEN P0_PEDIDA_A1:=FALSE;
    ELSIF PLANTAA1=1 THEN P1_PEDIDA_A1:=FALSE;
    ELSIF PLANTAA1=2 THEN P2_PEDIDA_A1:=FALSE;
    ELSIF PLANTAA1=3 THEN P3_PEDIDA_A1:=FALSE;
    END_IF;
END_IF;

IF (LLEGA_PLANTA_A2) THEN
    FOR j2:=0 TO 3 BY 1 DO
        BUFFER2[j2]:=BUFFER2[j2+1];
    END_FOR;
    i2:=i2-1;

    IF PLANTAA2=0 THEN P0_PEDIDA:=FALSE;
    ELSIF PLANTAA2=1 THEN P1_PEDIDA:=FALSE;
    ELSIF PLANTAA2=2 THEN P2_PEDIDA:=FALSE;
    ELSIF PLANTAA2=3 THEN P3_PEDIDA:=FALSE;
    END_IF;

    IF PLANTAA2=0 THEN P0_PEDIDA_A2:=FALSE;
    ELSIF PLANTAA2=1 THEN P1_PEDIDA_A2:=FALSE;
    ELSIF PLANTAA2=2 THEN P2_PEDIDA_A2:=FALSE;
    ELSIF PLANTAA2=3 THEN P3_PEDIDA_A2:=FALSE;
    END_IF;
END_IF;
```

Movimiento_A1 (SFC)



Muchas de las transiciones han tenido que hacerse en secciones aparte, pues son comparaciones numéricas. Algunos ejemplos:

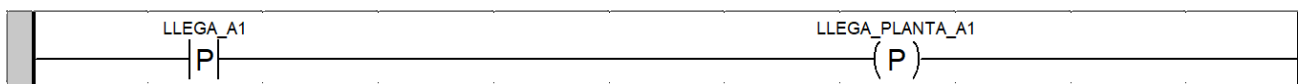
A1_LLAMADO:

```
(BUFFER1[0] <> -1)
```

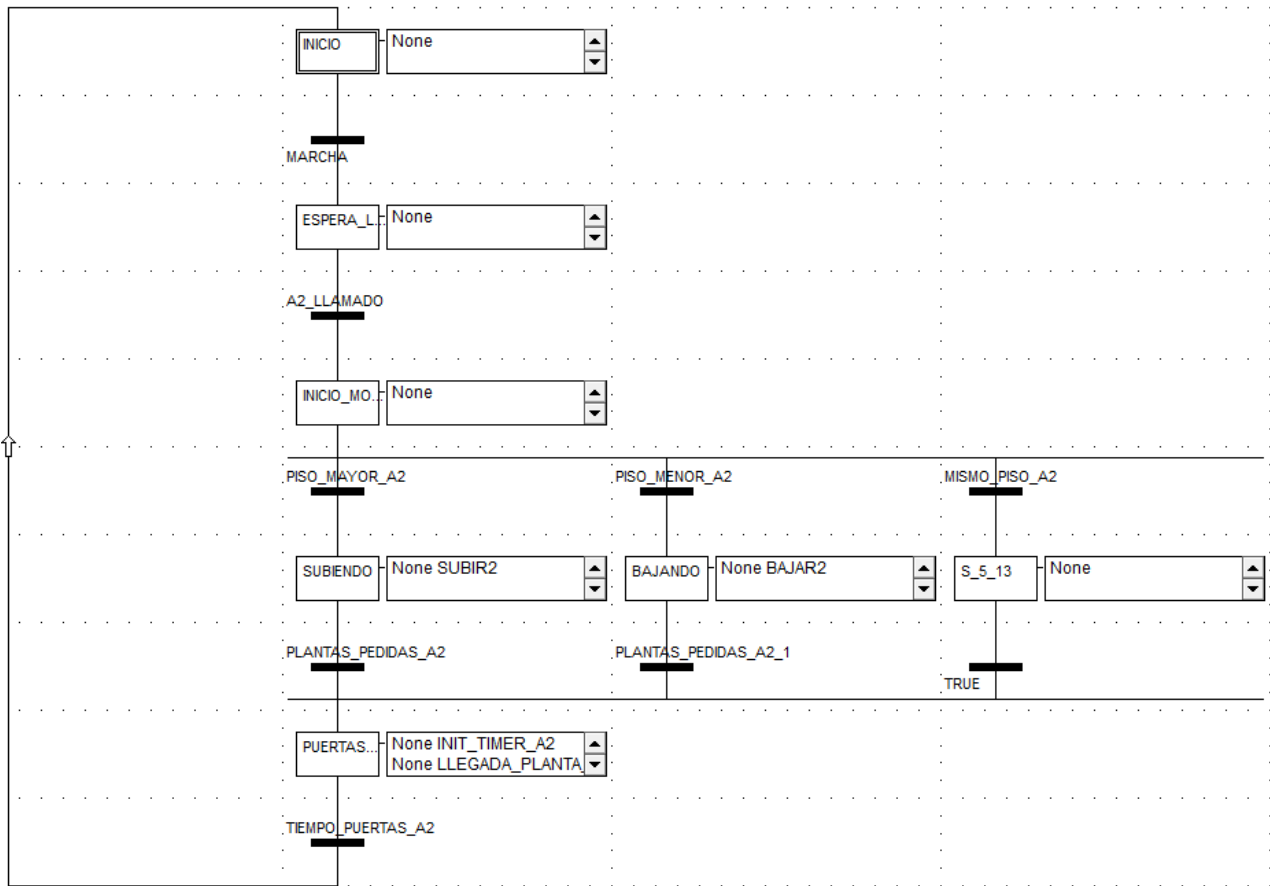
PISO_MAYOR_A1:

```
(BUFFER1[0] > PLANTAA1)
```

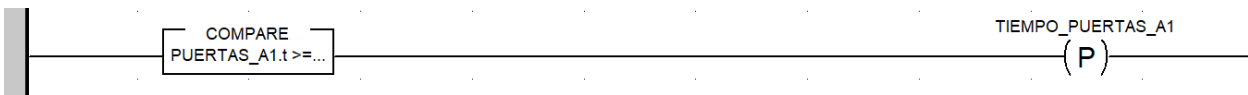
Para avisar de que un ascensor ha llegado a una planta se activa la variable LLEGA_PLANTA_A1:



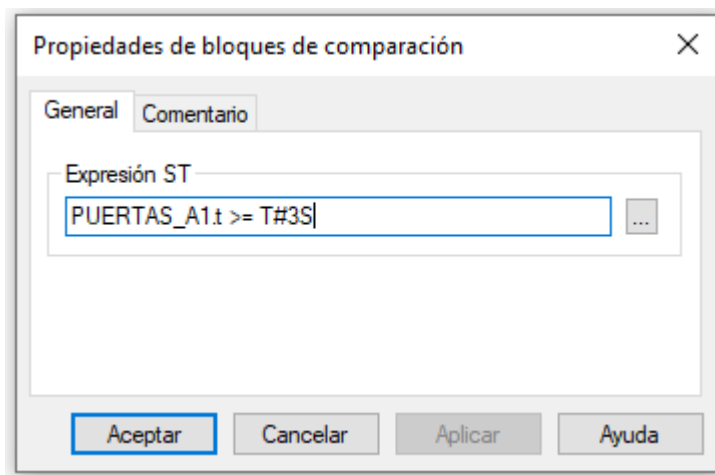
Movimiento_A2 (SFC)



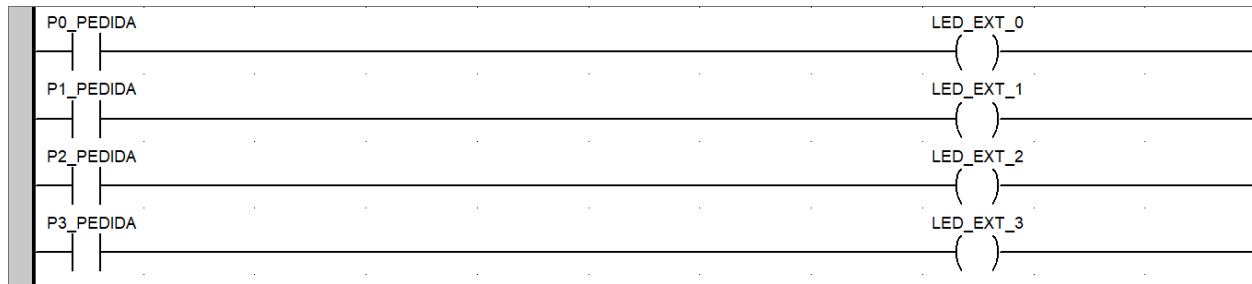
El temporizador se ha realizado mediante un comparador:



Con la siguiente configuración:



LUCES (LD)



MODO_SEMIAUTOMATICO (ST)

```
(*Cuando se recibe un flanco de subida del botón del SCADA,
se usa el mismo algoritmo que cuando se hace una llamada desde
un botón interno. Se apunta directamente en el buffer correspondiente,
se incrementa el índice y se activa la variable correspondiente*)
IF (FSB0I_SCADA) THEN
    BUFFER1[i1]:=0;
    i1:=i1+1;
    P0_PEDIDA_A1:=TRUE;
ELSIF (FSB1I_SCADA) THEN
    BUFFER1[i1]:=1;
    i1:=i1+1;
    P1_PEDIDA_A1:=TRUE;
ELSIF (FSB2I_SCADA) THEN
    BUFFER1[i1]:=2;
    i1:=i1+1;
    P2_PEDIDA_A1:=TRUE;
ELSIF (FSB3I_SCADA) THEN
    BUFFER1[i1]:=3;
    i1:=i1+1;
    P3_PEDIDA_A1:=TRUE;
END_IF;

IF (FSB0D_SCADA) THEN
    BUFFER2[i2]:=0;
    i2:=i2+1;
    P0_PEDIDA_A2:=TRUE;
ELSIF (FSB1D_SCADA) THEN
    BUFFER2[i2]:=1;
    i2:=i2+1;
    P1_PEDIDA_A2:=TRUE;
ELSIF (FSB2D_SCADA) THEN
    BUFFER2[i2]:=2;
    i2:=i2+1;
    P2_PEDIDA_A2:=TRUE;
ELSIF (FSB3D_SCADA) THEN
    BUFFER2[i2]:=3;
    i2:=i2+1;
    P3_PEDIDA_A2:=TRUE;
END_IF;
```

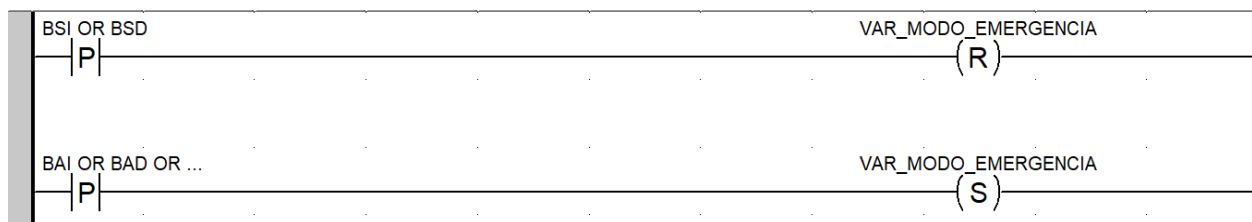
MODO_SEMIAUTOMATICO (LD)

B0	(P)	FSB0	(P)
B1S	(P)	FSB1S	(P)
B1B	(P)	FSB1B	(P)
B2S	(P)	FSB2S	(P)
B2B	(P)	FSB2B	(P)
B3	(P)	FSB3	(P)
B0I	(P)	FSB0I	(P)
B1I	(P)	FSB1I	(P)
B2I	(P)	FSB2I	(P)
B3I	(P)	FSB3I	(P)
B0D	(P)	FSB0D	(P)
B1D	(P)	FSB1D	(P)
B2D	(P)	FSB2D	(P)
B3D	(P)	FSB3D	(P)
B0I_SCADA	(P)	FSB0I_SCADA	(P)
B1I_SCADA	(P)	FSB1I_SCADA	(P)
B2I_SCADA	(P)	FSB2I_SCADA	(P)
B3I_SCADA	(P)	FSB3I_SCADA	(P)
B0D_SCADA	(P)	FSB0D_SCADA	(P)
B1D_SCADA	(P)	FSB1D_SCADA	(P)
B2D_SCADA	(P)	FSB2D_SCADA	(P)
B3D_SCADA	(P)	FSB3D_SCADA	(P)

MODO_MANUAL (LD)



ACTIVA_EMERGENCIA (LD)



MODO_EMERGENCIA (ST)

```
IF VAR_MODO_EMERGENCIA THEN
    SUBIR1:=FALSE;
    BAJAR1:=FALSE;
    SUBIR2:=FALSE;
    BAJAR2:=FALSE;
    VAR_MODO_AUTOMATICO:=FALSE;
    VAR_MODO_REINICIO:=FALSE;

    BUFFER1[0]:=-1;
    BUFFER1[1]:=-1;
    BUFFER1[2]:=-1;
    BUFFER1[3]:=-1;
    BUFFER1[4]:=-1;

    BUFFER2[0]:=-1;
    BUFFER2[1]:=-1;
    BUFFER2[2]:=-1;
    BUFFER2[3]:=-1;
    BUFFER2[4]:=-1;

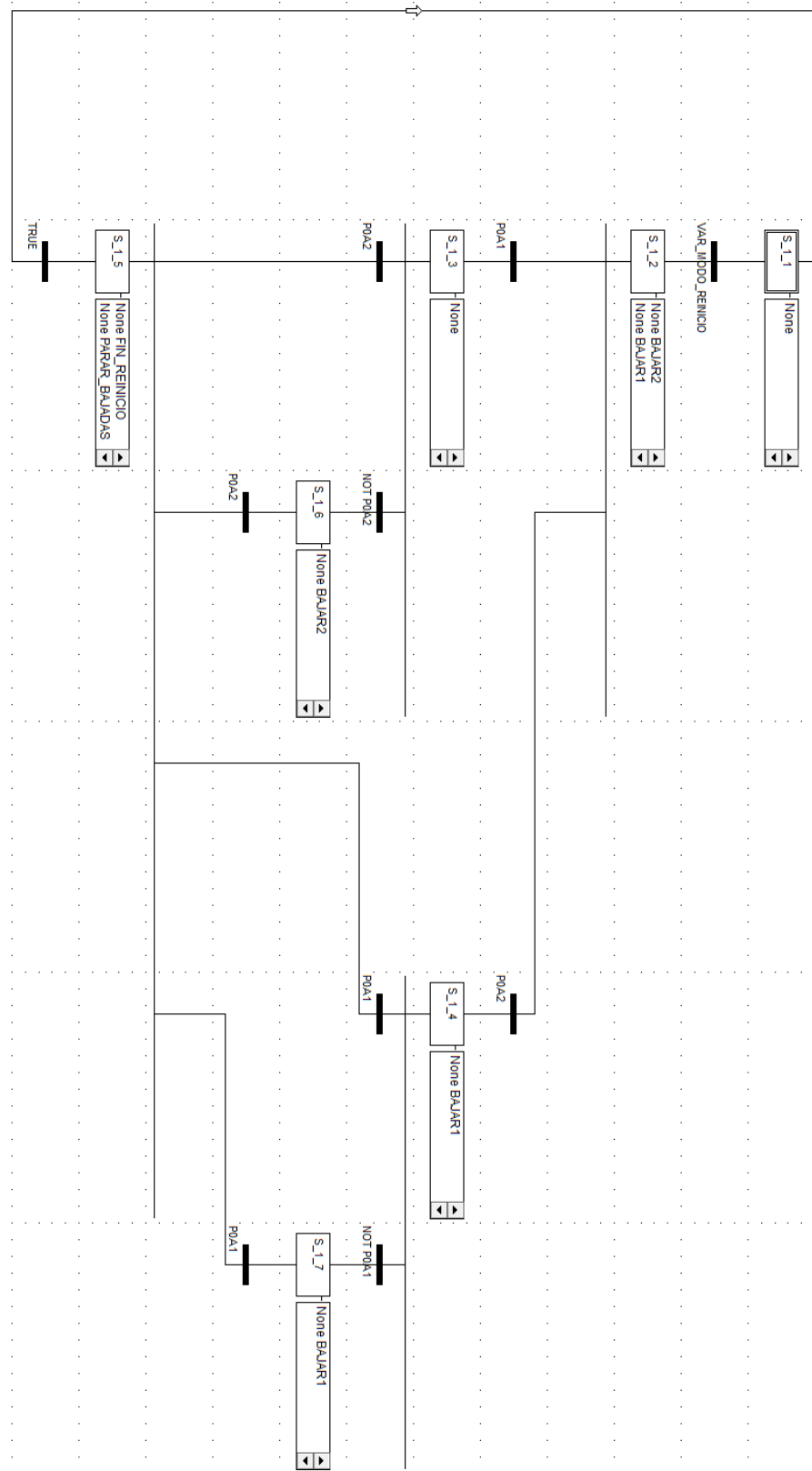
    i1:=0;
    i2:=0;

    SALIDA_INITSFC_A1:=INITCHART(Movimiento_A1, TRUE);
    SALIDA_INITSFC_A2:=INITCHART(Movimiento_A2, TRUE);

END_IF;

IF ACTIVA_MODO_MANUAL THEN
    VAR_MODO_EMERGENCIA:=FALSE;
END_IF;
```


MODO_REINICIO (SFC)



REFERENCIAS

- [1] Daniel Paredes García, *Automatización de un Sistema de Ascensores*, 2017.
- [2] Francisco Rodriguez Rubio, *Sistemas de ascensores controlados por un microprocesador*, 1981.
- [3] Sothis, *SCADA. ¿Qué es y qué permite hacer?* [online], <https://www.sothis.tech/scada-que-es-y-que-permite-hacer/>
- [4] Omnicon, *MES para todos* [online], <https://omnicon.co/es/mes-para-todos/>
- [5] CIC, *Industria 4.0, la cuarta revolución industrial y la inteligencia operacional* [online], <https://www.cic.es/industria-40-revolucion-industrial/>
- [6] Wikipedia, *Revoluciones industriales* [online], https://es.wikipedia.org/wiki/Revoluciones_industriales
- [7] Opiron, *¿Por qué OPC? La torre de Babel de las comunicaciones industriales* [online], <https://www.opiron.com/opc-torre-babel-comunicaciones/>
- [8] José Casares, *Historia de OPC* [online], <http://josecasares.com/historia-de-opc/>
- [9] Opiron, *5 Diferencias entre OPC UA y OPC Clásico* [online], <https://www.opiron.com/5-diferencias-entre-opc-ua-y-opc-clasico/>
- [10] Departamento de Ingeniería Mecánica, Universidad de Salamanca, *Ascensores* [online], <http://dim.usal.es/areaim/guia%20P.%20I/ascensores.htm>
- [11] GMV Eurolift, *¿Cómo funciona un cuadro de maniobras?* [online], <http://blog.gmveurolift.es/como-funciona-un-cuadro-de-maniobras/>
- [12] Ascensores J. Pascual, *Cables de acero en los ascensores* [online], <http://ascensoresjpascual.blogspot.com/2014/02/cables-de-acero-en-los-ascensores.html>
- [13] GMV Eurolift, *Foso: características y requerimientos* [online], <http://blog.gmveurolift.es/foso-caracteristicas-y-requerimientos/>
- [14] Schneider Electric, *Ecostruxure Control Expert: Manual de instalación* [online], 2020
- [15] Schneider Electric, *Ecostruxure Control Expert: Lenguajes y estructura del programa. Manual de referencia* [online], 2020
- [16] Francisco Javier Ramos Ramirez, *Control y supervisión de célula de fabricación flexible, Capítulo 8: Guía gemma* [online], 2008, <http://bibing.us.es/proyectos/abreproy/4553/fichero/08+-+Gu%C3%ADa+GEMMA.pdf>

-
- [17] Schneider Electric, *Youtube: Configurar la dirección IP del PLC M340 a través del Unity Pro XL* [online], <https://www.youtube.com/watch?v=1Z8StKujftk>
- [18] Schneider Electric, *Youtube: How to Enable OPC UA Communication Between Modicon PLCs and CitectSCADA Schneider Electric Support* [online], <https://www.youtube.com/watch?v=hBGqXfAqwxY/>
- [19] Schneider Electric, *Citect SCADA 2018 Installation and Configuration Guide* [online], <https://www.se.com/ww/en/faqs/FA361839/>
- [20] Instituto Schneider Electric de Formación, *Curso Vijeo Citect* [online], http://www.infopl.net/files/descargas/schneider/infoPLC_01_VIJE0_CITECT_Curso.pdf