



UNIVERSITAT JAUME I

ESCOLA SUPERIOR DE TECNOLOGIA I CIÈNCIES

EXPERIMENTALS

GRADO EN INGENIERÍA EN TECNOLOGÍAS

INDUSTRIALES

***PROYECTO DE AUTOMATIZACIÓN DEL
CENTRO DE SALUD DE VINAROS PARA
SU FUNCIONAMIENTO COMO EDIFICIO
INTELIGENTE***

TRABAJO FIN DE GRADO

AUTOR/A

Andrés Tendero Vegas

DIRECTOR/A

Julio Ariel Romero Pérez

Castellón, 4 Julio de 2020

Agradecimientos

En primer lugar, quiero agradecer a mis familiares por su apoyo incondicional, en especial a mis padres Miguel Andrés y Leonor.

A los profesionales de la empresa DISINEL los cuales han sido clave para poder desarrollar este proyecto.

Por último, a mi tutor Julio Ariel por las ayudas y consejos recibidos en la realización del proyecto.

ÍNDICE GENERAL

ÍNDICE TABLAS.....	6
ÍNDICE IMÁGENES.....	8
ÍNDICE ABREVIATURAS.....	10
MEMORIA.....	12
ANEXOS.....	86
PRESUPUESTO.....	154
PLIEGO DE CONDICIONES.....	158

ÍNDICE TABLAS

Tabla 1: Entradas y Salidas máximas del sector de Climatización por planta.	35
Tabla 2. Entradas y Salidas necesarias por planta. Total del sector de climatización.	35
Tabla 3. Entradas y Salidas necesarias por planta. Total del sector de iluminación. ...	36
Tabla 4. Entradas y Salidas necesarias por planta. Total del sector de seguridad	37
Tabla 5. Tabla final del método de decisión del analizador de redes.....	38
Tabla 6. Número de tarjetas EL1008 con su distribución en sectores.	41
Tabla 7. Número de tarjetas EL2008 con su distribución en sectores.	41
Tabla 8. Número de tarjetas EL3054 con su distribución en sectores.	41
Tabla 9. Número de tarjetas EL3204/3314 con su distribución en sectores.....	42
Tabla 10. Número de tarjetas EL4024 con su distribución en sectores.	42
Tabla 11. Número de tarjetas EL3423 con su distribución en sectores.	42
Tabla 12. Número de tarjetas EK1100 con su distribución en sectores.	42
Tabla 13. Resumen del presupuesto del proyecto.....	82
Tabla 14. Datos de partida del método de las jerarquías analíticas.	90
Tabla 15. Normalización de la valoración de las alternativas	90
Tabla 16. Método de las jerarquías analíticas.	90
Tabla 17. Decisión del método de las jerarquías analíticas.	90
Tabla 18. Características de las tarjetas a estudiar en el método.	91
Tabla 19. Distribución zonas iluminación centro salud Vinaoz.	101
Tabla 20. Distribución zonas climatización y climatizadores centro salud Vinaroz	102
Tabla 21. Distribución seguridad centro salud Vinaroz.	104
Tabla 22. Presupuesto de material 1: PC.....	156
Tabla 23. Presupuesto de material 2: Licencias de software.	156
Tabla 24. Presupuesto de material 3: Tarjetas de los dispositivos.	156
Tabla 25. Presupuesto de ejecución de material.	156
Tabla 26. Presupuesto total.	157

ÍNDICE IMÁGENES

Imagen 1. Esquema modelo de comunicación Dispositivo a Dispositivo	20
Imagen 2. Esquema modelo de comunicación Dispositivo a la Nube.....	21
Imagen 3. Esquema modelo de comunicación Dispositivo a puerta enlace. FUENTE: Tschofenig, H., et.al., Architectural Considerations in Smart Object Networking. Tech. no. RFC 7452. IAB, marzo de 2015. Web.	21
Imagen 4. Fuente: Revista internacional PC Control BECKHOFF.....	22
Imagen 5. Gráfico de la evolución del consumo eléctrico del hospital. Fuente: (Artículo ¿Machine Learning o Smart Humans? Aplicado a monitorización y gestión energética de la demanda en edificios (Francisco Conesa Cervantes, José Angel Peinador Aguilar, 19/10/2017)).	24
Imagen 6. Esquema de un sistema de control centralizado. Fuente: https://blogs.deusto.es/master-informatica/tipos-de-sistemas-de-control-industrial/	25
Imagen 7. Esquema de un sistema de control distribuido. Fuente: Revista internacional beckhoof_stage_technology	26
Imagen 8. Principio de funcionamiento del protocolo de comunicación de EtherCAT. Fuente: Revista EtherCAT Technology Group. Folleto EtherCAT- el bus de campo Ethernet.	29
Imagen 9. Esquema sincronización de equipos con EtherCAT. Fuente: Revista EtherCAT Technology Group. Folleto EtherCAT- el bus de campo Ethernet.	30
Imagen 10. Sistemas de redundancia con EtherCAT. Fuente: Revista EtherCAT Technology Group. Folleto EtherCAT- el bus de campo Ethernet.	30
Imagen 10. Esquema modelo centralizado del sistema de control del centro de salud de Vinaroz.....	32
Imagen 11. Esquema modelo distribuido del sistema de control del centro de salud de Vinaroz.....	33
Imagen 13. Esquema de funcionamiento de un temporizador tipo TON.....	45
Fuente: View Help TwinCAT BECKHOFF.	45
Imagen 14. Visualización principal ubicada en la sala de control.....	73
Imagen 15. Visualización principal de cada planta.	74
Imagen 16. Visualización de la pantalla de climatización en cada planta.	74
Imagen 17. Visualización de la pantalla que muestra las alarmas del sistema de climatización.....	75
Imagen 18. Visualización de la pantalla que permite modificar los parámetros generales del sistema.	75
Imagen 19. Visualización para acceder a las configuraciones horarias.	76

Imagen 20. Visualización de la pantalla de control de los fan coil.	77
Imagen 21. Visualización de la planta 1 de iluminación.	77
Imagen 22. Visualización de la consulta de medicina general 1 perteneciente al grupo 2 de iluminación.	78
Imagen 23. Visualización del grupo de persianas de la planta 1 en la zona 1.	78
Imagen 24. Visualización de la pantalla para acceder a los horarios para su modificación.	79
Imagen 25. Visualización de la pantalla de configuración de los horarios del sector de iluminación.	79
Imagen 26. Visualización de la seguridad de la planta 1.	80
Imagen 27. Esquema planta sótano del centro de salud de Vinaroz.	93
Imagen 28. Esquema planta baja del centro de salud de Vinaroz.	94
Imagen 29. Esquema planta primera del centro de salud de Vinaroz.	95
Imagen 30. Esquema planta segunda del centro de salud de Vinaroz.	96

ÍNDICE ABREVIATURAS

PID (Controlador proporcional, integral y derivativo)

IoT (Internet of Things)

PLC (Programmable logic controller)

IP (Internet Protocol)

UIT (Unión Internacional del Comunicaciones)

GPS (Global positioning system)

WI-FI

MQTT (MQ Telemetry Transport)

PC (Personal Computer, Ordenador Personal)

CAN (Controller Area Network)

ASI (AS Interface)

DALI (Digital Addressable Lighting Interface)

DMA (Directy Memory Acces)

MAC (Media Access Control)

CPU (Central Processing Unit)

TÜV (Technischer Überwachungs-Verein)

HMI (Human Machine Interface)

SCADA (Supervisory Control And Data Acquisition)

FB (Function Block)

ADS (Automatic Dependent Surveillance Broadcast)

OPC-UA (OPC Unified Architecture)

POU (Program Organization Unit)

MEMORIA

ÍNDICE MEMORIA	
ÍNDICE GENERAL	4
ÍNDICE TABLAS	6
ÍNDICE IMÁGENES	8
ÍNDICE ABREVIATURAS.....	10
MEMORIA.....	12
1. OBJETIVO DEL PROYECTO.....	18
2. ALCANCE DEL PROYECTO.....	18
3. ESTUDIOS PREVIOS.	19
3.1. IoT.	19
3.1.1 DEFINICIÓN.....	19
3.1.2 APLICACIONES.	20
3.1.3 MODELOS DE COMUNICACIÓN.....	20
3.2 SMART BUILDINGS.....	22
3.2.1 DEFINICIÓN.....	22
3.2.2ELEMENTOS NECESARIOS.....	22
3.2.3BENEFICIOS DE LOS EDIFICIOS INTELIGENTES.....	23
4. ANALISIS DE SOLUCIONES.	25
4.1 SISTEMAS DE CONTROL.....	25
4.2 REDES DE COMUNICACIÓN.....	26
4.2.1 TOPOLOGÍA DE RED.	26
4.2.2 MEDIO FÍSICO.	27
4.2.3 MÉTODO DE ARBITRAJE.....	27
4.2.4 MODO DE FUNCIONAMIENTO.	27
4.2.5 TAMAÑO Y ESTRUCTURA DE LOS MENSAJES.....	27
4.2.6 DISTANCIA DE TRANSMISIÓN.....	27
4.2.7 VELOCIDAD DE TRANSMISIÓN.	28
4.3 BUSES DE CAMPO.....	28
4.3.1 EtherCAT.....	28
4.3.1.1 PRINCIPIO DE FUNCIONAMIENTO.....	28
4.3.1.2 SINCRONIZACIÓN DEL PROCESO.....	29
4.3.1.3 SISTEMAS DE REDUNDANCIA.....	30
4.3.1.4 XFC.....	30
4.3.1.5 EtherCAT SEGURIDAD.....	30

4.4 ESQUEMAS SOLUCIÓN SISTEMA DE CONTROL.....	31
4.4.1 MODELO CENTRALIZADO.....	32
4.4.2 MODELO DISTRIBUIDO.....	33
4.5 EQUIPOS NECESARIOS.....	35
4.5.1 CLIMATIZACIÓN.....	35
4.5.2 ILUMINACIÓN.....	36
4.5.3 SEGURIDAD.....	37
4.5.4 ANALIZADOR DE REDES.....	37
4.5.5 HMI/SCADA.....	38
4.5.6 IOT.....	39
4.6 EQUIPOS.....	40
4.6.1 PC.....	40
4.6.2 ENTRADAS Y SALIDAS.....	40
4.6.3 ANALIZADOR DE REDES.....	40
4.6.4 OTROS.....	41
4.7 DISPOSICIÓN DE LOS EQUIPOS.....	41
5. RESULTADOS FINALES.....	43
5.1 DUT.....	44
5.1.1ConfiguracionHoraria.....	44
5.2 FUNCIONES GENERALES.....	44
5.2.1 FB Horario.....	44
5.2.2 FB_DiaMes.....	47
5.2.3 FB_DiaSemana.....	47
5.2.4 FB_Hora.....	48
5.2.5 FB_EnMes.....	48
5.3 CLIMATIZACIÓN.....	49
5.3.1 FB CLIMATIZACION.....	49
5.3.2 FB_CONFIGURACIONHORARIA.....	56
5.3.3 FB_PARAM_PID.....	57
5.3.4 FB_EXTRACTOR_FALLOTERMICO.....	57
5.4 ILUMINACIÓN.....	58
5.4.1 GRUPO 1.....	58
5.4.2 GRUPO 2.....	59
5.4.3 GRUPO 3.....	60
5.4.4 GRUPO 4.....	61

5.4.5 PERSIANAS.....	62
5.4.6 PROGRAMACIÓN HORARIA.	64
5.5 SEGURIDAD.	65
5.6 IoT.....	66
5.7 MAIN.....	68
6. VISUALIZACIONES.....	73
6.1 PANTALLA PRINCIPAL DEL SISTEMA.....	73
6.2 PANTALLA PRINCIPAL DE LAS PLANTAS.....	73
6.3 CLIMATIZACIÓN.....	74
6.4 ILUMINACIÓN.....	77
6.5 SEGURIDAD.....	80
7. ESTUDIO DE VIABILIDAD ECONÓMICA.....	81
8. RESUMEN DEL PRESUPUESTO.....	82
9.REFERENCIAS.....	83
ANEXOS.....	86
PRESUPUESTO.....	154
PLIEGO DE CONDICIONES.....	158

1. OBJETIVO DEL PROYECTO.

El objetivo del proyecto es dar una visión general de como implementaríamos un sistema de edificio inteligente. El proyecto de desarrollará concretamente en base a las instalaciones del centro de salud de Vinaroz.

Este proyecto se desarrollará a partir de un proyecto anterior elaborado por la empresa DISINEL S.L. dónde se implementó un sistema de climatización basado en quipos PID.

El propósito del proyecto es cambiar el modo de automatización del sistema de climatización e implementar la automatización de otros sectores del centro como son la iluminación y los dispositivos de seguridad.

Por último, se pretende introducir la tecnología IoT para el procesado de datos y así poder visualizar cómo evoluciona la eficiencia energética del centro. Los datos para evaluar la eficiencia energética se proporcionarán a partir de unos analizadores de redes instalados en los cuadros de control.

2. ALCANCE DEL PROYECTO.

El proyecto abordara una pequeña introducción de los denominados “Edificios Inteligentes” y la creación del sistema de control para las zonas de climatización, iluminación y elementos de seguridad del Centro de Salud de Vinaroz. Este sistema consistirá en la distribución de todos los elementos de control sobre plano, además del código PLC para la supervisión de todo el proceso.

Así mismo se realizará una pequeña implementación sobre IoT y la manera de, trabajar con estas plataformas para la captación de datos, en este caso en concreto, aspectos de eficiencia energética.

3. ESTUDIOS PREVIOS.

3.1. IoT.

3.1.1 DEFINICIÓN.

No hay una definición concreta para el concepto de “internet de las cosas”. Podemos encontrar varias definiciones, todas ellas correctas, pero con algunas diferencias entre ellas. Algunas definiciones especifican el concepto de protocolo de internet (IP), mientras que en otras no lo hacen. Veamos algunas de éstas.

En el artículo del Consejo de Arquitecturas de Internet (IAB)¹, *Architectural Considerations in Smart Object Networking*, tenemos la siguiente definición:

“El término “Internet de las Cosas” (IoT) denota una tendencia en que un gran número de dispositivos embebidos utilizan los servicios de comunicación que ofrecen los protocolos de Internet. A estos dispositivos suelen llamarles “objetos inteligentes” y no son operados directamente por un ser humano, sino que existen como componentes en edificios o vehículos o están distribuidos en el entorno.”

Cómo podemos observar en esta definición hay una vinculación con los protocolos de internet.

Por otra parte, tenemos la definición que propuso la ITU-T Y.2060, Unión Internacional de Telecomunicaciones (UIT) en 2012, en el artículo *Overview of the Internet of things*², dónde no encontramos referencias a estos protocolos.

“3.2.2 Internet de las Cosas (IoT): Infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperatividad de tecnologías de la información y la comunicación presentes y futuras.

Nota 1 — Gracias a la identificación, la adquisición y el procesamiento de datos y a las capacidades de comunicación, la IoT hace pleno uso de los objetos para ofrecer servicios a todo tipo de aplicaciones, garantizando a su vez el cumplimiento íntegro de los requisitos de seguridad y privacidad.

Nota 2 — Desde una perspectiva más amplia, la IoT puede considerarse una noción con repercusiones tecnológicas y sociales.”

Para el uso que vamos a hacer en este trabajo, el término IoT se refiere a la utilización de componentes electrónicos para la recogida de datos y su transmisión por medio de protocolos de internet para su posterior análisis y así poder hacer un control de los actuadores de la forma más eficiente.

3.1.2 APLICACIONES.

Hay muchos ámbitos en los que nos podemos encontrar modelos de trabajo basados en IoT como pueden ser los siguientes:

- Dispositivos unidos al cuerpo humano con los que podemos monitorizar y mantener la salud y el bienestar de las personas.
- Controladores y sistemas de seguridad integrados en el hogar para obtener una mayor comodidad en la estancia.
- Sistemas de auto pago o sistemas de gestión de inventario a distancia en tiendas, bancos, restaurantes...
- Gestión de la energía en las oficinas para mejorar la productividad de estos servicios.
- Optimización de los equipos en las fábricas para así reducir su uso y reflejarlo en variables económicas.
- En vehículos podemos encontrar sistemas de autoconducción vía GPS o sistemas para recibir información del estado de las carreteras de tu trayecto a tiempo real mientras conduces tu vehículo.
- En ciudades, podemos encontrar cómo los sistemas de iluminación se están optimizando para reducir su impacto medioambiental a través de estos sistemas IoT.

3.1.3 MODELOS DE COMUNICACIÓN.

Para el uso de estos sistemas tenemos que definir los tipos de modelos de comunicación que vamos a utilizar ya que, dependiendo del modelo, luego usaremos unos protocolos de comunicación u otros. Tenemos varios modelos de comunicación entre dispositivos:

- Dispositivo a Dispositivo.
Este modelo (*Imagen 1*) representa la conexión de dos o más dispositivos entre sí y se comunican directamente sin la necesidad de un servidor intermediario. Estos dispositivos se pueden comunicar sobre muchos tipos de redes, como puede ser un arad de tipo IP. Lo más habitual es que utilicen comunicaciones directas con protocolos bluetooth o similares. Este tipo de comunicación es muy utilizada en la automatización de hogar ya que los paquetes de datos que se envían y reciben son de un tamaño pequeño.



Imagen 1. Esquema modelo de comunicación Dispositivo a Dispositivo

- Dispositivo a la Nube.

En este modelo (*Imagen 2*) el dispositivo se conecta directamente a un servicio en la nube como pueden ser los ofrecidos por Microsoft Azure o Amazon Web Server. Los protocolos de comunicación pueden ser basados en conexiones Wi-Fi o también podemos recurrir al uso de protocolos más utilizados en la industria como MQTT³. Con este modelo podemos acceder a los datos en todo momento incluso algunas de estos servicios permiten configurar entornos de visualización para la mejor gestión de los datos almacenados.

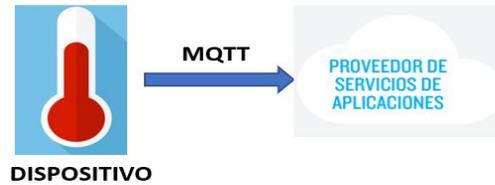


Imagen 2. Esquema modelo de comunicación Dispositivo a la Nube

- Dispositivo a puerta de enlace.

Este modelo (*Imagen 3*) se basa en la existencia de un dispositivo el cuál actúa como intermediario de la comunicación y provee de seguridad y otras funcionalidades como la traducción de protocolos o datos al sistema de comunicación.



Imagen 3. Esquema modelo de comunicación Dispositivo a puerta enlace. FUENTE: Tschofenig, H., et al., Architectural Considerations in Smart Object Networking. Tech. no. RFC 7452. IAB, marzo de 2015. Web. <https://www.rfc-editor.org/rfc/rfc7452.txt>.

Estos serían los modelos básicos de comunicación para poder realizar un sistema IoT. Hay muchos más, pero son utilizados en aplicaciones complejas con características muy concretas y que deben tener una seguridad muy alta para la transmisión de datos.

3.2 SMART BUILDINGS.

3.2.1 DEFINICIÓN.

Podemos decir que un edificio inteligente es aquel que con el uso de sistemas e instalaciones permiten su gestión y control de manera automatizada. El objetivo básico de estos edificios es reducir el consumo global del mismo para obtener una eficiencia energética mayor. Otros objetivos pueden ser la comodidad del usuario, una mayor sostenibilidad, la seguridad...

Los edificios inteligentes se han convertido en uno de los factores más importantes en las grandes ciudades, y que como podemos observar en la *Imagen 4*, dichos edificios son uno de los principales sectores en el uso de la energía eléctrica, por encima de las industrias y el sector del transporte.

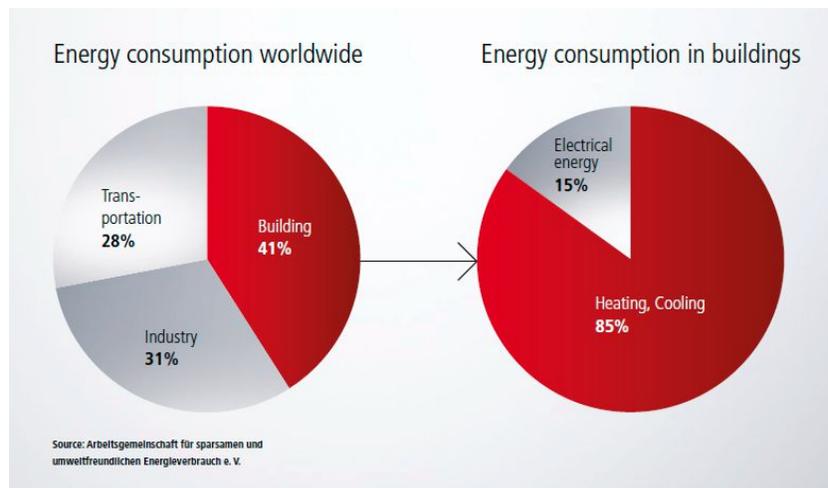


Imagen 4. Fuente: Revista internacional PC Control BECKHOFF.

Por lo tanto, la transformación del sistema residencial actual hacia uno más sostenible y eficiente, pasa por la implementación en nuestras casas de sistemas que las conviertan en edificios inteligentes.

3.2.2 ELEMENTOS NECESARIOS.

¿Qué elementos necesitamos para considerar que nuestros edificios son inteligentes?

Primeramente, debemos tener en cuenta que todos los elementos de nuestro hogar deben de estar interconectados entre ellos, las bombas de agua, las luces, alarmas de incendios... La habilidad de poder intercomunicarse es lo que hace que podamos denominar a un edificio inteligente.

Los elementos que se encargan de realizar la captación de los datos para poder comunicarlos serían los sensores, los cuales son una de las partes más importantes de nuestro proceso, ya que sin ellos no tendríamos la información necesaria para evaluar los equipos y controlarlos adecuadamente.

Seguidamente esos datos son llevados al sistema central de control el cual es capaz de procesarlos y así comunicarse con los equipos para que actúen modificando su estado.

Por último, es importante almacenar los datos que se ha proporcionado al sistema central de control para su posterior procesamiento, para realizar la detección de posibles fallos y evaluar cómo se comporta nuestro edificio.

3.2.3 BENEFICIOS DE LOS EDIFICIOS INTELIGENTES.

¿Cuáles son los beneficios de los edificios inteligentes?

Uno de los más importantes es la productividad que se puede llevar a cabo de las instalaciones. La calidad del aire, la seguridad, la iluminación son llevados a los niveles óptimos de producción.

Esto produce otro beneficio colateral, que es la eficiencia energética. Si estamos trabajando en los puntos óptimos de los equipos, el gasto energético es el menor posible.

Otro beneficio relacionado con lo mencionado anteriormente es el ahorro económico. Al haber reducido al mínimo el consumo energético de cada equipo, el gasto económico por su uso también se reduce al mínimo.

Algún ejemplo de la reducción del consumo aplicado en edificios reales podría ser el del hospital Infanta Sofía descrito por Francisco Conesa Cervantes y José Angel Peinador Aguilar en el artículo, " *¿Machine Learning o Smart Humans? Aplicado a monitorización y gestión energética de la demanda en edificios.* "

En el gráfico posterior (*Imagen 5*) podemos ver como el consumo eléctrico ha bajado respecto al consumo de referencia.

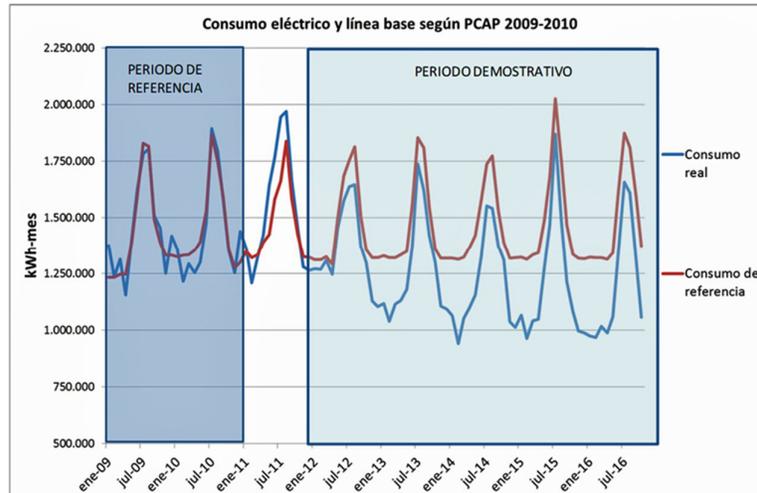


Imagen 5. Gráfico de la evolución del consumo eléctrico del hospital. Fuente: (Artículo ¿Machine Learning o Smart Humans? Aplicado a monitorización y gestión energética de la demanda en edificios (Francisco Conesa Cervantes, José Angel Peinador Aguilar, 19/10/2017)).

En este proyecto, el objetivo es dar una visión global de una implementación de un edificio inteligente en el centro de salud de Vinaroz y a su vez utilizar un sistema IoT para el tratamiento de los datos, como para poder comunicarnos con nuestro edificio.

4. ANALISIS DE SOLUCIONES.

Primero tenemos que mencionar que para efectuar el proyecto hemos decidido utilizar el hardware de BECKHOFF para la automatización del edificio, ya que es el producto que manejamos en la empresa para realizar este tipo de proyectos. La empresa decidió apostar por BECKHOFF ya que sus productos se desmarcan de la competencia como OMRON o SCHNEIDER, ya que presentan un nuevo concepto en este mundo, como es la automatización con PC. Sus sistemas de control se caracterizan por estar compuestos por PCs los cuales les dan más versatilidad a los proyectos que los PLC convencionales. Por este motivo, el proyecto se desarrolla basado en la automatización con PC. Asimismo, toda la introducción que viene a continuación tendrá referencia a la posterior aplicación en la estructura de proyecto que realizaremos con dichos productos.

4.1 SISTEMAS DE CONTROL.

Hay dos tipos de control, el control centralizado y el control descentralizado.

Se denomina sistema de control centralizado (*Imagen 6*) a aquel que sólo tiene un equipo de control y a partir de este se controlan todos los procesos del sistema. Los sensores y actuadores están conectados a este sistema y es el equipo de control el que se encarga de distribuir la información adecuadamente. Para que todo funcione correctamente los equipos deben estar conectados entre sí mediante una red de comunicación.

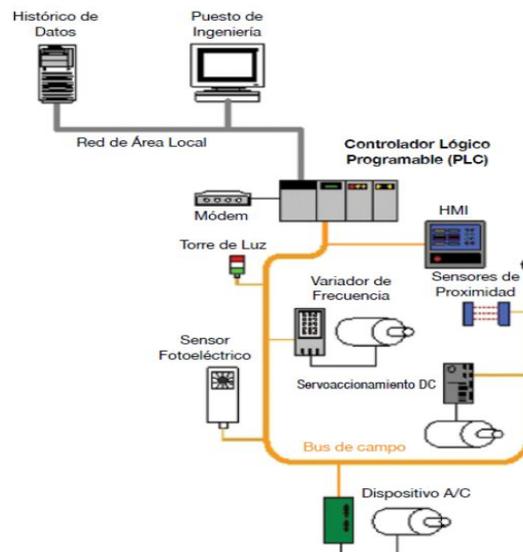


Imagen 6. Esquema de un sistema de control centralizado. Fuente: <https://blogs.deusto.es/master-informatica/tipos-de-sistemas-de-control-industrial/>

Por otra parte, tenemos los sistemas de control distribuidos o descentralizados (*Imagen 7*). Son aquellos en los que hay varios equipos de control, cada uno de los cuales controla una parte del proceso global. En este caso los sensores y actuadores de cada parte están

4.2.2 MEDIO FÍSICO.

El medio físico es el medio por donde se transmiten los datos en una red de comunicaciones. Las comunicaciones pueden ser por cable o inalámbrica.

La conexión inalámbrica es útil cuando las distancias son especialmente largas, aunque perdemos mucha velocidad de transmisión.

Respecto a la comunicación por cable es la más utilizada y se pueden encontrar varios tipos, como pueden ser, par trenzado sin apantallar, par trenzado apantallado, cable coaxial o fibra óptica.

4.2.3 MÉTODO DE ARBITRAJE.

Se denomina método de arbitraje, al método por el cual se distribuye entre los nodos de la red el derecho a utilizar ésta para transmitir un mensaje.

Algunos de los métodos más utilizados serían:

- Maestro Único.
- Pase de Testigo.
- CSMA/CD (Carrier Sense Multiple Acces/Collision Detection).
- CSMA/BA (Carrier Sense Multiple Acces/Bitwise Arbitration).

4.2.4 MODO DE FUNCIONAMIENTO.

Respecto al modo de funcionamiento, cada red de comunicaciones comercial utiliza un modo. Algunos de los modos serían:

- Maestro-Esclavo con maestro único
- Maestro-Esclavo con varios maestros
- Punto a Punto
- Productor/Consumidor
- Cliente/Servidor

4.2.5 TAMAÑO Y ESTRUCTURA DE LOS MENSAJES.

El tamaño de los mensajes es el número de bits que contienen.

Respecto a la estructura de los mensajes es la que determina el significado de cada bit que se transmite.

El tamaño y estructura de los mensajes junto al método de arbitraje y el modo de funcionamiento definen lo que se conoce como protocolo de comunicación.

4.2.6 DISTANCIA DE TRANSMISIÓN.

La distancia de transmisión depende únicamente de las características eléctricas de la red.

4.2.7 VELOCIDAD DE TRANSMISIÓN.

La velocidad es la propiedad que mide cuantos bits se pueden enviar por segundo a través de la red. Las velocidades de redes comerciales pueden variar entre unos 10 Kb/s hasta unos 10 Mb/s.

4.3 BUSES DE CAMPO.

Un bus de campo es un sistema de transmisión de información que simplifica enormemente la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción.

En la actualidad existen infinidad de buses de campo como podrían ser CAN, ASI, DeviceNet, Profibus, Modbus...

Cada uno de estos tiene unas características que se adecuan mejor a un tipo de aplicaciones.

Podemos destacar el uso del bus de campo llamado DALI (Digital Addressable Lighting Interface). Se trata de un bus para la comunicación digital entre los equipos básicamente de luminotecnía. Podríamos gastar este protocolo para hacer la implementación de la parte de iluminación, pero hemos preferido integrarlo todo en el bus EtherCAT.

Nosotros nos centraremos en el bus de campo EtherCAT para la comunicación entre las diferentes tarjetas y los PC de control ya que los equipos BECKHOFF tienen optimizado su uso y así aumentamos al máximo el rendimiento total de la instalación. Aunque para la comunicación de los equipos entre ellos utilizaremos un bus de campo de EtherNET. A continuación, haremos una introducción al uso de EtherCAT como bus de campo y sus características.

4.3.1 EtherCAT.

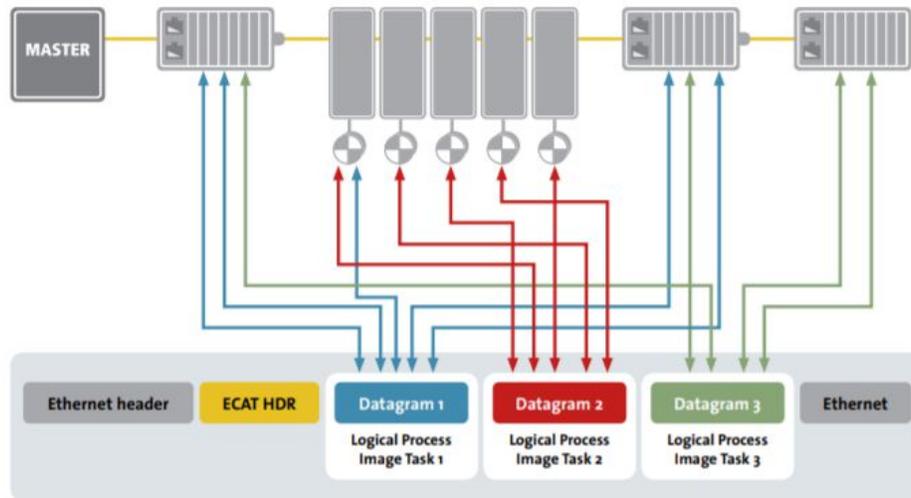
EtherCAT es una tecnología de Ethernet industrial de alto rendimiento, bajo coste y fácil de usar, que presenta una topología flexible.

4.3.1.1 PRINCIPIO DE FUNCIONAMIENTO.

El principio funcional clave de EtherCAT (*Imagen 8*) reside en cómo sus nodos procesan las tramas Ethernet: cada nodo lee los datos direccionados a él y escribe sus datos de nuevo en la trama, todo mientras la trama se mueve aguas abajo. Esto conduce a una mejor utilización del ancho de banda (una trama por ciclo suele ser suficiente para la comunicación), al tiempo que elimina la necesidad de conmutadores.

Con EtherCAT, el dispositivo maestro únicamente necesita llenar una sola trama EtherCAT con nuevos datos de salida y enviar la trama a través del Acceso Directo a Memoria (DMA, del inglés Direct Memory Access) al controlador MAC. Cuando se

recibe una trama con nuevos datos de entrada a través del controlador MAC, el dispositivo maestro puede volver a copiar la trama a través de DMA en la memoria del ordenador, todo ello sin que la CPU tenga que copiar activamente ningún dato. Además de los datos cíclicos, se pueden utilizar otros datagramas para comunicaciones asíncronas o impulsadas por eventos.



Inserción de datos de proceso sobre la marcha

Imagen 8. Principio de funcionamiento del protocolo de comunicación de EtherCAT. Fuente: Revista EtherCAT Technology Group. Folleto EtherCAT- el bus de campo Ethernet.

4.3.1.2 SINCRONIZACIÓN DEL PROCESO.

En aplicaciones con procesos distribuidos espacialmente que requieren acciones simultáneas, la sincronización exacta es particularmente importante. Este es el caso por ejemplo en aplicaciones en las que varios servoejes ejecutan movimientos coordinados.

Los módulos esclavos de EtherCAT disponen de un reloj de 64 bits con resolución de 1 nSeg.

Para realizar el sincronismo (*Imagen 9*), primero cada esclavo pone en marcha el cronómetro cuando llega el telegrama y lo para cuando vuelve de regreso. De esta forma cada esclavo mide su retraso hasta el “final de recorrido”. Así el maestro obtiene el tiempo de cada esclavo y puede calcular el retraso entre los mismos. Con este procedimiento con EtherCAT podemos sincronizar todos los relojes de los esclavos con un margen de nanosegundos.

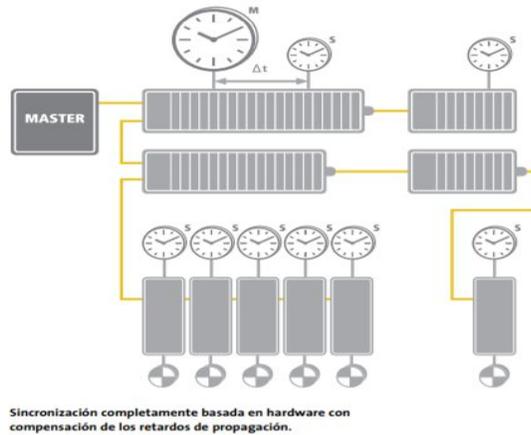


Imagen 9. Esquema sincronización de equipos con EtherCAT. Fuente: Revista EtherCAT Technology Group. Folleto EtherCAT- el bus de campo Ethernet.

4.3.1.3 SISTEMAS DE REDUNDANCIA.

EtherCAT ofrece una solución para aplicaciones que requieran de redundancia de cable (*Imagen 10*). No se requiere un doble canal en cada esclavo, cosa que encarece cada nodo, como sería el caso de ControlNet, por ejemplo. Sino que sería suficiente con un puerto EtherNet adicional y un software.

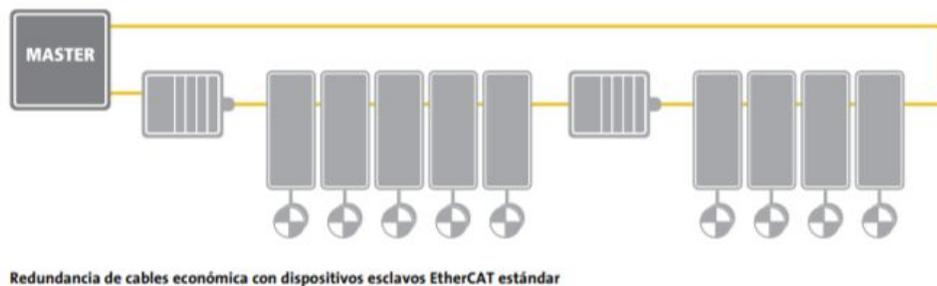


Imagen 10. Sistemas de redundancia con EtherCAT. Fuente: Revista EtherCAT Technology Group. Folleto EtherCAT- el bus de campo Ethernet.

4.3.1.4 XFC.

XFC es una tecnología que permite capturar entradas y activar salidas de forma ultra rápida, muestrear señales analógicas a frecuencias de hasta 1000 KHz y controlar servodrivs con una fluctuación del retardo o jitter <math>< 1 \mu \text{Seg}</math>. Para ello se dispone de una serie de módulos específicos que incorporan reloj, time-stamping (estampación del tiempo) y oversampling (sobre-muestreo) además de E/S rápidas.

4.3.1.5 EtherCAT SEGURIDAD.

La tecnología de seguridad EtherCAT ha sido desarrollada según el estándar IEC 61508, cuenta con la certificación TÜV y está estandarizada en IEC 61784-3. El protocolo es adecuado para aplicaciones de seguridad con un nivel de integridad de seguridad de hasta SIL 3.

EtherCAT ofrece una solución totalmente integrada. Reduce el número de buses e interfaces y centraliza la configuración el diagnóstico y el mantenimiento de E/S seguras y no seguras en una única herramienta.

El principio de funcionamiento se basa en unas tramas de seguridad conocidas como Safety Containers, las cuales contienen los datos críticos para la seguridad e información adicional para proteger esos datos.

Teniendo claro todos estos conceptos sobre como diseñar un sistema de control ya podemos analizar y dar una solución para nuestro proyecto.

4.4 ESQUEMAS SOLUCIÓN SISTEMA DE CONTROL.

Primero se han considerado algunas restricciones, ya que, al no ser un proyecto de nueva creación, hay que ajustar la programación y la selección de los equipos a instalar para la automatización con los equipos existentes en las instalaciones. Por ejemplo, para la climatización se utilizarán los climatizadores ya instalados.

También hay que mencionar que no se han realizado cambios en el sistema de iluminación, se ha adaptado el proyecto a la distribución existente.

Para hacer la automatización se han considerado dos configuraciones:

- La primera consiste en un sistema de control centralizado, con un PC central dónde se haría la visualización, el control y la utilización de la plataforma IIoT. A su vez, desde este PC central se distribuiría la periferia para controlar los distintos módulos de cada planta.
- Respecto a la segunda configuración, consistiría en un control distribuido, dividiendo el centro de salud en zonas. En cada zona se dispondría de un PC con menos potencia para el control de cada una de ellas. En la zona de control dispuesta en la planta baja, se instalaría uno de mayor potencia para así poder controlar los otros PC de las distintas plantas.

A continuación, mostraremos los esquemas de estas configuraciones con sus características.

4.4.1 MODELO CENTRALIZADO.

Las ventajas de esta configuración son:

- Al tener un equipo de control, abarata el coste del proyecto.
- Sólo es necesaria la instalación de software a un PC, esto reduce el tiempo de mano de obra de los técnicos.

Los inconvenientes de esta configuración son:

- Si el equipo tuviera algún tipo de problema la reparación supondría la parada de todo el sistema de control.
- Otro problema derivado es que al ser un equipo potente la disponibilidad de recambios es más escasa ya que el coste de este producto es mayor y su período de fabricación podría ser más extenso respecto a los productos menos potentes.

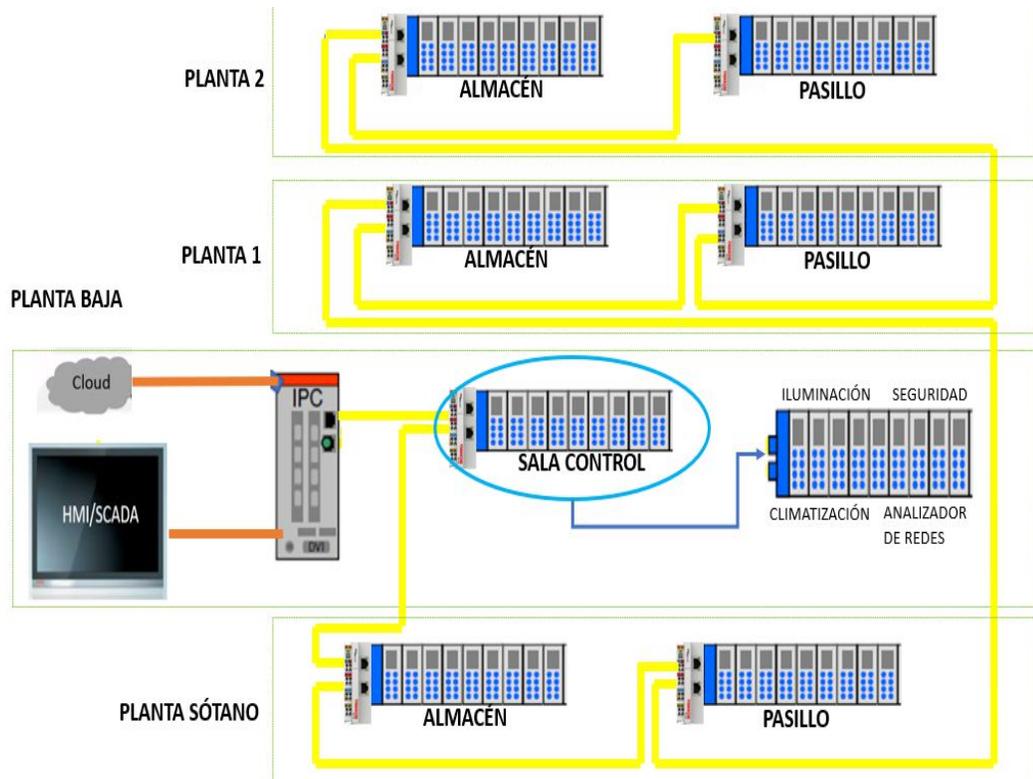


Imagen 10. Esquema modelo centralizado del sistema de control del centro de salud de Vinaroz.

4.4.2 MODELO DISTRIBUIDO.

Las ventajas de esta configuración son:

- Si tuviéramos algún problema con el equipo al estar distribuido en varias zonas, solo tendríamos que parar esa zona para la reparación del equipo y las demás podrían seguir funcionando sin ningún tipo de problema al ser independientes. También es más fácil la detección de problemas.
- Como los equipos que utilizamos son menos potentes el coste de cada equipo es menor que uno de mayor potencia, esto nos permitiría tener un equipo de repuesto por si ocurriera una incidencia poderlo cambiar rápidamente.

Los inconvenientes de esta configuración son:

- La instalación en conjunto tiene un mayor coste ya que necesitas más equipos para controlar toda la infraestructura.
- Para hacer la instalación de código habría que hacerlo para cada equipo y en cada equipo el código adecuado.

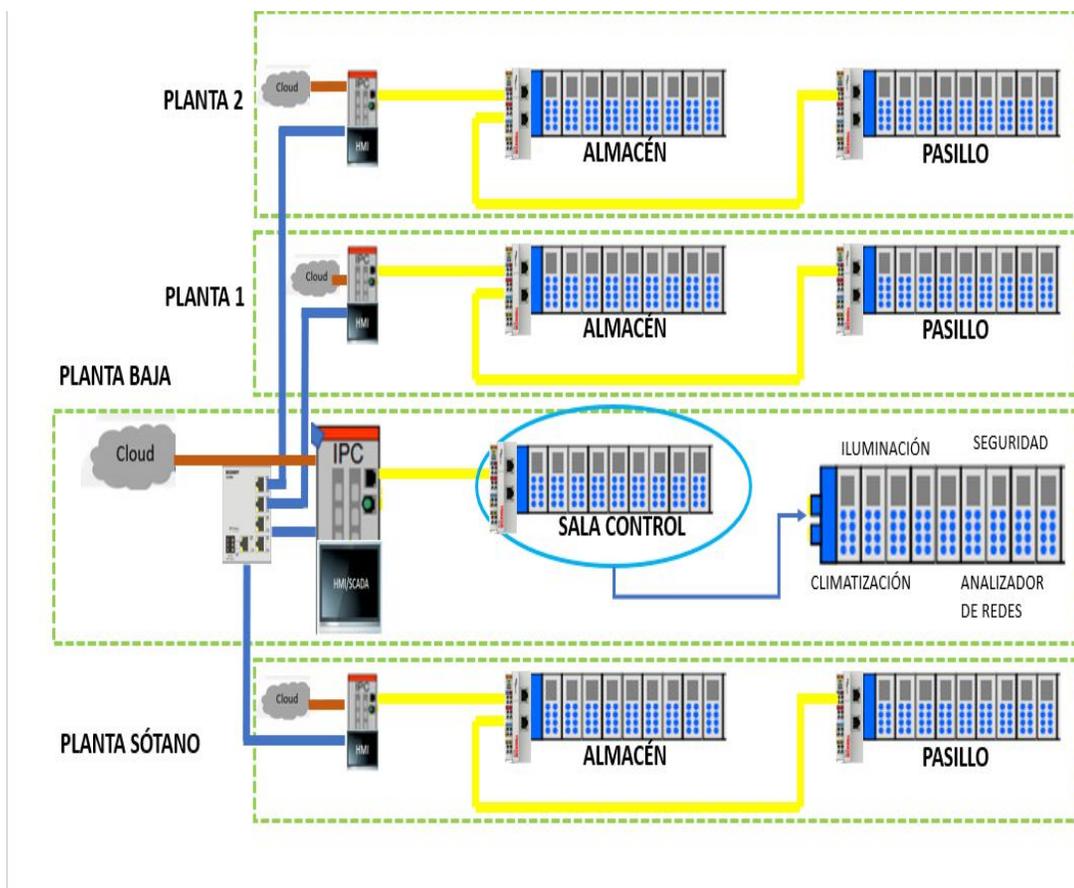


Imagen 11. Esquema modelo distribuido del sistema de control del centro de salud de Vinaroz.

Las particularidades que tendríamos utilizando los equipos de BECKHOFF se basarían principalmente en que BECKHOFF trabaja con PC que al añadirles el runtime de TwinCAT, se comportan como PC+PLC. Esto quiere decir que aparte de poder ejecutar programas de PLC también podemos hacer uso de programas de Windows compatibles.

Tanto en el caso del modelo centralizado como en el modelo distribuido el uso de los equipos BECKHOFF nos ahorraría la compra de un PC y PLC, ya que en el mismo equipo lo podríamos tener todo.

Se ha decidido implantar un sistema de control distribuido por varias razones:

- El hecho de que este tipo de sistema de control es más común en el mundo de la automatización
- La aparición de algún error en cualquier planta o equipo no suponga el cierre del centro de salud. Ya que con el modelo centralizado como se ha comentado, si el equipo falla, se para todo el sistema hasta la reparación de este.
- Buscamos una robustez en el sistema para poderlo tener siempre operativo y que los tiempos de sustitución sean los menores posibles. Eso lo conseguimos con el sistema de control distribuido.

Respecto a la topología de red de comunicación que se ha implantado, se pensó en dos soluciones, una topología tipo bus o una tipo árbol con un modo de funcionamiento maestro-esclavo con un único maestro.

La topología implantada ha sido de tipo árbol ya que se tiene una rama principal y de esta salen varias ramas para cada planta donde se conectan los equipos de control de cada planta.

Respecto a los buses de campo se utilizarán EtherCAT y EtherNET como hemos comentado anteriormente.

Por lo tanto, se obtiene un sistema de control distribuido con una tipología de tipo árbol con un modo de funcionamiento Maestro-Esclavo con maestro único y con buses de campo EtherCAT y EtherNET.

Teniendo claro la solución a implantar para el control de todo el sistema, comentaremos que equipos se necesitan en cada sector a controlar.

4.5 EQUIPOS NECESARIOS.

4.5.1 CLIMATIZACIÓN.

En el caso del sector de climatización se han utilizado los climatizadores ya instalados en el centro de salud. Para hacer el control de esta zona se ha estudiado el manual de uso del anterior PLC de control del CLIMATIZADOR_M90 (Anexo II), dónde se detallaban todas las funcionalidades de los climatizadores.

Para controlar este sector BECKHOFF dispone de unas librerías en las cuales ya vienen incorporadas algunas funciones de PID para el control de la temperatura de los climatizadores.

No obstante, se ha decidido utilizar los PID de las librerías básicas ya que ofrecías una mayor flexibilidad a la hora de tener el control sobre el PID.

Las entradas y salidas necesarias en cada planta para el control se detallan a continuación en las Tabla 1 y Tabla 2.

ENT.DIG (DI)	SAL.DIG (DO)	SONDAS TEMP (AI)
FC1_EnMarcha	FC1 VM_ABRIR	AI0 FC1 STempReg
FC2_EnMarcha	FC1 VM_CERRAR	AI1 FC2 STempReg
FC3_EnMarcha	FC1 MARCHA	AI2 FC3 STempReg
FC4_EnMarcha	FC2 VM_ABRIR	AI3 FC4 STempReg
Extractor EnMarcha	FC2 VM_CERRAR	AI5 STemp INV/VER
Fallo Térmicos	FC2 MARCHA	
	FC3 VM_ABRIR	
	FC3 VM_CERRAR	
	FC3 MARCHA	
	FC4 VM_ABRIR	
	FC4 VM_CERRAR	
	FC4 MARCHA	
	EXTRACTOR MARCHA	
	CAMBIO INVIERNO/VERANO	

Tabla 1: Entradas y Salidas máximas del sector de Climatización por planta.

	ENT.DIG (DI)	SAL.DIG (DO)	SONDAS TEMP (AI)
PLANTA 2	5	11	4
PLANTA 1	5	11	4
PLANTA BAJA	4	8	3
SEMI-SÓTANO	5	11	4
TOTAL SECTOR	19	41	15

Tabla 2. Entradas y Salidas necesarias por planta. Total del sector de climatización.

4.5.2 ILUMINACIÓN.

En la iluminación se han definido dos tipos de sensores que actúan sobre las luminarias y dos actuadores.

- Sensores de Presencia (Señal digital)
- Sensores de Luminosidad (Señal analógica)
- Luminarias (Señal Digital)
- Persianas Motorizadas (Señal analógica)

Los sensores no se incluyen en el proyecto, la búsqueda y compra de los mismos corre por parte del usuario, al no disponer de este tipo de sensores en la empresa.

También se han definido 4 grupos de iluminación para dividir las diferentes zonas de las plantas con su sistema de iluminación correspondiente y un sistema de persianas motorizadas para controlar la luz solar a introducir en las estancias.

- Grupo 0: Estancias con Iluminación autónoma. No incluida en el sistema de control
- Grupo 1: Estancias con Iluminación sin sensores de luminosidad y sin sensores de presencia.
- Grupo 2: Estancias con Iluminación sin sensores de luminosidad y con sensores de presencia.
- Grupo 3: Estancias con Iluminación con sensores de luminosidad y sin sensores de presencia.
- Grupo 4: Estancias con iluminación con sensores de luminosidad y con sensores de presencia.
- Persianas: Son grupos de láminas que se controla el nivel de apertura mediante un sistema motorizado.

En la siguiente tabla (Tabla 3) se detallan las entradas y salidas correspondientes a cada planta.

	LUMINARIAS DO	S. PRESENCIA DI	SENSORES LUZ AI	PERSIANAS AO
PLANTA 2	13	8	18	5
PLANTA 1	17	10	16	5
PLANTA BAJA	18	10	15	3
SEMI- SÓTANO	16	9	14	3
TOTAL SECTOR	64	37	63	16

Tabla 3. Entradas y Salidas necesarias por planta. Total del sector de iluminación.

4.5.3 SEGURIDAD.

La seguridad del edificio estará compuesta por detectores de humo, pulsadores de alarma, detectores de presencia y unas sirenas. Los detectores y los pulsadores son entradas digitales, mientras que las sirenas son salidas digitales.

En la siguiente tabla (Tabla 4) se detallarán las entradas y salidas necesarias en cada planta.

	ENT.DIG (DI)	SAL.DIG (DO)
PLANTA 2	12	1
PLANTA 1	12	1
PLANTA BAJA	11	1
SEMI-SÓTANO	10	1
TOTAL SECTOR	45	4

Tabla 4. Entradas y Salidas necesarias por planta. Total del sector de seguridad

4.5.4 ANALIZADOR DE REDES.

BECKHOFF ofrece una variedad de analizadores de redes para poder obtener datos para la evaluación de la eficiencia de nuestros equipos y en total la de todo el edificio. Podemos obtener valor de presión de agua que envía la bomba al circuito de saneamiento, presiones de aire del sistema de ventilación, temperatura del ambiente dentro del edificio, de la corriente en las fases eléctricas...

En este proyecto se dará una visión de cómo incorporar unas tarjetas analizadoras de red, con las que se puede obtener datos como los de voltajes, corrientes y potencias eléctricas que tienen nuestros equipos.

A continuación, veremos que analizadores nos ofrece BEKHOFF y cuál es el óptimo para nuestro fin.

Para la toma de decisión de la tarjeta de red óptima para nuestra aplicación se ha realizado un sencillo método. El elegido es el denominado método de las jerarquías analíticas. Se ha optado por éste ya que es simple y fácil de entender con el que obtenemos buenos resultados a la hora de la toma de decisiones.

Seguidamente podemos observar la tabla final (Tabla 5) obtenida por este método donde se constata que las mejores tarjetas para nuestro uso son la EL3423 o la EL3443. Nos decidimos a coger la EL3423 por el hecho de que las características que nos ofrece son suficientes para el uso previsto y su valor es inferior al de la otra tarjeta.

4- DECISIÓN			
	PRECIO	CARACTERÍSTICAS	TOTAL
EL3423	0,33	0,33	0,67
EL3443	0,17	0,50	0,67
EL3453	0,00	0,17	0,17
EL3483	0,50	0,00	0,50

Tabla 5. Tabla final del método de decisión del analizador de redes.

Cómo no se dispone de la tarjeta de red para hacer las pruebas pertinentes, en el código del programa posterior solo se han creado las variables necesarias para luego poder acceder a las mediciones de las tarjetas. De todos modos, dejaremos un pequeño comentario a continuación para que quede claro como configurar estas tarjetas.

Primero se instalan las tarjetas. Cuando ya se han montado las tarjetas, en la parte de código, se tendrán que escanear los dispositivos para tener sus valores de entrada, luego estos valores de entrada los relacionamos con las variables globales del sistema creadas y así ya se pueden leer todos los datos que envía el analizador de red.

4.5.5 HMI/SCADA.

Para la visualización y control de los parámetros del sistema se han pensado dos opciones para su implementación, un sistema HMI o un SCADA.

“Un sistema SCADA (Supervisory Control And Data Acquisition) es aquel que permite que un ordenador se comuniquen con los equipos que controlan un proceso, usualmente PLC, y así un operario pueda supervisar desde el ordenador el comportamiento de todo el proceso.”

Esta definición se encuentra en el libro Automatización Industrial⁴

Este sistema se podría implementar con el hardware de Unitronics.

Respecto al sistema HMI (*Human Machine Interface*) es muy parecido al sistema SCADA, la finalidad de este sistema es el mismo, dar una visión del comportamiento del proceso al operario desde un ordenador y poder modificar parámetros sobre los equipos estando en el equipo de supervisión.

En este proyecto, por dar una solución integrada en BECKHOFF, se han utilizado las librerías desarrolladas para el uso del HMI. Asimismo, se crearán unas visualizaciones para replicar el comportamiento del proyecto propuesto. Estas visualizaciones son las que se instalarán en los PC de cada planta.

4.5.6 IOT.

Como todo el proyecto va a estar desarrollado sobre los productos de BECKHOFF se ha investigado sobre las soluciones que ofrece BECKHOFF para gestionar el IoT.

BECKHOFF ofrece una solución integral para la integración de plataformas IOT. Estas soluciones se componen tanto de software como de hardware.

Respecto al hardware, BECKHOFF ha desarrollado un dispositivo, en concreto el EK9160 (IoT Bus Coupler), que sin la necesidad de un programa de control es capaz de subir datos a la nube. El control de este dispositivo se ejecuta desde la web dónde tenemos el servicio de IoT. Este equipo de control está certificado para configurarse desde las plataformas de AWS y AZURE.

En la parte de software, BECKHOFF incorpora varias soluciones para poder establecer la comunicación con los servicios de las diferentes plataformas de IoT.

Tenemos los FBs que ofrece TwinCAT 3 IoT Communication (MQTT) con referencia TF6701 en la web de BECKHOFF. Esos FBs se utilizan para publicar datos desde TwinCAT y recibir desde el servidor. Este FB IotMqttClient, es el que utilizamos para comunicarnos con el servidor y solo sirve para comunicación por el protocolo de comunicación MQTT o AMQP.

Luego encontramos otro paquete de funciones como es TwinCAT 3 IoT Functions el cual nos permite tener una comunicación bidireccional con los servicios de la nube Microsoft AZURE y AMS de forma sencilla desde el PLC. Esta comunicación se puede establecer por varios protocolos de comunicación, estos son MQTT, AMQP, ADS o OPC-UA.

También encontramos la función TC3 IoT Data Agent. Esta es una aplicación que se puede instalar en un controlador o en un ordenador. Con ella podemos comunicarnos con la nube de una forma sencilla. El suplemento TF6720 TC3 IoT Data Agent permite subir los datos a la nube sin necesidad de programar nada en el programa de PLC de TwinCAT. Es un servicio que corre sobre Windows (no se ejecuta en el entorno de tiempo real) que permite recoger datos por ADS (para comunicar con equipos de Beckhoff con TwinCAT 2 o TwinCAT 3) o por OPC UA (para equipos de terceros) y los puede subir a Microsoft Azure, Amazon Web Services o a un bróker MQTT genérico.

Por último, tenemos un nuevo FB de la librería Tc3_IoTCommunicator (TF6730) con la que podemos enviar datos desde TwinCAT PLC a un dispositivo móvil. Esta aplicación es más usada para el intercambio de notificaciones, ya que desde el móvil no podemos enviar información al PLC. La aplicación móvil es gratis y se puede encontrar en los servicios de Google Play o App Store, tiene como nombre TwinCAT IoT Communicator.

Si quisiéramos hacer el tratamiento de estos datos en nuestro PC sin depender de las limitaciones que puede aportar el servicio de almacenamiento en la nube, BECKHOFF ofrece un software llamado TwinCAT Analytics con el que podemos hacer el análisis de los datos obtenidos desde nuestro PC y mostrarlos en una aplicación HMI por ejemplo. Este software ofrece una multitud de algoritmos relacionados con el tratamiento de datos de eficiencia que nos facilita la tarea de procesamiento.

Entre las distintas soluciones, se ha optado por utilizar solo software y de las distintas opciones se ha elegido el uso de la función TwinCAT 3 IoT Communication. Los datos se subirán a un servidor de Microsoft Azure que se ha creado para hacer las pruebas pertinentes de la comunicación IoT.

4.6 EQUIPOS.

Los equipos escogidos para hacer el control son:

4.6.1 PC.

Hemos optado por coger dos tipos de paneles PC, uno mas grande para la zona de control (15”) y otro más pequeño para las otras plantas (7”). Ambos paneles cuentan con 1GB de RAM DDR3, y como procesador un ARM Cortex™-A8. Las licencias de TwinCAT de ambos equipos son del tipo TC3:30.

- CP6606-0001-0020 (Control Panel 7”)
- CP6602-0000-0020 (Control Panel 15”)

4.6.2 ENTRADAS Y SALIDAS.

Respecto a las entradas y salidas:

Las elecciones de las tarjetas se han basado en la intención de estandarizar todas estas entradas y salidas y así poder disponer de repuestos para un cambio rápido, en caso de fallo de alguna de estas. Por lo tanto, en todas las plantas que se necesiten alguna de estas tarjetas serán del tipo descrito posteriormente.

Entradas digitales:

- Tarjeta EL1008 con 8 entradas digitales.

Salidas Digitales:

- Tarjeta EL2008 con 8 salidas digitales.

Entradas Analógicas:

- Para la iluminación se necesitan entradas analógicas de 4-20mA. Para este fin se ha encontrado la tarjeta EL3054 de 4 entradas analógicas.
- Para la climatización se usan entradas analógicas para termopares/termistores. Por lo tanto, se utilizarán las tarjetas EL3204/EL3314.

Salidas Analógicas:

- Tarjeta EL4024 con 4 salidas analógicas de 4-20mA.

4.6.3 ANALIZADOR DE REDES.

- EL3423.

4.6.4 OTROS.

Por último, se necesita el Bus Coupler EK1100 para poder hacer las conexiones de EtherCAT entre las diferentes tarjetas y un switch CU2005 para hacer la conexión entre los equipos por EtherNET.

También se necesitarán las tapas de final del bus EL9011 para los terminales.

4.7 DISPOSICIÓN DE LOS EQUIPOS.

En este apartado se mostrarán unas tablas resumen de las entradas y salidas referenciadas a cada planta y su ubicación. Los datos de esta tabla han sido obtenidos de las tablas de diseño del sistema de control dónde aparece la ubicación del cuadro eléctrico donde se instalarán los equipos.

DI	EL1008						
PLANTA	CLIMATIZACIÓN	ILUMINACIÓN	SEGURIDAD	TARJETAS SECTORES	ALMACEN	PASILLO	SALA DE CONTROL
SEGUNDA	5	8	12		4	0	0
PRIMERA	5	10	12		4	0	0
BAJA	5	10	11		0	0	4
SÓTANO	5	9	10		3	0	0

Tabla 6. Número de tarjetas EL1008 con su distribución en sectores.

DO	EL2008						
PLANTA	CLIMATIZACIÓN	ILUMINACIÓN	SEGURIDAD	TARJETAS SECTORES	ALMACEN	PASILLO	SALA DE CONTROL
SEGUNDA	11	13	1		3	1	0
PRIMERA	11	17	1		3	1	0
BAJA	8	18	1		0	0	4
SOTANO	11	16	1		3	1	0

Tabla 7. Número de tarjetas EL2008 con su distribución en sectores.

AI	EL3054				
PLANTA	ILUMINACIÓN	TARJETAS SECTORES	ALMACEN	PASILLO	SALA DE CONTROL
SEGUNDA	18		4	2	0
PRIMERA	16		3	2	0
BAJA	15		0	0	4
SOTANO	14		3	2	0

Tabla 8. Número de tarjetas EL3054 con su distribución en sectores.

AI	EL3204		TARJETAS SECTORES	ALMACEN	PASILLO	SALA DE CONTROL
PLANTA	CLIMATIZACIÓN	ALMACEN		PASILLO	SALA DE CONTROL	
SEGUNDA	4	1		0	0	
PRIMERA	4	1		0	0	
BAJA	3	0		0	1	
SOTANO	4	1	0	0		

Tabla 9. Número de tarjetas EL3204/3314 con su distribución en sectores.

AO	EL4024		TARJETAS SECTORES	ALMACEN	PASILLO	SALA DE CONTROL
PLANTA	ILUMINACIÓN	ALMACEN		PASILLO	SALA DE CONTROL	
SEGUNDA	5	2		0	0	
PRIMERA	5	2		0	0	
BAJA	3	0		0	1	
SOTANO	3	1	0	0		

Tabla 10. Número de tarjetas EL4024 con su distribución en sectores.

ANALIZADOR DE REDES	EL3423		TARJETAS SECTORES	ALMACEN	PASILLO	SALA DE CONTROL
PLANTA	CLIMATIZACIÓN	ILUMINACIÓN		ALMACEN	PASILLO	SALA DE CONTROL
SEGUNDA	3	2		4	1	0
PRIMERA	3	2		4	1	0
BAJA	2	2		0	0	4
SOTANO	3	2	4	1	0	

Tabla 11. Número de tarjetas EL3423 con su distribución en sectores.

EtherCAT	EK1100			TARJETAS SECTORES	ALMACEN	PASILLO	SALA DE CONTROL
PLANTA	CLIMATIZACIÓN	ILUMINACIÓN	SEGURIDAD		ALMACEN	PASILLO	SALA DE CONTROL
SEGUNDA	2				1	1	0
PRIMERA	2				1	1	0
BAJA	1				0	0	1
SOTANO	2			1	1	0	

Tabla 12. Número de tarjetas EK1100 con su distribución en sectores.

5. RESULTADOS FINALES.

Por lo tanto, para la realización del proyecto se necesitan los siguientes equipos:

- **3xCP6606**
- **1xCP6602**
- **15xEL1008**
- **16xEL2008**
- **20xEL3054**
- **4xEL3204**
- **6xEL4024**
- **19xEL3423**
- **7xEK1100**
- **1XCU2005**
- **7xEL9011**

Mencionados en el apartado anterior de la memoria.

Seguidamente se procederá a detallar la estructura de código que se ha implementado para la solución del sistema de control.

Dividiremos la información del siguiente modo.

Primero se detallará la estructura creada para su posterior uso en los códigos relacionados con la programación horaria y se explicará los códigos con funciones generales a todos los sectores. A continuación, se seguirá con la explicación de los códigos implementados en los sectores. Por último, se mostrará un ejemplo de cómo llamar a estos fragmentos de código para hacerlos funcionar en un programa principal.

Se puede observar que la programación se basa en el estándar IEC-61131. Destacando el uso de dos lenguajes de programación recogidos en esta norma, como son el texto estructurado (ST) y el Diagrama de escalera o Ladder (LD).

También hay que diferenciar entre los diferentes POU utilizados en la programación.

Tenemos varios Programas, algunos FBs y diversas Funciones.

Las diferencias entre estos tipos de POU son las siguientes:

Los POU de nivel más bajo son las Funciones las cuales soportan varios elementos de entrada, pero solo un valor de salida. Las Funciones pueden ser llamadas por Programas, FBs o por otras Funciones, pero no pueden llamar a Programas o FBs.

En el escalón intermedio de los POU tenemos a los FBs los cuales soportan varios elementos de entrada como de salida. Los FBs pueden ser llamados por Programas o por otros FBs y tienen la posibilidad de llamar a las Funciones, pero no pueden llamar a Programas.

Por último, el POU de más nivel es el Programa. Este POU es el principal y el que se ejecuta para hacer las llamadas a otros Programas, FBs o Funciones. Generalmente suele haber un POU principal que se ejecuta y va llamando a los demás POU en función de las condiciones especificadas en el código.

Para la comprobación de la programación horaria es necesario el uso de una estructura de datos la cual ha sido creada específicamente para este fin llamada ConfiguracionHoraria.

5.1 DUT.

5.1.1 ConfiguracionHoraria.

Con la creación de esta estructura se ha intentado hacer más claro el código, ya que, sin la creación de esta estructura, cada vez que necesitaríamos utilizar una programación horaria deberíamos poner todos los parámetros mostrados en el código posterior.

```
1  TYPE ConfiguracionHoraria :
2  STRUCT
3      Mes_Inicio: SINT;
4      Mes_Final: SINT;
5      Dia_Inicio: WORD;
6      Dia_Final: WORD;
7      Dias_Semana: DWORD;
8      Hora_Inicio: WORD;
9      Hora_Final: WORD;
10     Minuto_Inicio: WORD;
11     Minuto_Final: WORD;
12
13 END_STRUCT
14 END_TYPE
```

5.2 FUNCIONES GENERALES.

Dentro de las funciones generales se encuentran las utilizadas para la gestión horarias y la linealización de los parámetros de las sondas.

5.2.1 FB Horario.

Este FB sirve para evaluar si las condiciones de horarios para la puesta en marcha de los sistemas están activas comparándolas con las condiciones horarias del equipo.

```
1  FUNCTION_BLOCK EnHorario
2  VAR_INPUT
3      config:ConfiguracionHoraria;
4
5  END_VAR
6
7  VAR_OUTPUT
8      EnHorarios:BOOL;
9
10 END_VAR
```

Aquí se puede observar cómo los parámetros de entrada es una estructura con las configuraciones horarias, del tipo de estructura ConfiguracionHoraria explicada anteriormente, y de salida tenemos una variable de tipo booleano que indica 0 si no está en horario y 1 si está en horario.

```

2
3 //Temporizador para activar periodicamente la FB NT_GetTime1
4 Timer1(IN:= NOT Timer1.Q , PT:=T#2S , Q=> , ET=> );
5
6
7 //Obtenemos tiempo de Windows
8 NT_Get_Time1(
9     NETID:= ,
10    START:=Timer1.Q ,
11    TMOUT:= ,
12    BUSY=> ,
13    ERR=> ,
14    ERRID=> ,
15    TIMESTR=> miTimeStruct );
16

```

En la parte del programa, primero se ejecuta el FB interno, creado en la librería de BECKHOFF, NT_GET_TIME para obtener el tiempo del sistema y seguidamente se almacena en una estructura propia de las librerías, donde se almacenan el año, el mes, el día, el día de la semana, la hora, los minutos, los segundos y los milisegundos.

Para obtener cíclicamente el tiempo del sistema se utiliza un temporizador de tipo TON (Imagen 13).

Su funcionamiento es muy sencillo, cuando el valor interno del temporizador llega al PT se activa la salida Q. Para activar el temporizador indicamos que se active cuando no este activo así nos aseguramos que cuando llegue al PT vuelva a empezar.

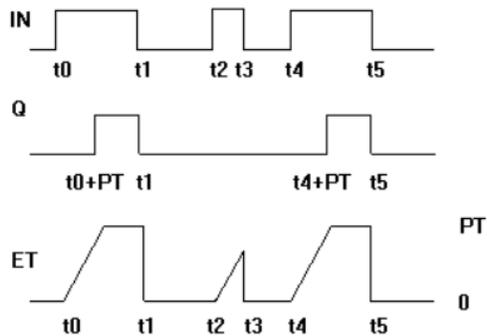


Imagen 13. Esquema de funcionamiento de un temporizador tipo TON.

Fuente: View Help TwinCAT BECKHOFF.

A continuación, se comprueba si estamos en un mes seleccionado. Para ello utilizamos el FB FB_EnMes que detallaremos más adelante. Se utiliza la función SETBIT32 de las librerías de TwinCAT para hacer una máscara en la variable meses_activo activando los bits de los meses activos. Después en el FB_EnMes se gestiona la información de los meses que deben estar activos, comparando la máscara creada anteriormente con el mes en que se encuentra el sistema.

```

17
18 //Comprovamos que estamos en mes
19 bitI:=(config.Mes_Inicio);
20 bitF:=(config.Mes_Final);
21 meses_Activo:=SETBIT32(inVal32:=meses_Activo , bitNo:= bitI );
22 meses_Activo:=SETBIT32(inVal32:=meses_Activo , bitNo:=bitF );
23 IF miTimeStruct.wMonth > bitI AND miTimeStruct.wMonth < bitF THEN
24     MesOk:= TRUE;
25 ELSIF miTimeStruct.wMonth > bitI THEN
26     MesOk:=FALSE;
27 ELSE
28     EnMes(meses_Activo:=meses_Activo , mes_actual:=miTimeStruct.wMonth , Mes=> );
29     MesOk:=EnMes.Mes;
30 END_IF
31

```

Las siguientes comprobaciones que se efectúan en el programa, son las que hacen referencia al día de la semana, las horas de funcionamiento del equipo y el rango del día de inicio y final de la programación horaria. Estas comprobaciones se realizan con la llamada a los FB correspondientes a las mismas. Estos FB se detallarán más adelante. Aquí podemos observar cómo hacer su llamada y los parámetros necesarios para su correcto funcionamiento.

```

32 //Comprovamos el dia de la semana
33 EnDiaSemana(
34     Dias_Activo:=config.Dias_Semana ,
35     Dia_Actual:=miTimeStruct.wDayOfWeek ,
36     Dias_Semana=> );
37 DiaSemanaOk:=EnDiaSemana.Dias_Semana;
38
39 //Comprovamos las horas de funcionamiento
40 EnHora(
41     hora_inicio:= config.Hora_Inicio,
42     hora_final:= config.Hora_Final,
43     minuto_inicial:= config.Minuto_Inicio,
44     minuto_final:= config.Minuto_Final,
45     hora_actual:= miTimeStruct.wHour ,
46     minutos_actual:= miTimeStruct.wMinute,
47     EnHora=> );
48 HoraOK:=EnHora.EnHora;
49
50 //Comprobamos que estemos en dia de mes para mes de inicio y final
51 EnDiaMes(
52     Dia_Mes_Inicio:= config.Dia_Inicio ,
53     Dia_Mes_Final:=config.Dia_Final ,
54     Dia_Actual:=miTimeStruct.wDay ,
55     Dia_Inicio=> ,
56     Dia_Final=> );
57 Dia_InicioOK:=EnDiaMes.Dia_Inicio;
58 Dia_FinalOK:=EnDiaMes.Dia_Final;

```

Cuando ya hemos evaluado todas las condiciones se comprueba que se cumplan todas para activar la salida del FB EnHorario y así poder activar las variables pertinentes en los demás fragmentos de código.

```

60 //Acción de control de la programación horaria
61 IF (miTimeStruct.wMonth= config.Mes_Inicio) THEN
62     EnHorarios:=MesOk AND DiaSemanaOk AND HoraOK AND Dia_InicioOK;
63 ELSIF (miTimeStruct.wMonth= config.Mes_Final) THEN
64     EnHorarios:=MesOk AND DiaSemanaOk AND HoraOK AND Dia_FinalOK;
65 ELSE
66     EnHorarios:= MesOk AND DiaSemanaOk AND HoraOK;
67 END_IF
68

```

Seguidamente mostraremos los FB que utilizamos en EnHorario con más detalle.

5.2.2 FB_DiaMes.

En este FB se evalúa si el día en que está el sistema es mayor al día de inicio indicado por el usuario o si es menor que el día final indicado. Las variables de salida son booleanas y son tratadas en el FB_EnHorario como se ha visto anteriormente.

```

1 //Dias del mes en el rango de bits 1...31,
2 //Ejemplo dia 18 es 16#00040000 porque va del bit 1 al 31 no escribe en el bit 0.
3
4 //Hacemos comparaciones para evaluar si está en día del mes
5 Dia_Inicio:= (Dia_Actual >= Dia_Mes_Inicio);
6 Dia_Final:=(Dia_Actual <= Dia_Mes_Final);

```

5.2.3 FB_DiaSemana.

En este FB se verifica si el día de la semana actual coincide con los marcados por el usuario. Se utiliza la función GETBIT32, de las librerías de TwinCAT, para obtener si el bit de la variable Dias_Activo, coincide con el día actual, si es así se escribe un 1 en la variable Dias_Semana, en caso contrario un 0.

```

1 //Hacemos comparaciones para evaluar si está en día de la semana
2 bitRevisar:= WORD_TO_SINT(Dia_Actual);
3 Dias_Semana:=GETBIT32(inVal32:=Dias_Activo , bitNo:=bitRevisar);

```

5.2.4 FB_Hora.

En este FB se identifica si la hora actual coincide con la marcada por el usuario. Para este fin se pasa la hora a minutos y se comprueba si estos minutos son mayores o menores que los de referencia. Si se cumplen las condiciones, estamos en hora, en caso contrario pondríamos un 0 en la variable de salida.

```
2 //Pasamos todo a minutos
3 inicio := hora_inicio*60+minuto_inicial;
4 mFinal := hora_final*60+minuto_final;
5 mActual := hora_actual *60+minutos_actual;
6
7 //Comparamos para evaluar si está en horario
8 EnHora:= ((mActual >= inicio) AND (mActual<= mFinal));
9
```

5.2.5 FB_EnMes.

Por último, tenemos el código para saber si estamos dentro del rango del mes. Se utiliza la función GETBIT32 para comprobar que en la máscara creada anteriormente, está el bit a 1 si está el mes activo o a 0 en caso contrario. Estos valores se almacenan en la variable booleana Mes.

```
1
2 //Hacemos comparaciones para evaluar si está en el mes
3 bitRevisar:= WORD_TO_SINT(mes_actual);
4 Mes:=GETBIT32(inVal32:=meses_Activo , bitNo:=bitRevisar );
5
```

Estas serían las funciones generales aplicadas a los diferentes sectores del sistema de control. Ahora se procederá con la explicación del código de cada uno de los sectores.

5.3 CLIMATIZACIÓN.

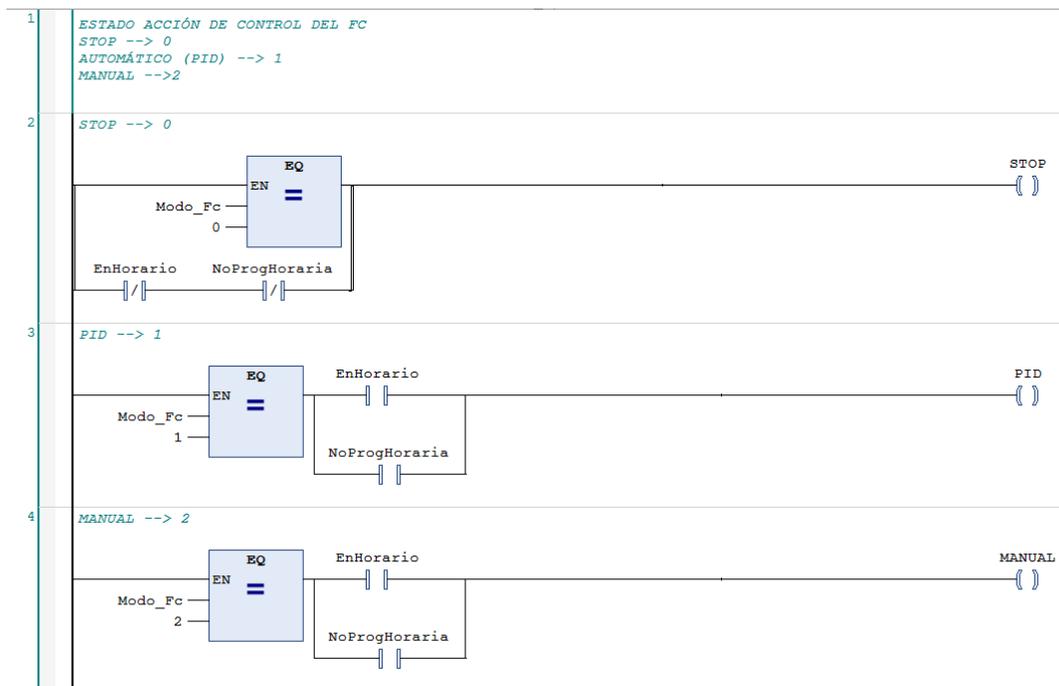
Este sector de sistema está compuesto por varios FB, estos son: CLIMATIZACION, CONFIGURACIONHORARIA, PARAM_PID y EXTRAXTOR_FALLOTERMICO.

Empezaremos con el FB CLIMATIZACION. Este FB ha sido desarrollado en LD.

5.3.1 FB CLIMATIZACION.

En este FB se desarrolla el código necesario para el control de una unidad de Fan Coil.

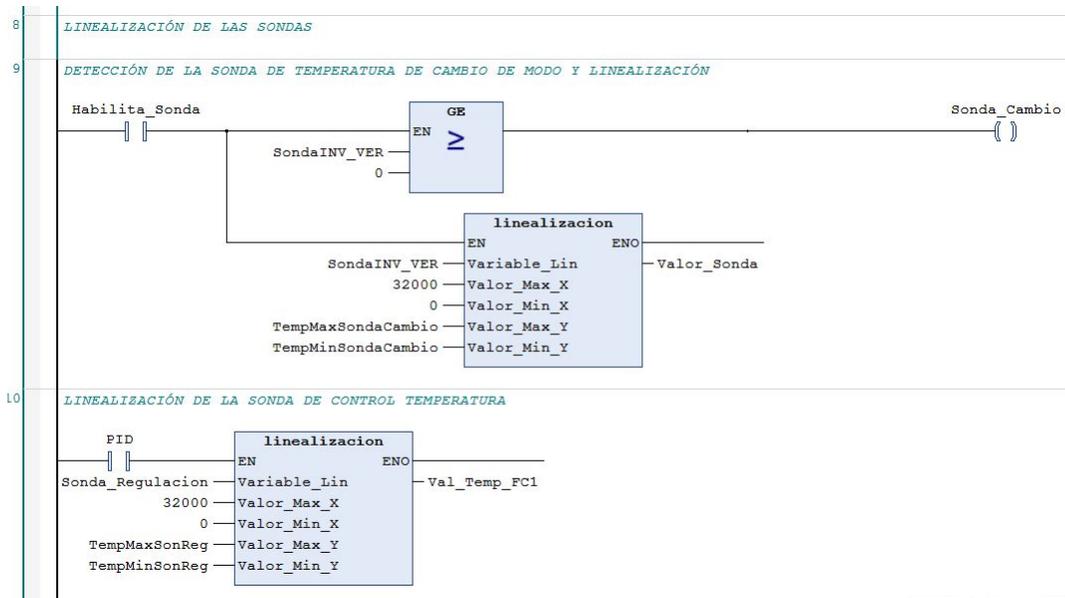
En la primera parte del código se hace la gestión del modo de funcionamiento. Con la variable de tipo entero Modo_FC se evalúa si está en modo STOP, PID (Automático) o MANUAL y se lo indicamos al sistema. Como particularidad se puede destacar que los modos PID y MANUAL no pueden funcionar si la programación horaria no es verdadera o esta desactivada.



Seguidamente, se comprueba la programación horaria para activar los modos de funcionamiento mencionados anteriormente.



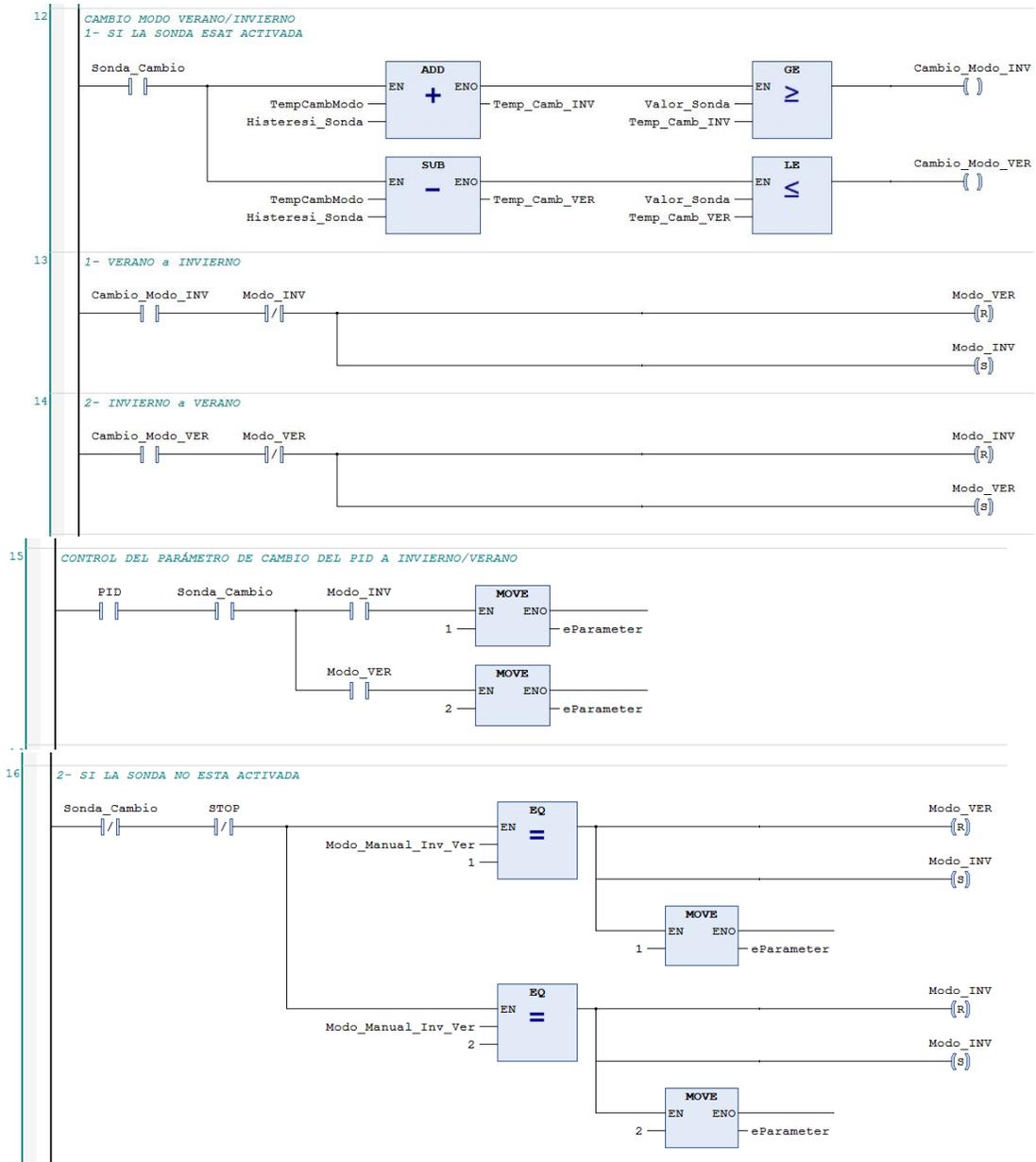
El siguiente paso es linealizar las sondas de temperatura utilizadas para el control de los Fan Coils. La sonda de cambio de modo solo esta activa si lo habilita el usuario, en caso contrario se desactiva la función de poder hacer el cambio de forma automática.



En este fragmento se realizan los cambios de modo de invierno a verano. Se pueden encontrar dos situaciones, cuando la sonda esta activada y cuando la deshabilitamos.

También se observa que se ha creado una histéresis para poder controlar mejor el cambio de modo cuando la temperatura sea inestable alrededor de un punto.

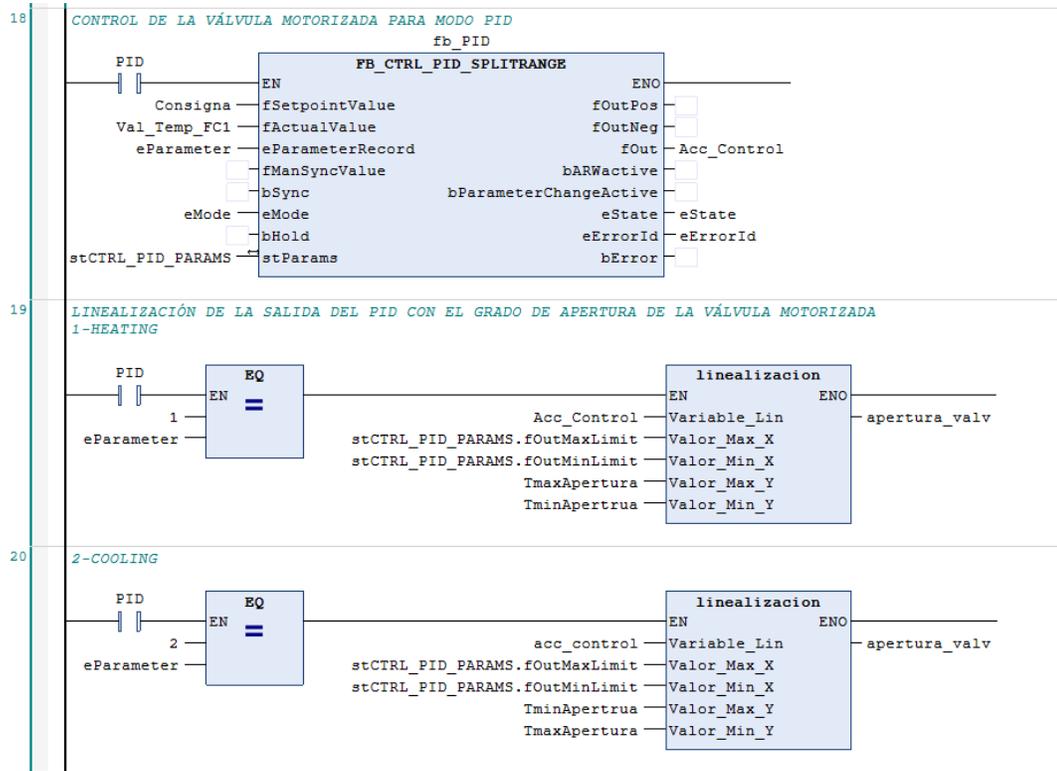
Las variables Modo_INV y Modo_VER son las que se utilizan más adelante en la gestión de salidas para comunicar al climatizador que proceda al cambio de modo y por último para poder realizar el cambio de modo en el PID tenemos la variable eParameter que es la que le indica al PID de control en qué modo debe funcionar.



Para el control de la regulación de temperatura del climatizador se ha optado por implementar un algoritmo del tipo PID ya recogido en un FB de las librerías de TwinCAT. El PID elegido tiene la particularidad que puede funcionar en 2 modos con los parámetros característicos del PID (K_p , T_i , T_d) diferentes en cada modo. Solo tienes que definir esos parámetros en la estructura de tipo `stCTRL_PID_PARAMS` y cambiar el modo de funcionamiento con la variable `eParameterRecord`.

Los parámetros de los PID han sido obtenidos del anterior sistema de control existente, ya que ha sido posible obtener esta información. Estos parámetros se volverían a reajustar en el momento de la instalación con la función *autotune* que proporcionan los PID utilizados.

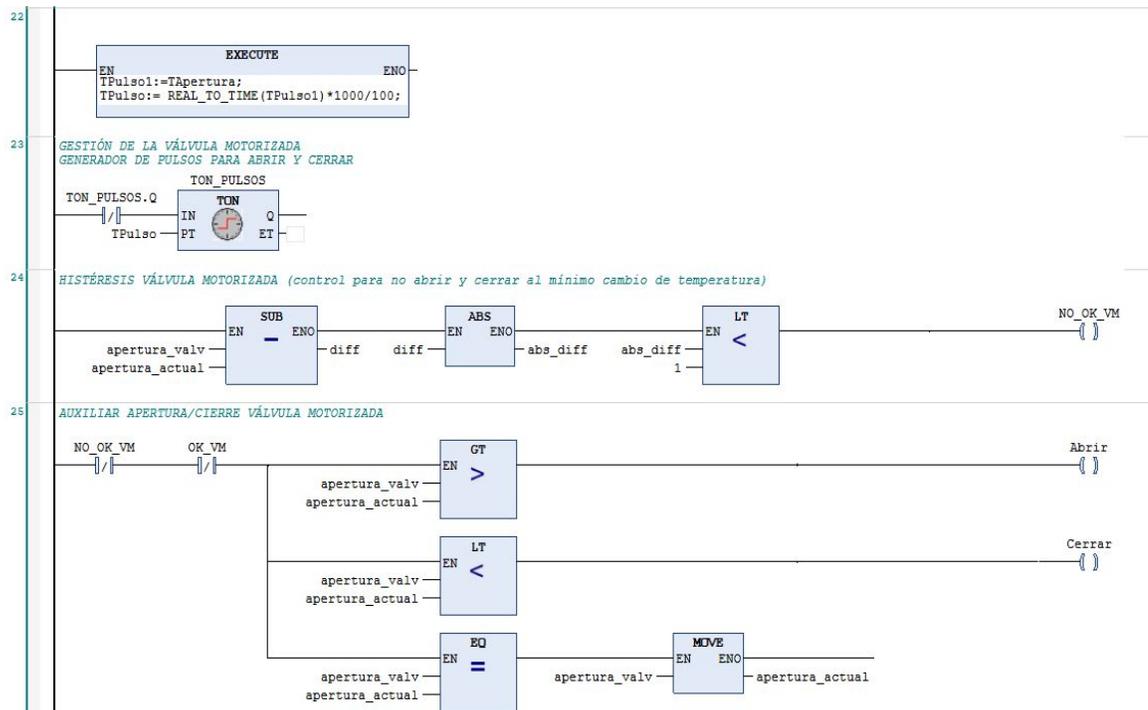
Luego la acción del control que se obtiene del PID la linealizamos para obtener la apertura de la válvula motorizada que deseamos en un rango especificado de 0-100%. Como se observa tenemos dos linealizaciones, una para el modo verano y otra para el modo invierno.



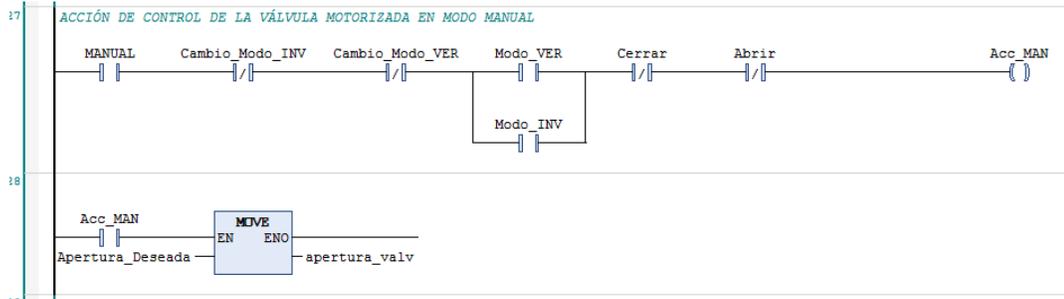
Cuando ya tenemos la acción de control, hay que actuar sobre la válvula motorizada. En esta primera parte se generan pulsos para abrir y cerrar la válvula, con un temporizador TON, ya que no se dispone de un sistema de alimentación continuo para su de movimiento.

Se define una histéresis para no abrir y cerrar la válvula innecesariamente cuando la fluctuación de la temperatura sea menor de 1°C.

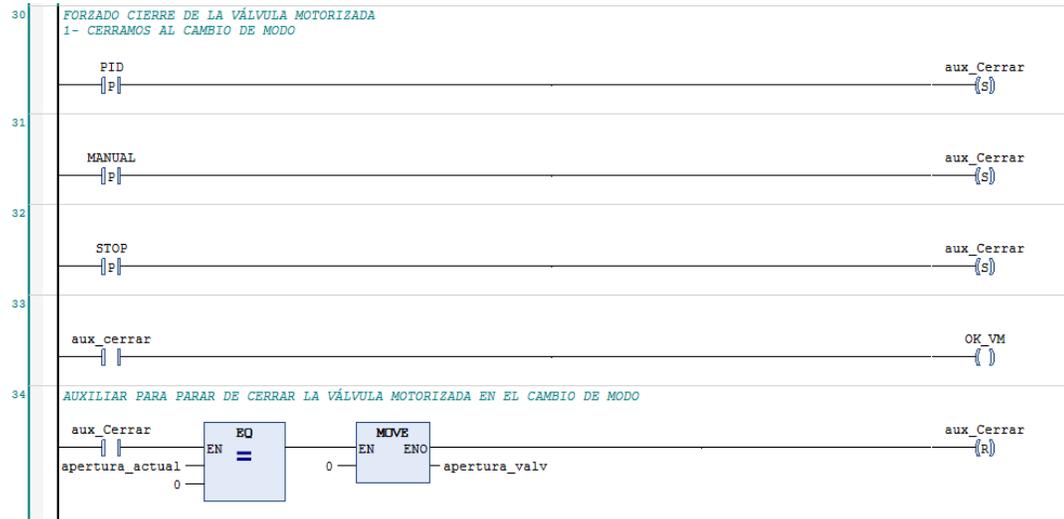
Seguidamente se activa o desactiva unos auxiliares para abrir y cerrar la válvula en función de si la apertura actual es mayor o menor a la deseada.



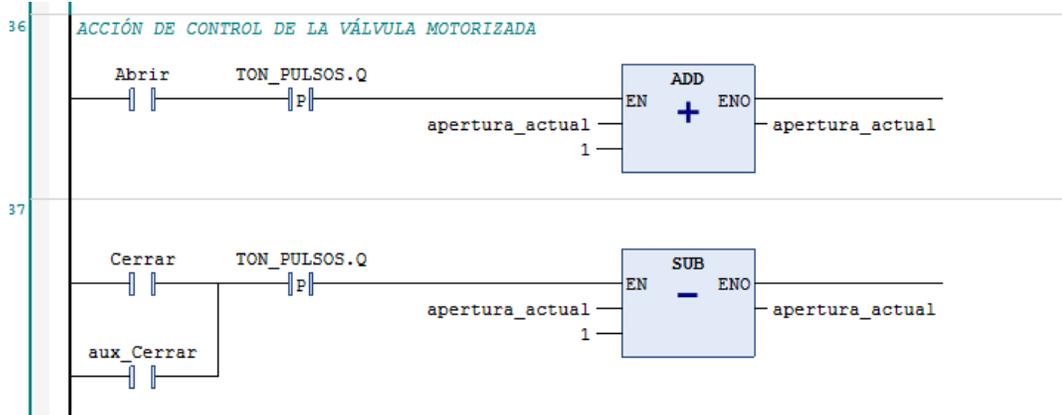
Cuando se tiene el modo de funcionamiento en modo manual se puede abrir la válvula motorizada al gusto del usuario y esto se consigue con el siguiente fragmento de código. La apertura deseada introducida por el usuario en las pantallas de control es copiada a la variable apertura_valv la cual es tratada por el resto de código para abrir o cerrar la válvula según proceda.



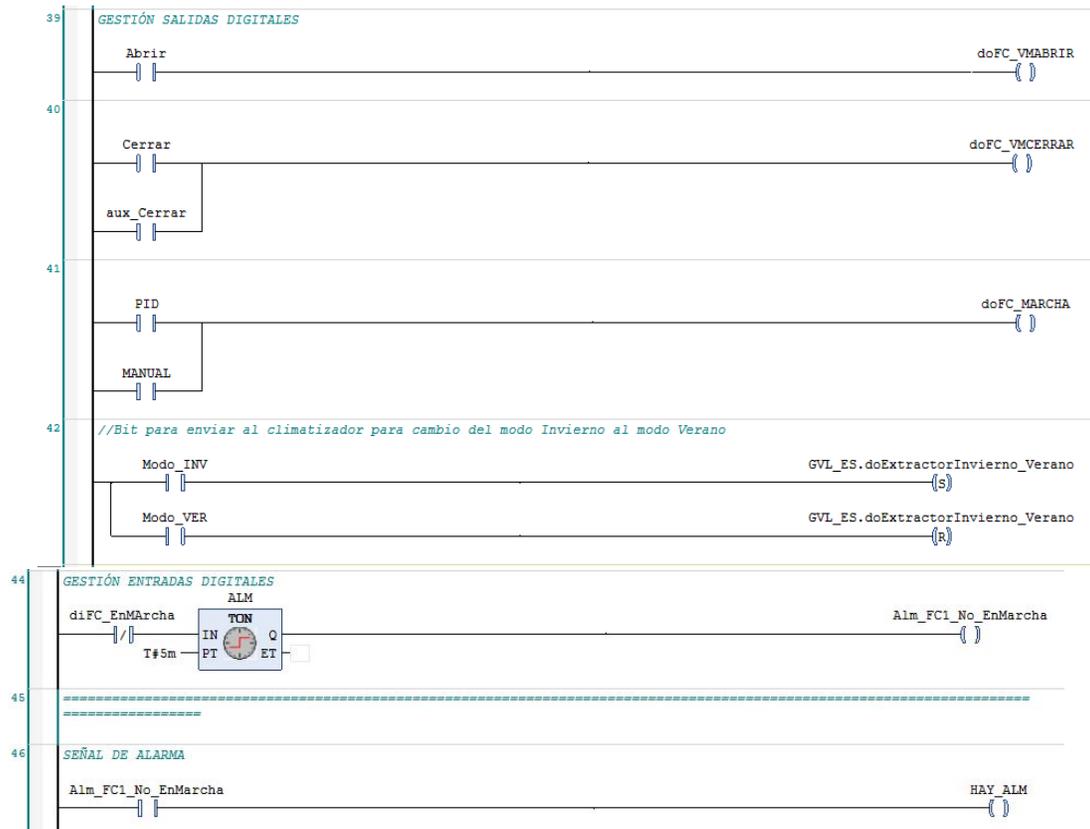
Una funcionalidad que deben tener los climatizadores es que en el cambio de modo de funcionamiento se debe cerrar la válvula motorizada completamente, esto se consigue con las siguientes líneas de código.



Por último, para tener implementado el control de la válvula motorizada se tienen que actualizar los valores de referencia de la válvula, ya que no se dispone de un sensor que muestre en que posición se encuentra la válvula. Para este fin se ha implementado un contador que aumenta o decrementa la apertura actual en función de los pulsos que está enviando el temporizador a la válvula motorizada para abrirla o cerrarla.



Finalmente se hace la gestión de las salidas digitales, las entradas digitales y las señales de alarma



5.3.2 FB_CONFIGURACIONHORARIA.

Para crear las configuraciones horarias del sector de la climatización se ha creado un FB donde se almacenan en las variables de la estructura los datos proporcionados por el usuario. Todas las asignaciones son directas excepto la de los días de la semana que están activos, ya que el usuario selecciona varios días. Para almacenar esa información en la variable de la estructura se necesita un bucle FOR y la función SETBIT32 para cambiar en la variable Dias_Semana los bits correspondientes de 0 a 1 en los días que están activos.

```
1  Dias_Semana:=0;
2  i:=0;
3      Config1.Mes_Inicio:=Mes_Inicio;
4      Config1.Mes_Final:=Mes_Final;
5      Config1.Dia_Inicio:=Dia_Inicio;
6      Config1.Dia_Final:=Dia_Final;
7      Config1.Hora_Inicio:=Hora_Inicio;
8      Config1.Hora_Final:=Hora_Final;
9      Config1.Minuto_Inicio:=Minuto_Inicio;
10     Config1.Minuto_Final:=Minuto_Final;
11
12     FOR i:=0 TO 6 DO
13         IF Dia_Semana_0 = TRUE AND i=0 THEN
14             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=0);
15         ELSIF Dia_Semana_1 = TRUE AND i=1 THEN
16             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=1);
17         ELSIF Dia_Semana_2 = TRUE AND i=2 THEN
18             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=2);
19         ELSIF Dia_Semana_3 = TRUE AND i=3 THEN
20             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=3);
21         ELSIF Dia_Semana_4 = TRUE AND i=4 THEN
22             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=4);
23         ELSIF Dia_Semana_5 = TRUE AND i=5 THEN
24             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=5);
25         ELSIF Dia_Semana_6 = TRUE AND i=6 THEN
26             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=6);
27         END_IF
28     END_FOR
29     Config1.Dias_Semana:=Dias_Semana;
30
```

5.3.3 FB_PARAM_PID.

También se ha implementado un FB para almacenar los parámetros del PID ya que en cada climatizador se podrían definir variables distintas. Otro de los factores que se ha tenido en cuenta a la hora de crear este FB es para dar una mejor visualización de código hacia futuras modificaciones.

```
1      stCTRL_PID_PARAMS.tCtrlCycleTime:= tCtrlCycleTime;
2      stCTRL_PID_PARAMS.tTaskCycleTime:= tTaskCycleTime;
3      stCTRL_PID_PARAMS.fKp_heating:= fKp_heating;
4      stCTRL_PID_PARAMS.tTn_heating:= tTn_heating;
5      stCTRL_PID_PARAMS.tTv_heating:= tTv_heating;
6      stCTRL_PID_PARAMS.tTd_heating:= tTd_heating;
7      stCTRL_PID_PARAMS.fKp_cooling:= fKp_cooling;
8      stCTRL_PID_PARAMS.tTn_cooling:= tTn_cooling;
9      stCTRL_PID_PARAMS.tTv_cooling:= tTv_cooling;
10     stCTRL_PID_PARAMS.tTd_cooling:= tTd_cooling;
11     stctrl_pid_params.nParameterChangeCycleTicks:=nParameterChangeCycleTicks;
12     stCTRL_PID_PARAMS.fOutMaxLimit:= fOutMaxLimit;
13     stCTRL_PID_PARAMS.fOutMinLimit:= fOutMinLimit;
```

5.3.4 FB_EXTRACTOR_FALLOTERMICO.

Este FB ha sido creado para el control de las alarmas a nivel de planta. Lo que nos permite este FB es no tener que repetir la gestión de las alarmas en el POU de climatización ya que sino tendríamos repetidas estas alarmas tantas veces como fan coils tengamos operativos en cada planta. De esta forma solo lo gestionamos una vez por planta. Como podemos observar controlamos las alarmas proporcionadas, si hay un fallo en el extractor y los fallos térmicos que puedan ocurrir.



5.4 ILUMINACIÓN.

En la parte de iluminación se ha realizado un POU por cada grupo de iluminación, mencionados anteriormente.

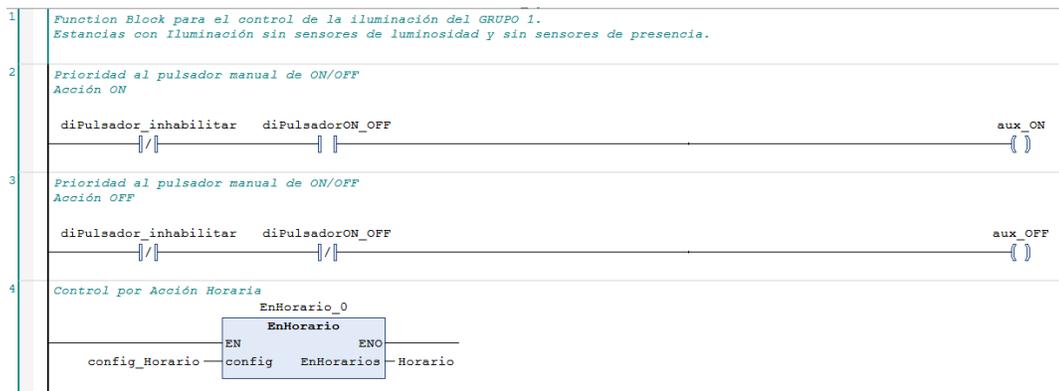
Los POU son muy parecidos entre si ya que tienen la misma estructura de programación, por eso se explicará el POU más básico, correspondiente al del Grupo 1, y sobre los otros grupos ese explicará las líneas de código añadidas para el tratamiento de los sensores correspondientes a cada grupo.

5.4.1 GRUPO 1.

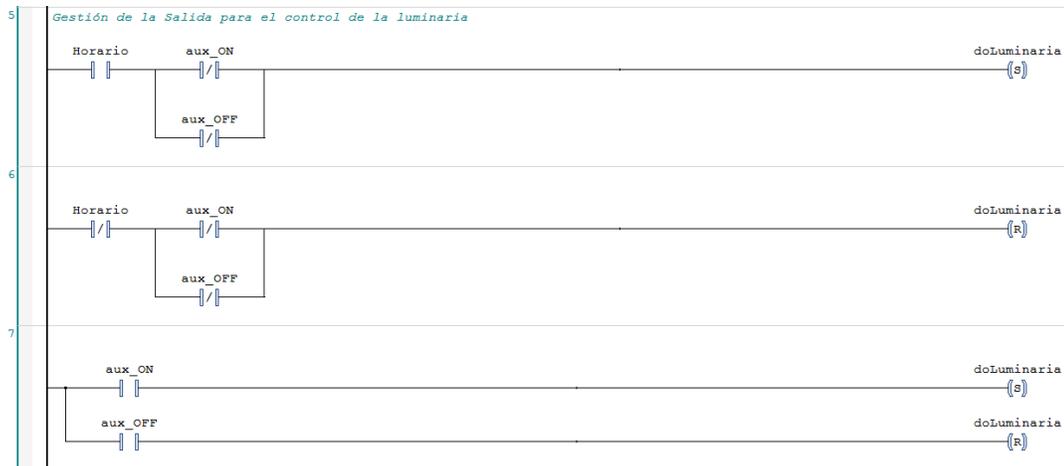
El funcionamiento básico del grupo 1 se basa en el control de la iluminación por medio de la programación horaria definida por el usuario y un pulsador manual, el cual puede ser inhabilitado.

En la primera parte del código se comprueba el estado del pulsador que inhabilita el pulsador manual, ya que si esta pulsado, para la gestión de las luces solo se utiliza la programación horaria. En caso contrario, cuando no está pulsado, se utiliza tanto la programación horaria como el pulsador manual.

La última línea del fragmento de código hace referencia al FB explicado anteriormente para la comprobación de la programación horaria.



A continuación, se muestran las últimas líneas de código las cuales hacen referencia al tratamiento de la variable de salida, para indicar si se procede a encender las luminarias.



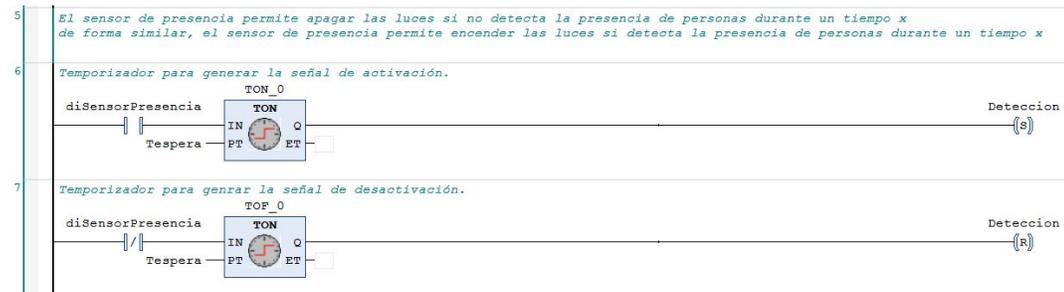
5.4.2 GRUPO 2.

En el grupo 2 la primera parte del código es idéntica a la del grupo 1 ya que tiene las mismas funcionalidades.

La nueva propiedad de este grupo es la incorporación del sensor de presencia.

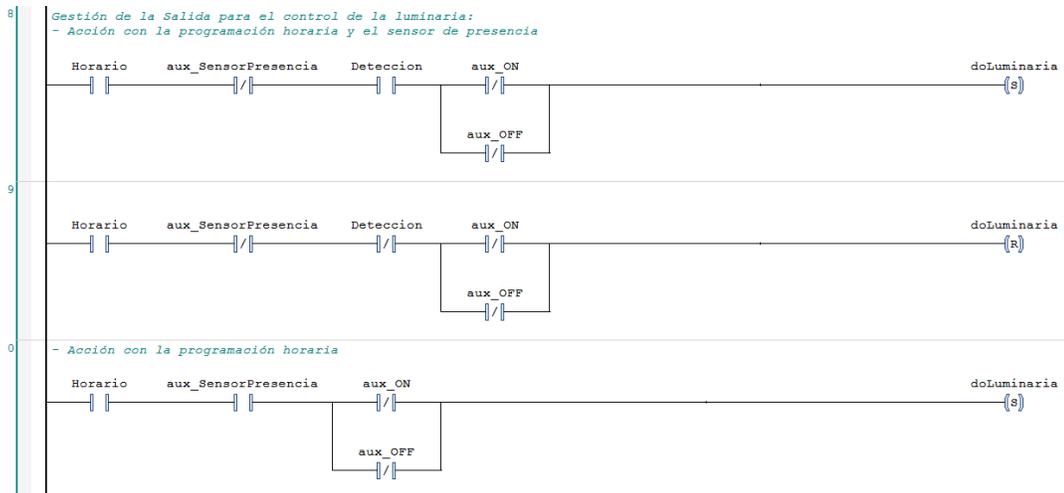
Seguidamente mostraremos el código necesario para gestionar estos sensores.

Estos sensores se caracterizan por el hecho que para encender las luces tienen que detectar durante un tiempo mínimo la presencia de alguna persona. Esto se consigue con la incorporación de temporizadores de tipo TON. La misma filosofía aplicamos para la desactivación de las luminarias.



En la parte de gestión de las salidas solo se incorpora el funcionamiento de los sensores de presencia, el resto es análogo al grupo 1.

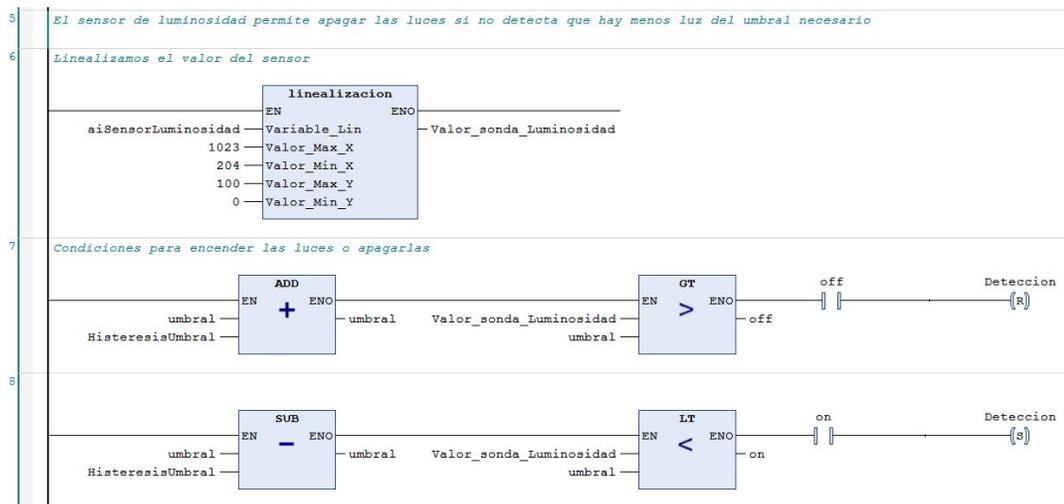
Por último, la variable `aux_SensorPresencia` que se muestra en el fragmento de código posterior, hace referencia a una entrada digital con la que podemos deshabilitar el sensor desde las pantallas de visualización de cada sector de luminarias.



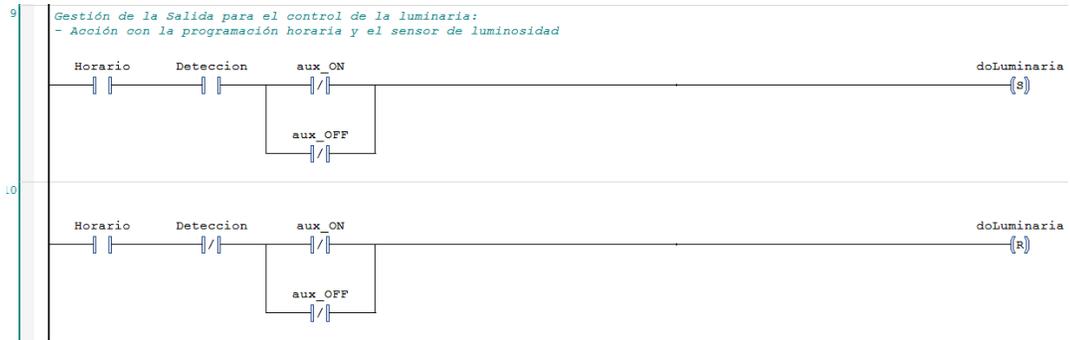
5.4.3 GRUPO 3.

Este nuevo grupo es el que incorpora sensores de luminosidad, los cuales detectan un nivel de luminosidad y lo transmiten a través de una señal analógica. Luego esta señal se normaliza y compara con las franjas definidas por el usuario. Dependiendo si está dentro de la franja se encienden o apagan las luces.

Aquí se puede observar cómo realizamos esa normalización con la función linealización y luego hacemos unas comparaciones con el umbral definido por el usuario. Podemos observar cómo tenemos una histéresis, esto es para evitar la activación y desactivación de las luminarias cuando el nivel de luz oscile poco entorno al nivel de referencia.



Por último se muestran las nuevas líneas añadidas para la gestión de la salida.



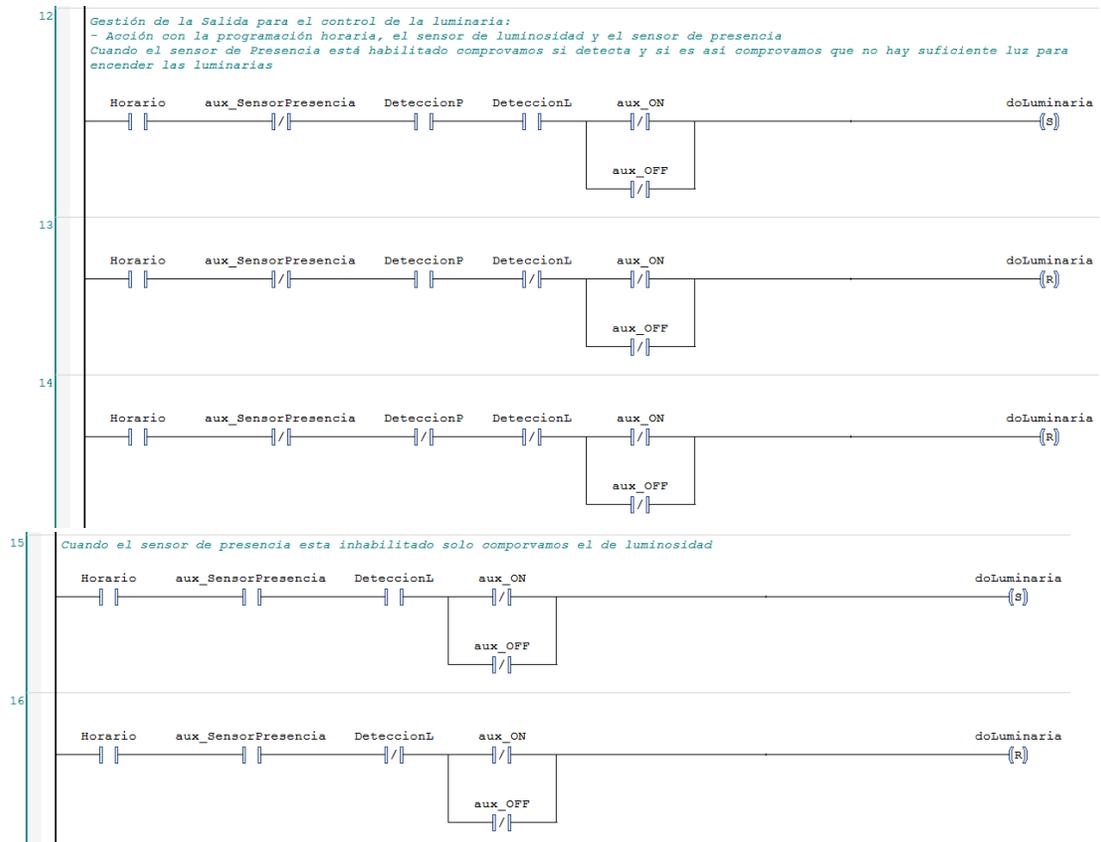
5.4.4 GRUPO 4.

Este último grupo es el que integra los dos sensores, iluminación y presencia, la programación horaria y el pulsador manual.

Por lo tanto, las partes de tratamiento de los sensores, la programación horaria y el pulsador son idénticas a las descritas en los otros grupos.

Lo único que cambia son las condiciones que se comprueban para realizar la activación o desactivación de la luminaria y es lo que mostramos en el fragmento de código siguiente.

Las variables DeteccionP y DeteccionL hacen referencia al sensor de presencia y luminosidad respectivamente.

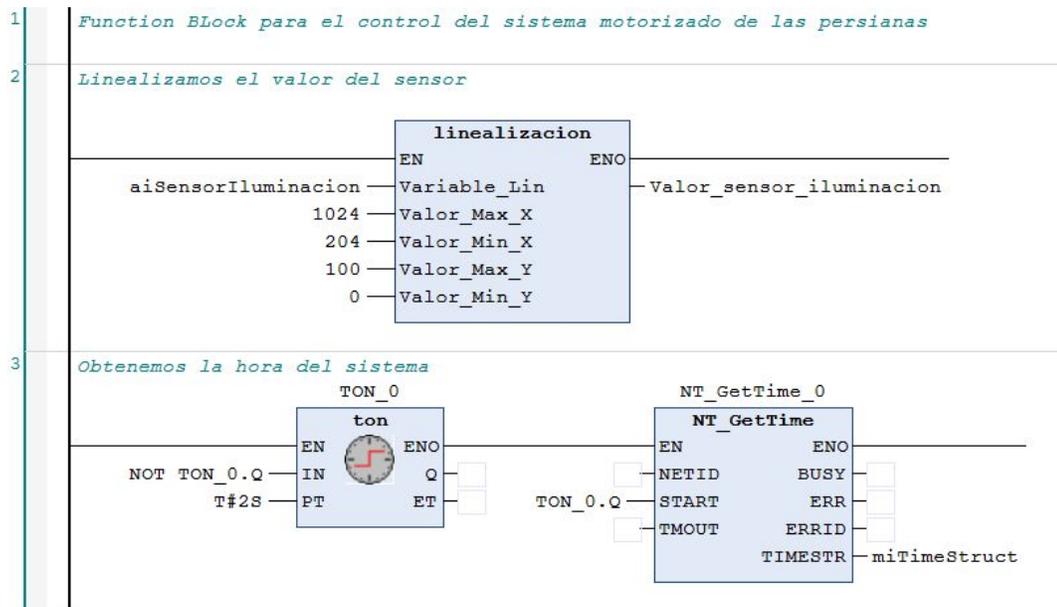


5.4.5 PERSIANAS.

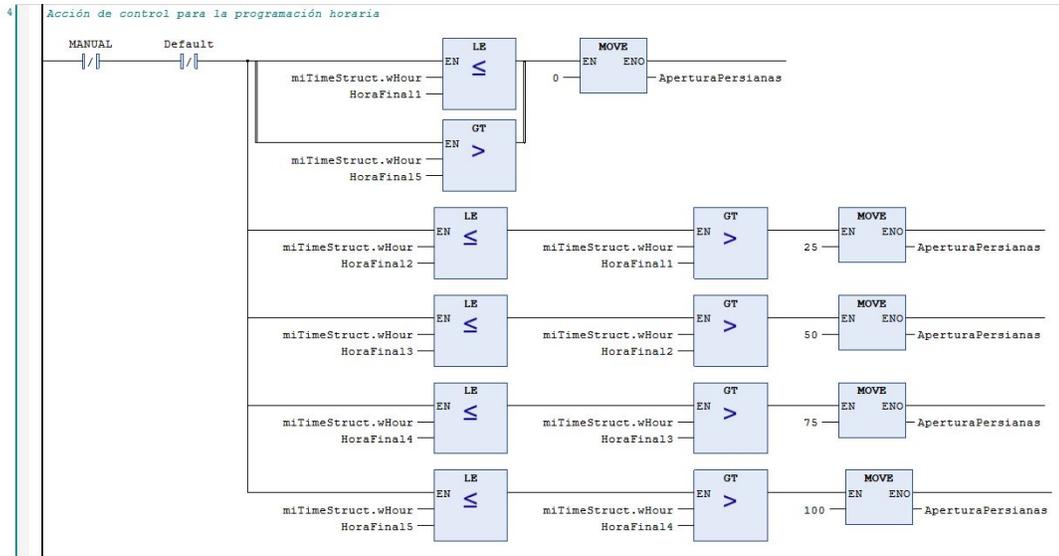
Las persianas se componen de grupos de láminas en las que se controla el nivel de apertura mediante un sistema motorizado, que se controla mediante una salida analógica (4-20 mA --> 0...100%). Las opciones de control que tenemos son las siguientes:

- Por niveles de apertura mediante programación horaria.
- Por niveles de apertura en función de la intensidad de luz exterior. Se definen franjas de luminosidad y cuando nos encontremos en una determinada franja, esta se corresponderá con un determinado nivel de apertura.
- Control manual para abrir o cerrar las persianas el (grado deseado de apertura entre 0...100%) con prioridad a los otros métodos.

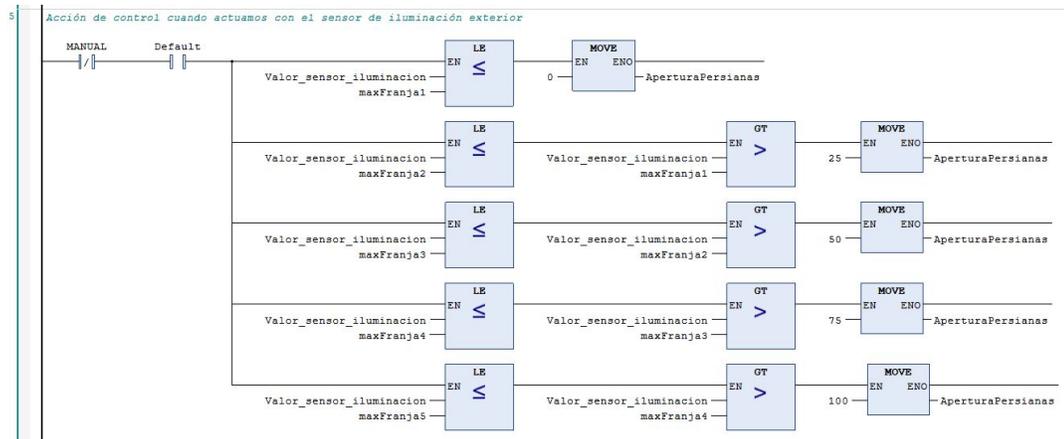
Primeramente, tenemos la gestión de las señales del sensor de luminosidad, que se normaliza de forma similar como en los grupos de iluminación, y el tratamiento de la programación horaria.



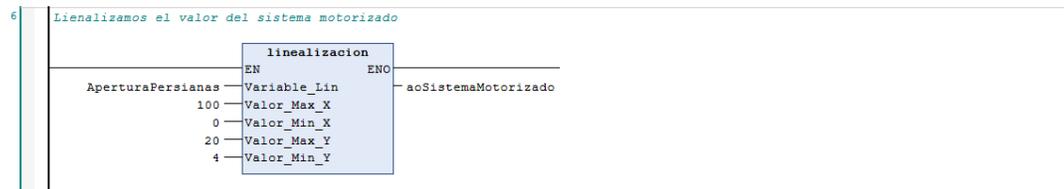
Aquí se pueden observar las condiciones impuestas para la apertura de las persianas cuando está funcionando con la programación horaria. Se comprueba entre la hora actual del sistema y las franjas definidas por el usuario. Sólo está permitido el uso de 5 franjas horarias.



De forma similar a la acción de control para la programación horaria, se actúa con el sensor de luminosidad, lo único que varía es la comprobación del valor linealizado del sensor y se compara con las franjas de luminosidad que ha definido el usuario. Volvemos a tener un máximo de 5 Franjas.



Seguidamente se linealiza el valor del sistema motorizado para indicar a la salida analógica su acción de control.



Por último, se gestiona la acción de control del modo manual y la gestión de las prioridades de los modos de funcionamiento.



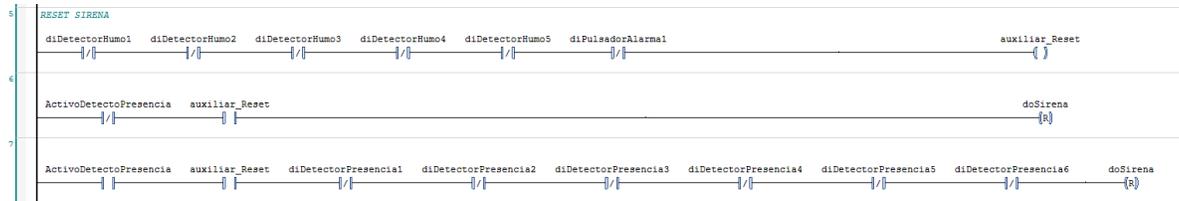
5.4.6 PROGRAMACIÓN HORARIA.

En la iluminación se utiliza una programación horaria un poco diferente a la de la climatización ya que solo se controlan los días de la semana que están activos y las horas a funcionar. Por lo tanto para no tener que modificar el POU de EnHorario ni tener que hacer una nueva estructura para las configuraciones horarias de la iluminación, hemos hecho un nuevo POU parecido al de configuración Horaria realizado para la climatización, pero fijando tanto los meses de inicio y final como los días desde el 1 de enero hasta el 31 de diciembre, así no nos afectara en nuestra programación horaria.

```

1  Dias_Semana:=0;
2  i:=0;
3      Config1.Mes_Inicio:=1;
4      Config1.Mes_Final:=12;
5      Config1.Dia_Inicio:=1;
6      Config1.Dia_Final:=31;
7      Config1.Hora_Inicio:=Hora_Inicio;
8      Config1.Hora_Final:=Hora_Final;
9      Config1.Minuto_Inicio:=Minuto_Inicio;
.0     Config1.Minuto_Final:=Minuto_Final;
.1
.2     FOR i:=0 TO 6 DO
.3         IF Dia_Semana_0 = TRUE AND i=0 THEN
.4             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=0);
.5         ELSIF Dia_Semana_1 = TRUE AND i=1 THEN
.6             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=1);
.7         ELSIF Dia_Semana_2 = TRUE AND i=2 THEN
.8             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=2);
.9         ELSIF Dia_Semana_3 = TRUE AND i=3 THEN
.10            Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=3);
.11           ELSIF Dia_Semana_4 = TRUE AND i=4 THEN
.12            Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=4);
.13           ELSIF Dia_Semana_5 = TRUE AND i=5 THEN
.14            Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=5);
.15           ELSIF Dia_Semana_6 = TRUE AND i=6 THEN
.16            Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=6);
.17           END_IF
.18       END_FOR
.19       Config1.Dias_Semana:=Dias_Semana;
.20

```

5.6 IoT.

Como se ha comentado en anteriormente en el apartado de IoT, el objetivo de esta parte de código es subir los datos al servidor de AZURE, ya que el servidor debe de estar creado y operativo para el uso estimado por parte del usuario.

Para poder establecer la conexión se necesitan tres variables relacionadas con el dispositivo del servidor. Estas variables son: el nombre del dispositivo, el cual lo almacenamos en la variable sDeviceId, un certificado el cual debe estar en el PC con el que queremos transmitir los datos, de este certificado se necesita la ruta para almacenarla en la variable sRutaCertificadoAzure, y una clave llamada SAS Token que se genera en la plataforma de Microsoft Azure, y luego almacenamos posteriormente en la variable sAzureSasToken.

Teniendo estas variables ya es posible establecer la conexión con el dispositivo.

```

39  VAR CONSTANT
40  sDeviceId : STRING := 'PC_PLANTA_2';
41  sRutaCertificadoAzure : STRING(255) := 'C:\Users\PCDisinel\Desktop\TFG\Documentos TFG\Azure.cer
42  sAzureSasToken : STRING(511) := 'HostName=IoTHubAndresDisinel.azure-devices.net;DeviceId=PC_PLA
43  END_VAR

```

Para hacer la conexión, se ha utiliza la variable bInicializacion que solo esta activa en el primer ciclo de scan. Lo que se hace es utilizar el FB_IotMqttClient para crear el cliente comunicando por MQTT y preparar el topic al que publicar los datos.

Para que se pueda conectar al dispositivo, en el FB siempre tiene que estar activa la variable.Execute. Para ello se ha creado la variable bConnectar que siempre está en TRUE.

```

1  //Hacemos la conexión al dispositivo IoT
2  IF bInicializacion THEN
3      bInicializacion := FALSE;
4      // Preparar topics a los que publicar
5      sTopicAlQuePublicar := 'devices/PC_PLANTA_2/messages/events/readpipe/';
6      // Datos para la conexión a Azure IoT Hub por MQTT
7      fbMqttClient.stTLS.sCA := sRutaCertificadoAzure;
8      fbMqttClient.stTLS.sAzureSas := sAzureSasToken;
9      fbMqttClient.ipMessageQueue := fbMessageQueue;
10  END_IF
11  //Comprobar si esta conectado
12  fbMqttClient.Execute(bConnectar);

```

Si ya se ha creado la conexión en la variable bConnected, encontraremos un valor TRUE, por lo tanto, ya se puede enviar valores a nuestro dispositivo.

Para enviar los mensajes de forma periódica se utiliza un temporizador programado con un tiempo que deberá establecer el usuario.

Por último, para enviar los datos al dispositivo se ha recurrido al uso de los mensajes de tipo JSON. La secuencia que utilizamos es: creación del objeto y su apertura, adición de las claves y valores del mensaje, clausura del mensaje, conversión del mensaje en variable tipo STRING y finalmente la publicación del telegrama JSON al dispositivo.

```
19 // Si hemos conectado con el Dispositivo tendremos una realimentación TRUE.
20 IF fbMqttClient.bConnected THEN
21     // Dar la orden de envío de datos cada minuto
22     fbTemporizador(IN:=TRUE,PT:=T#1M);
23     IF fbTemporizador.Q THEN
24         fbTemporizador(IN:=FALSE);
25         // Crear telegrama JSON a enviar
26         // Vacía el objeto JSON
27         fbCrearJson.ResetDocument();
28         // Añade la llave de apertura
29         fbCrearJson.StartObject();
30         // Añade las parejas clave/valor al mensaje
31         fbCrearJson.AddKey('Energía Activa');
32         fbCrearJson.AddReal(Energ_Activa);
33         fbCrearJson.AddKey('Energía Reactiva');
34         fbCrearJson.AddReal(Energ_Reactiva);
35         fbCrearJson.AddKey('Energía Aparente');
36         fbCrearJson.AddReal(Energ_Aparente);
37         fbCrearJson.AddKey('Fecha');
38         fbCrearJson.AddString(Fecha);
39         // Añade la llave de clausura
40         fbCrearJson.EndObject();
41         // Obtiene el STRING con el mensaje JSON
42         sJson := fbCrearJson.GetDocument();
43         // Publicar el telegrama JSON a la cuenta de Azure
44         fbMqttClient.Publish(sTopic := sTopicAlQuePublicar,
45                             pPayload:= ADR(sJson),
46                             nPayloadSize:= TO_UDINT(LEN(sJson)),
47                             eQoS:= TcIotMqttQos.AtMostOnceDelivery,
48                             bRetain:= FALSE,
49                             bQueue:= FALSE );
50     END_IF
51 END_IF
```

5.7 MAIN.

A continuación, se mostrarán algunos fragmentos los cuales deben de estar en el POU principal del control del edificio. Aquí se explicará cómo hay que pasar la información a todos los POU que controlan los diferentes sectores del centro de salud.

Primeramente, mostraremos cómo configurar un fan coil del climatizador, para añadir más fan coils solo habría que replicar estas líneas de código cambiando las variables del fan coil 1 al fan coil 2.

```
85 //Almacenamos el valor de la temperatura a conseguir con el PID
86 IF Modo_FC1 = 1 THEN
87     Temp_Deseada_FC1:=Temp_Deseada_VER_FC1;
88 ELSE
89     Temp_Deseada_FC1:=Temp_Deseada_INV_FC1;
90 END_IF
91 (*Repetiríamos la estructura anterior para los demás Fan Coils*)
92
93 //Control de la Planta 1
94 // Programa para el control del FC1
95 CLIMATIZACION(
96     TempMaxSonReg:= TempMaxSondaRegulacionFC1 ,
97     TempMinSonReg:= TempMinSondaRegulacionFC1,
98     Sonda_Regulacion:= aiFC1_STempReg,
99     doFC_VMABRIR:= doFC1_VMABRIR,
100    doFC_VMCERRAR:= doFC1_VMCERRAR,
101    doFC_MARCHA:= doFC1_MARCHA,
102    diFC_EnMarcha:= diFC1_EnMarcha,
103    SondaINV_VER:= aiSTempINV_VER,
104    TempMaxSondaCambio:= TempMaxSonLin,
105    TempMinSondaCambio:= TempMinSonLin,
106    Habilita_Sonda:= HabilitaSonda_Pl,
107    TempCambModo:= TempCambioModo,
108    Histeresi_Sonda:= HisteresiSonda,
109    Modo_Manual_Inv_Ver:= Modo_INVVER,
110    Consigna:= Temp_Deseada_FC1,
111    Modo_Fc:= Modo_FC1,
112    TApertura:= Tiempo_AperturaFC1,
113    Apertura_Deseada:= Apertura_Deseada_FC1,
114    NoProgHoraria:= OFFProg_Horaria,
115    conf_Horaria0:=Configuracion_Horaria ,
116    stCTRL_PID_PARAMS:= stCTRL_PID_PARAMS,
117    HAY_ALM=> );
```

Para la creación y gestión de las diferentes configuraciones horarias se añade el siguiente código.

```
8 //Creación de la configuración horaria
9 ConfHor1(
10     Mes_Inicio:=Mes_Inicio,
11     Mes_Final:= Mes_Final,
12     Dia_Inicio:= Dia_Inicio,
13     Dia_Final:= Dia_Final,
14     Dia_Semana_0:= Dia_Semana_0,
15     Dia_Semana_1:= Dia_Semana_1,
16     Dia_Semana_2:= Dia_Semana_2,
17     Dia_Semana_3:= Dia_Semana_3,
18     Dia_Semana_4:= Dia_Semana_4,
19     Dia_Semana_5:= Dia_Semana_5,
20     Dia_Semana_6:= Dia_Semana_6,
21     Hora_Inicio:= Hora_Inicio,
22     Hora_Final:= Hora_Final,
23     Minuto_Inicio:= Minuto_Inicio,
24     Minuto_Final:= Minuto_Final,
25     Config1=> );
--
```

Si quisiéramos más configuraciones horarias sería la misma función, pero cambiando las variables por las que proceda.

Para la selección de la programación horaria en cada fan coil se añade una sentencia de tipo CASE con la cual elegimos la programación dependiendo de la elección que el usuario indique en la pantalla principal de su fan coil.

```

47| CASE Conf_FC1 OF
48|   1: //Asignamos la configuración horaria 1
49|     Configuracion_Horaria:=ConfHor1.Config1;
50|     (* Haríamos la asignación de la configuración horaria cuando
51|     estubieran creadas todas las configuraciones.
52|   2: //Asignamos la configuración horaria 2
53|     Configuracion_Horaria:=ConfHor2.Config1;
54|   3: //Asignamos la configuración horaria 3
55|     Configuracion_Horaria:=ConfHor.Config1;
56|   4: //Asignamos la configuración horaria 4
57|     Configuracion_Horaria:=ConfHor.Config1;
58|     *)
59| END_CASE
60|

```

Para los parámetros de los PID utilizaríamos la siguiente lista de instrucciones. La cual repetiríamos para los demás Fan coils.

```

61| //Almacenamos los parámetros del PID
62| tTn_cool:=INT_TO_TIME(tTn_cooling)*1000;
63| tTv_cool:=INT_TO_TIME(tTv_cooling)*1000;
64| tTn_heat:=REAL_TO_TIME(tTn_heating)*1000;
65| tTv_heat:=INT_TO_TIME(tTv_heating)*1000;
66|
67| stPID(
68|   tCtrlCycleTime:= tCtrlCycleTime ,
69|   tTaskCycleTime:= tTaskCycleTime,
70|   fKp_heating:= fKp_heating,
71|   tTn_heating:= tTn_heat,
72|   tTv_heating:= tTv_heat,
73|   tTd_heating:= tTd_heating,
74|   fKp_cooling:= fKp_cooling,
75|   tTn_cooling:= tTn_cool,
76|   tTv_cooling:= tTv_cool,
77|   tTd_cooling:= tTd_cooling,
78|   nParameterChangeCycleTicks:= nParameterChangeCycleTicks,
79|   fOutMaxLimit:= fOutMaxLimit,
80|   fOutMinLimit:= fOutMinLimit,
81|   stCTRL_PID_PARAMS=> );
82| stCTRL_PID_PARAMS:=stPID.stCTRL_PID_PARAMS;

```

Por último, para el control total del sector de climatización debemos añadir un FB por planta para la gestión de las alarmas de esta.

```

120| //Function Block para la gestión de las alarmas de la planta 1
121| EXT_FALLO_TERMICO(
122|   FC1_EnMarcha:= diFC1_EnMarcha,
123|   FC2_EnMarcha:= diFC2_EnMarcha,
124|   FC3_EnMarcha:= diFC3_EnMarcha,
125|   FC4_EnMarcha:= diFC4_EnMarcha,
126|   ExtractorMarcha:= diExtractorMarcha,
127|   FalloTermico:= diFalloTermico,
128|   Alm_NoExtractorEnMarcha=> ,
129|   Alm_Fallo_Termico=> ,
130|   HAY_ALM=> );
131| // Señal para activar la alarma de la pantalla principal
132| Hay_Alarma:= CLIMATIZACION.HAY_ALM OR EXT_FALLO_TERMICO.HAY_ALM;

```

Para el control de la iluminación y seguridad indicaremos cuales son los FBs a ejecutar para cada grupo. El uso de estos FBs es muy sencillo ya que solo hay que indicar las variables de entrada correspondiente a cada grupo y luego gestionar la salida que nos entrega con las salidas de nuestro grupo.

Grupo1.

```
172 //Grupo 1
173 //Pasillo interior Planta 2, uso de 2 luminarias.
174 PasilloP2_G1(
175     diPulsador_inhabilitar:= diPulsadorInhabilitar,
176     diPulsadorON_OFF:= diPulsadorONOFF,
177     config_Horario:= Horario_Iluminacion,
178     doLuminaria=> );
179 //Control de las dos luminarias
180 doLuminaria1:=PasilloP2_G1.doLuminaria;
181 doLuminaria2:=PasilloP2_G1.doLuminaria;
182
```

Grupo2.

```
183 //Grupo 2
184 //Consultas médicas 1 Planta 1, uso de 1 luminaria y 1 sensor de presencia
185
186 Consulta1_P1(
187     diPulsador_inhabilitar:=diPulsadorInhabilitar2 ,
188     diPulsadorON_OFF:= diPulsadorONOFF2,
189     diSensorPresencia:= diSensorPresencia,
190     aux_SensorPresencia:= inhabilitarSensorPresencia,
191     Tespera:= Tesperal,
192     config_Horario:=Horario_Iluminacion ,
193     doLuminaria=> );
194 //Control de las luminarias
195 doLuminaria3:=Consulta1_P1.doLuminaria;
196
```

Grupo3.

```
197 //Grupo 3
198 //Iluminación Enfermería 2 Planta 1, uso de 1 luminaria y 1 sensor de Luminosidad
199 Enfermeria2_P1(
200     diPulsador_inhabilitar:= diPulsadorInhabilitar3,
201     diPulsadorON_OFF:= diPulsadorONOFF3,
202     aiSensorLuminosidad:= aiSensorLuminosidad,
203     umbral:= umbral,
204     HisteresisUmbral:= histeresiLuminosidad,
205     config_Horario:= Horario_Iluminacion,
206     doLuminaria=> );
207 //Control de las luminarias
208 doLuminaria4:=Enfermeria2_P1.doLuminaria;
209
```

Grupo4.

```
210 //Grupo 4
211 //Iluminación Enfermería 3 Planta 1, uso de 1 luminaria, 1 sensor de luminosidad y 1 sensor de presencia
212 Enfermeria3_P1(
213   diPulsador_inhabilitar:= diPulsadorInhabilitar4,
214   diPulsadorON_OFF:= diPulsadorONOFF4,
215   aiSensorLuminosidad:= aiSensorLuminosidad2,
216   umbral:= umbral2,
217   HisteresisUmbral:= histeresiLuminosidad2,
218   diSensorPresencia:= diSensorPresencia2 ,
219   aux_SensorPresencia:= inhabilitarSensorPresencia2,
220   Tespera:= Tespera2,
221   config_Horario:= Horario_Iluminacion,
222   doLuminaria=> );
223 //Control de las luminarias
224 doLuminaria5:=Enfermeria3_P1.doLuminaria;
---
```

Persianas.

```
226 //Control Persianas
227 //Laminas Persianas patio P2
228 PatioP2(
229   aiSensorIluminacion:= aiSensorIluminacion,
230   maxFranja1:= maxFranja1,
231   maxFranja2:= maxFranja2,
232   maxFranja3:= maxFranja3,
233   maxFranja4:= maxFranja4,
234   maxFranja5:= maxFranja5,
235   HoraFinal1:= HoraFinal1,
236   HoraFinal2:= HoraFinal2,
237   HoraFinal3:= HoraFinal3,
238   HoraFinal4:= HoraFinal4,
239   HoraFinal5:= HoraFinal5,
240   PrioridadHorario:= PrioridadHoraria,
241   PrioridadSensor:= PrioridadSensor,
242   Apertura:= Apertura,
243   MANUAL:=Manual,
244   aoSistemaMotorizado=> );
245 //Control de las láminas de las persianas
246 aoSistemaMotorizado:=PatioP2.aoSistemaMotorizado;
247
```

Seguridad.

```
3 (*=====
3 =====*)
1 //Código para el control de la SEGURIDAD.
1 //Ejemplo del uso del function block de seguridad.
2
3 seguridad_p2(
4     diDetectorHumo1:= diDetectorHumo1 ,
5     diDetectorHumo2:= diDetectorHumo2,
5     diDetectorHumo3:= diDetectorHumo3,
7     diDetectorHumo4:= diDetectorHumo4,
3     diDetectorHumo5:= diDetectorHumo5,
3     diPulsadorAlarma1:= diPulsadorAlarma1,
3     diPulsadorAlarma2:=diPulsadorAlarma2 ,
1     diPulsadorAlarma3:=diPulsadorAlarma3,
2     diPulsadorAlarma4:=diPulsadorAlarma4,
3     diDetectorPresencia1:= diDetectorPresencia1,
1     diDetectorPresencia2:= diDetectorPresencia2,
5     diDetectorPresencia3:= diDetectorPresencia3,
5     diDetectorPresencia4:= diDetectorPresencia4,
7     diDetectorPresencia5:= diDetectorPresencia5,
3     diDetectorPresencia6:= diDetectorPresencia6,
3     ActivoDetectoPresencia:= ActivoDetectoPresencia,
3     doSirena=> );
1
2 //Control de la sirena de alarma
3 doSirena:=seguridad_p2.doSirena;
1
```

Por último, tenemos que mencionar que, a la hora de la creación de las variables globales, debemos tener en cuenta que las variables relacionadas con las configuraciones de los parámetros de los equipos configurables por el usuario deberían ser de tipo PERSISTENT. Esto es debido para que la aparición de un fallo de tensión en el sistema no conlleve al usuario a volver a poner todos los datos, así con las variables PERSISTENT conseguimos que todo vuelva a funcionar con normalidad. En el código de variables globales del sistema hemos puesto ejemplos de algunas variables que hemos utilizado las cuales deberían ser de este tipo. Lo podemos observar en la siguiente figura el caso concreto de las variables de la programación horaria 1 de la parte de la climatización.

```
177 VAR_GLOBAL PERSISTENT
178     Configuracion_Horaria:ConfiguracionHoraria;
179     Conf_FC1:INT;
180
181     //Variables programación Horaria 1 CLIMATIZACIÓN
182     Mes_Inicio: SINT;
183     Mes_Final: SINT;
184     Dia_Inicio: WORD;
185     Dia_Final: WORD;
186     Dia_Semana_0: BOOL;//Domingo
187     Dia_Semana_1: BOOL;
188     Dia_Semana_2: BOOL;
189     Dia_Semana_3: BOOL;
190     Dia_Semana_4: BOOL;
191     Dia_Semana_5: BOOL;
192     Dia_Semana_6: BOOL;//Sabado
193     Hora_Inicio: WORD;
194     Hora_Final: WORD;
195     Minuto_Inicio: WORD;
196     Minuto_Final: WORD;
197 END_VAR
```

6. VISUALIZACIONES.

Por lo que respecta a la parte de HMI se han creado unas visualizaciones para el control del sistema.

Como nuestro sistema de control va a ser distribuido, tenemos una pantalla principal con la cual podemos acceder a la información de todas las plantas y luego otras con las que controlaremos los sectores cada planta. A continuación, se mostrará la pantalla principal, la cual estará en la sala de control y las pequeñas pantallas que se instalarán en el cuadro de control de cada planta.

6.1 PANTALLA PRINCIPAL DEL SISTEMA.

Desde esta pantalla, que solo se encuentra en la sala de control principal, se puede acceder a las pantallas de las demás plantas.

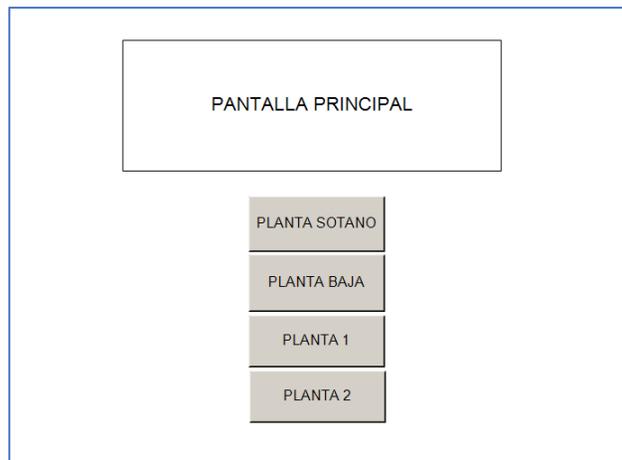


Imagen 14. Visualización principal ubicada en la sala de control.

6.2 PANTALLA PRINCIPAL DE LAS PLANTAS.

La pantalla principal (*Imagen 15*) permite visualizar todas las pantallas con una intuitiva y fácil interfaz de navegación entre las pantallas de cada planta.

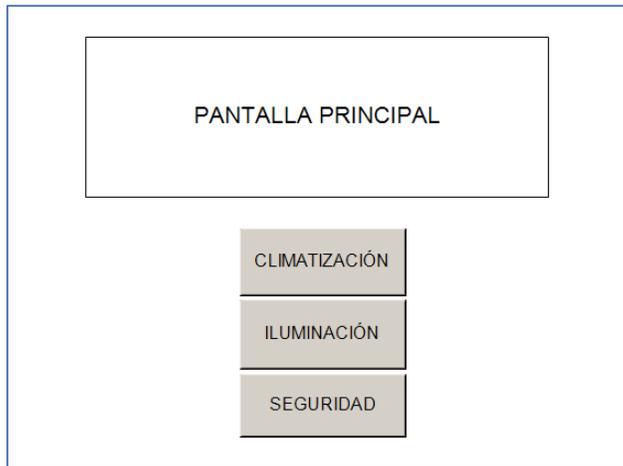


Imagen 15. Visualización principal de cada planta.

6.3 CLIMATIZACIÓN.

Si accedemos al control de la climatización, se mostrará esta pantalla principal (Imagen 16) del sector en cada dispositivo, dónde podemos observar todos los fan coil de la planta a la que accedamos desde la pantalla principal del cuadro de control de la planta baja o en cada planta sus respectivos equipos. Desde aquí podemos visualizar los modos, la temperatura y la función y podemos seleccionas la programación horaria al igual que entrar en la pantalla del fan coil para modificar sus parámetros, los parámetros generales y acceder a la pantalla de alarmas de la planta.



Imagen 16. Visualización de la pantalla de climatización en cada planta.

Con los botones de FAN COIL accedemos a las visualizaciones de cada fan coil y con el de la programación horaria accedemos a la pantalla para su programación.

También tenemos los botones para acceder a las pantallas de las alarmas y los parámetros generales (Imagen 17 e Imagen 18).



Imagen 17. Visualización de la pantalla que muestra las alarmas del sistema de climatización.

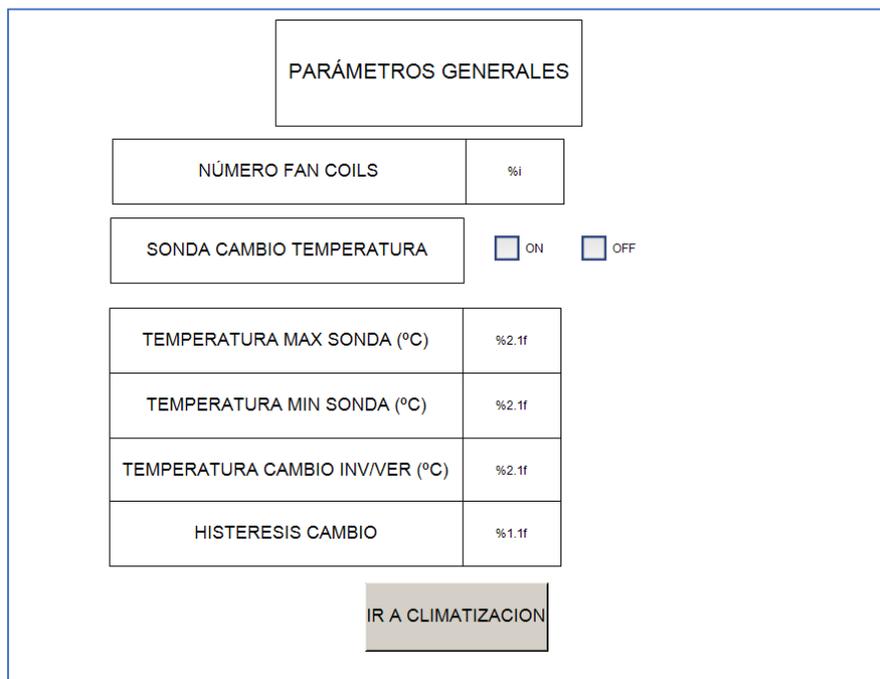


Imagen 18. Visualización de la pantalla que permite modificar los parámetros generales del sistema.

Por otro lado, tenemos la pantalla resultante de pulsar el botón de PROGRAMACIÓN HORARIA, es la siguiente imagen 19, que nos permite acceder a las distintas configuraciones horarias.

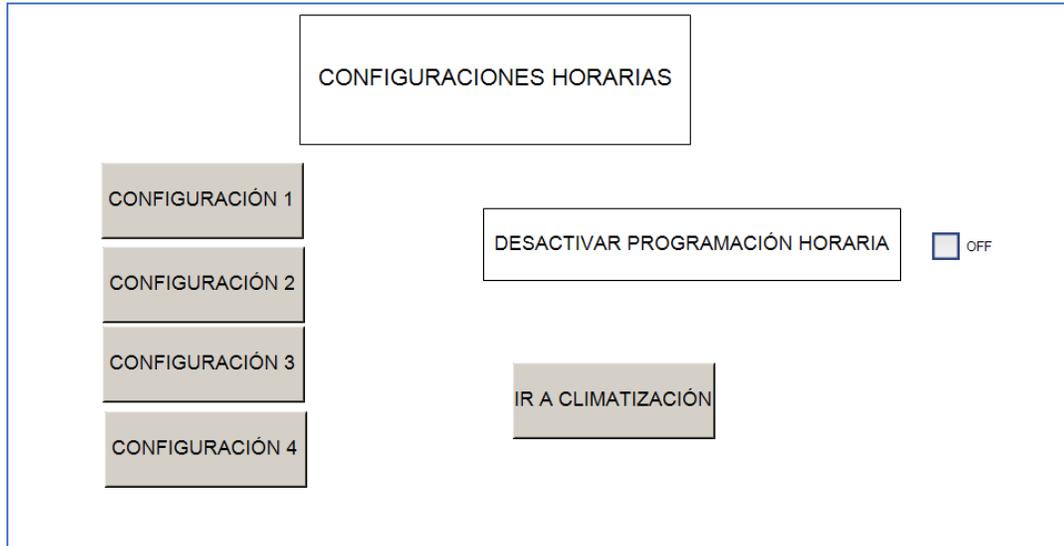


Imagen 19. Visualización para acceder a las configuraciones horarias.

Si entramos a cualquier Fan Coil nos encontraremos con la siguiente visualización (Imagen 20) para configurar este dispositivo con los parámetros adecuados establecidos por los operarios.

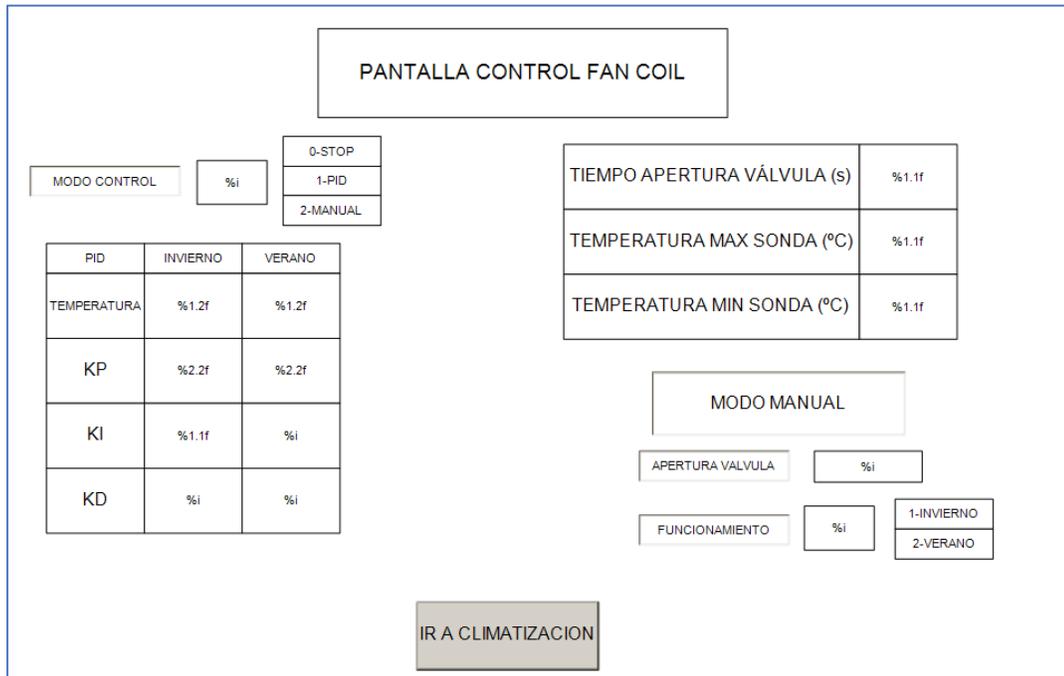


Imagen 20. Visualización de la pantalla de control de los fan coil.

6.4 ILUMINACIÓN.

En la iluminación se ha creado una estructura de visualizaciones distribuidas por planta dónde encontramos todos los grupos existentes en todas las plantas (Grupo1, Grupo2, Grupo3, Grupo4 y Persianas).

Para acceder a estas visualizaciones de los grupos utilizamos la pantalla principal de cada planta (Imagen 15) y al pulsar sobre la iluminación, encontraríamos la pantalla de cada planta. Como por ejemplo la pantalla de la planta 1 (Imagen 21).



Imagen 21. Visualización de la planta 1 de iluminación.

Dónde ubicamos todos los grupos en sus respectivas localizaciones.

Al pulsar en cada de grupo abrimos la visualización para el control de esa estancia como por ejemplo la de la consulta de medicina general 1 que tendría un aspecto como se muestra en la imagen 22.

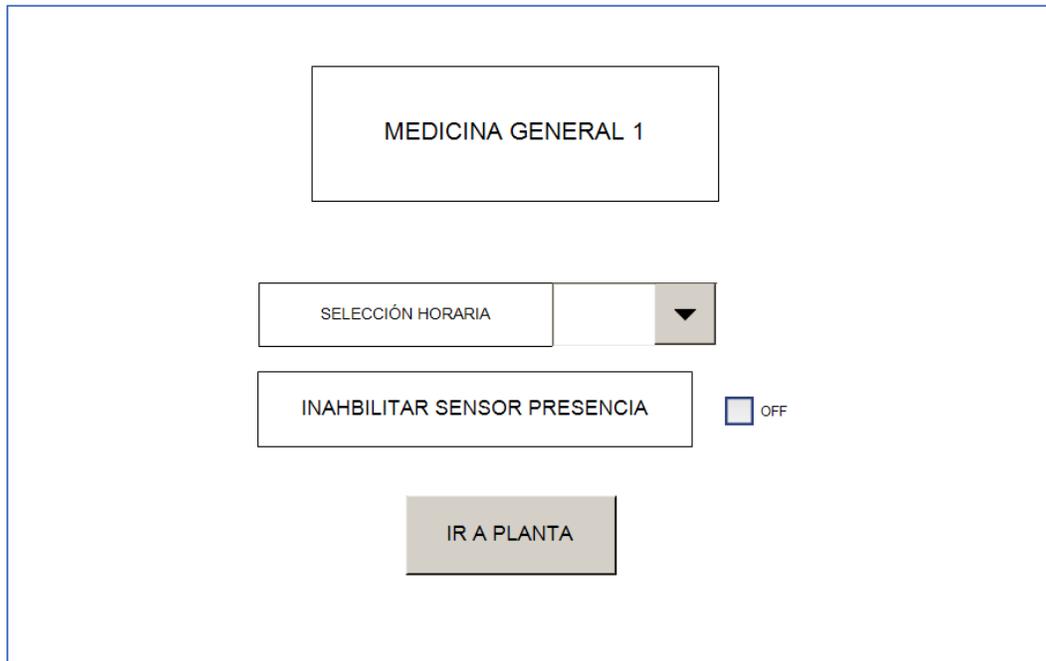


Imagen 22. Visualización de la consulta de medicina general 1 perteneciente al grupo 2 de iluminación.

Cuando accedemos a las persianas nos encontramos con la siguiente pantalla (Imagen 23). Esta es un poco más compleja que las de los grupos de iluminación con luminarias, ya que tiene más parámetros a configurar.

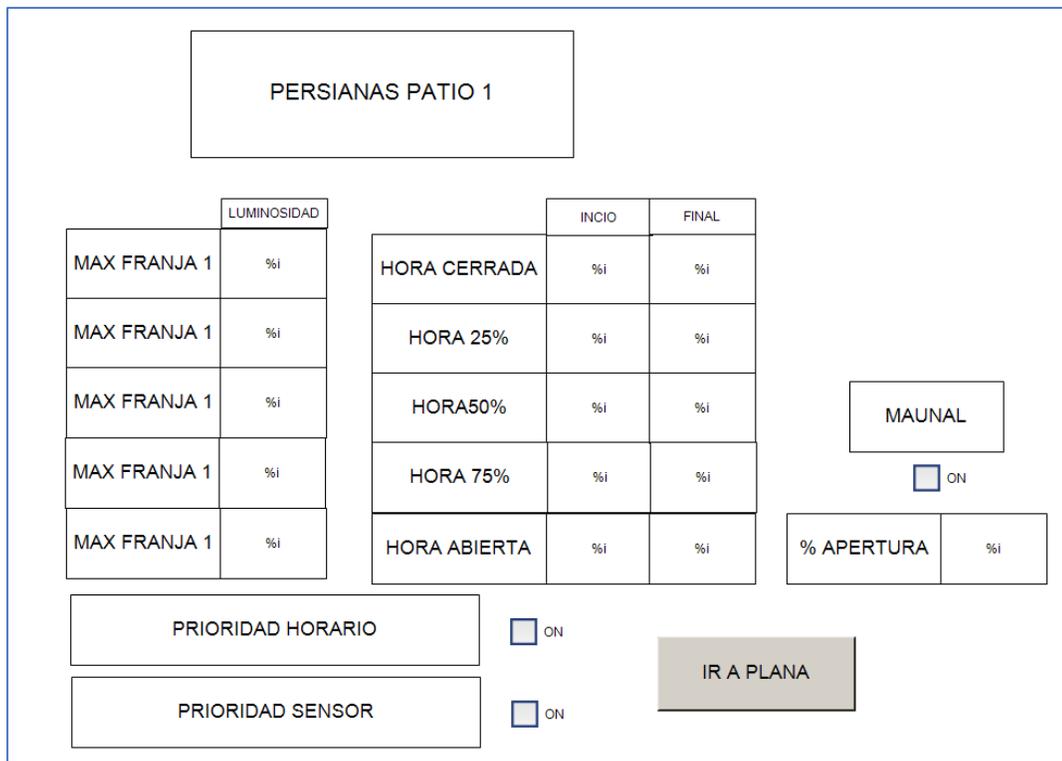


Imagen 23. Visualización del grupo de persianas de la planta 1 en la zona 1.

Las pantallas de la configuración horaria serían las siguientes imágenes (Imagen 24 e Imagen 25).

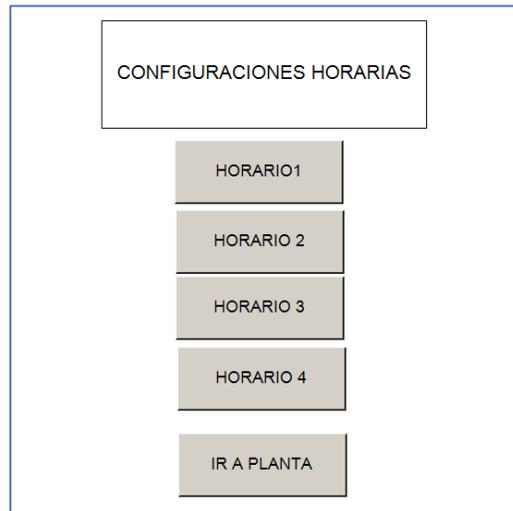


Imagen 24. Visualización de la pantalla para acceder a los horarios para su modificación.

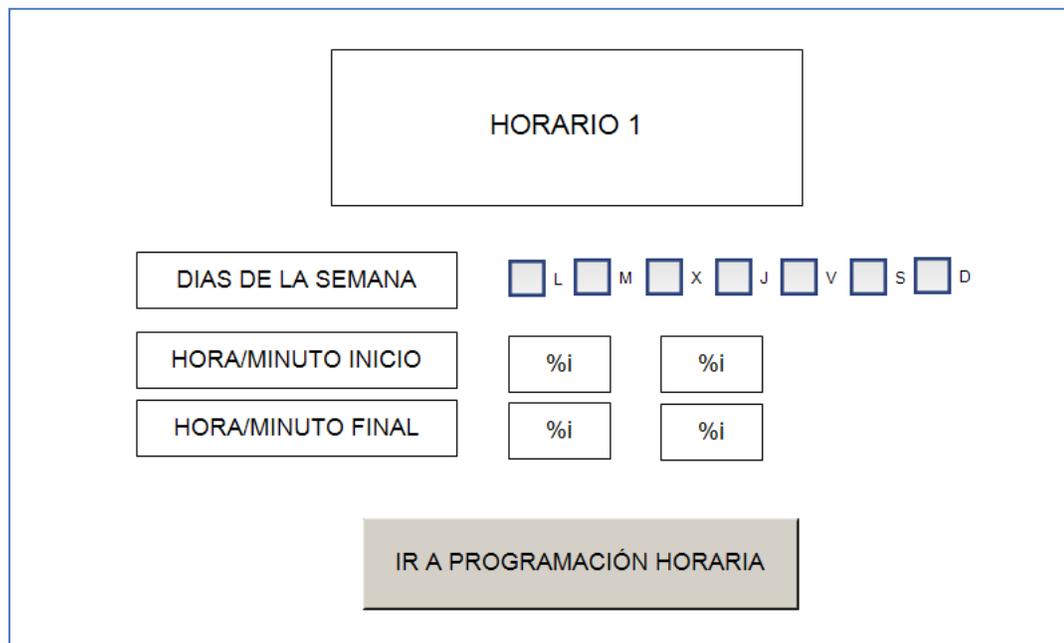


Imagen 25. Visualización de la pantalla de configuración de los horarios del sector de iluminación.

6.5 SEGURIDAD.

Respecto a la seguridad tenemos una visualización muy sencilla (Imagen 26), ya que en lo único que podemos actuar es en la inhabilitación de los sensores de presencia de toda la planta. Además, hemos incluido todos los sensores para visualizar cuando están activos o no y su ubicación.

Para acceder a esta pantalla de control lo hacemos desde la pantalla principal de cada planta (Imagen 15).

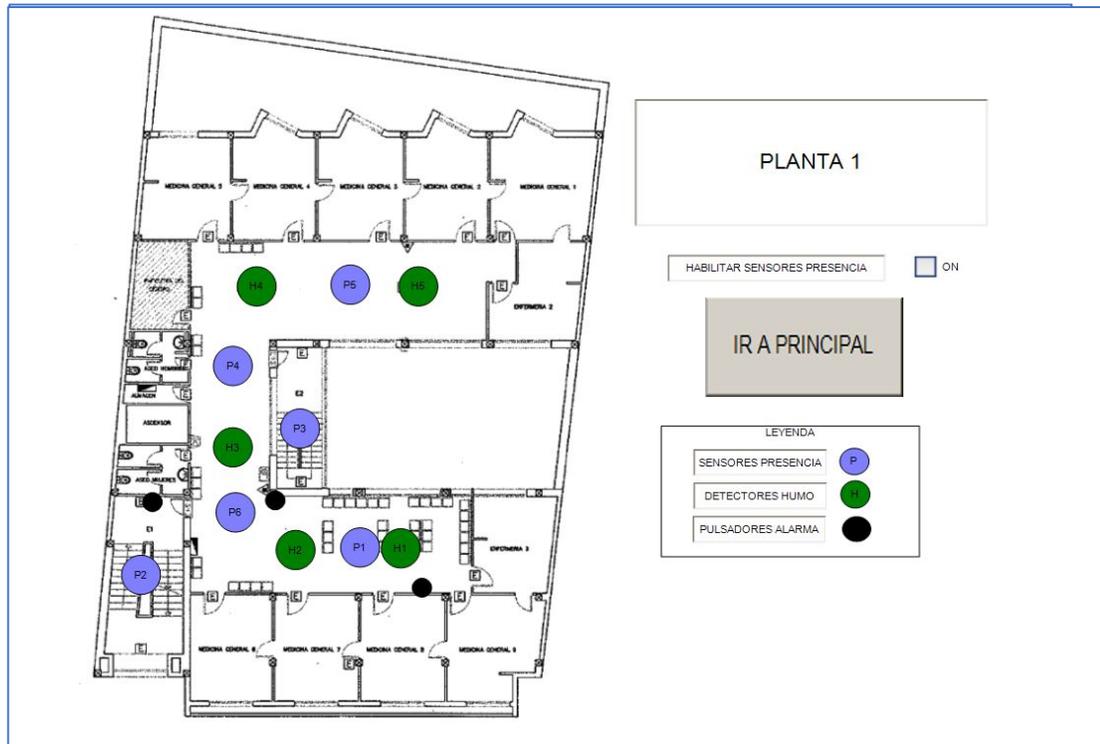


Imagen 26. Visualización de la seguridad de la planta 1.

7. ESTUDIO DE VIABILIDAD ECONÓMICA

En este proyecto no se ha procedido al estudio de viabilidad económica ya que no se disponen de los datos de energía consumida, ni se pueden medir en las instalaciones.

Otra opción podría haber sido obtener una estimación de la energía, basándonos en estudios realizados, pero se ha decidido no realizarlo ya que la complejidad de los sistemas a instalar es alta en comparación a la precisión de los resultados que se obtendrían. No sería un estudio realista de ahorro obtenido con la instalación de los sistemas.

Si que podemos indicar, basándonos en estudios reales de otros edificios relacionados, que el ahorro en porcentaje suele estar entre un 10-15%. (*James Sinopoli, Smart Building Systems for Architects, Owners, and Builders, 2010*)

En otro estudio realizado por Siemens Building Technologies afirman que este ahorro puede aumentar hasta el 25% en costes energéticos.

8. RESUMEN DEL PRESUPUESTO

A continuación, se mostrará el presupuesto reducido del coste del proyecto.

PRESUPUESTO TOTAL	COSTE (€)
PEM	16633,61
20% CARGAS FISCALES Y GENERALES	3326,722
6% BENEFICIO INDUSTRIAL	998,0166
PRESUPUESTO DE EJECUCIÓN POR CONTRATA (PEC)	20958,3486
7% PROYECTO Y DIRECCIÓN DE OBRA	1467,0844
TOTAL	22425,43
TOTAL + 21% IVA	27134,77

Tabla 13. Resumen del presupuesto del proyecto.

Por lo tanto, el presupuesto total asciende a veintisiete mil ciento treinta y cuatro Euros y setenta y siete céntimos.

9.REFERENCIAS

1. RFC 7452, “Architectural Considerations in Smart Object Networking” (Marzo de 2015), (<https://tools.ietf.org/html/rfc7452>)
2. “Overview of the Internet of Things.” ITU, 15 de junio de 2012.
(<https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=Y.2060>)
3. Protocolo MQTT. (<https://en.wikipedia.org/wiki/MQTT>)
- 4.” Automatización Industrial”, UJI 2010, Roberto Sanchís Llopis, Julio Ariel Romero Pérez, Carlos Vicente Ariño.
5. Revista BECKHOFF. Integrated Building Automation Solutions.
6. Revista BECKHOFF. PC-based Control for Integrated Buiding Automation.
7. Revista BECKHOFF. PC-based Control from Beckhoff for Storage and Show Technology.
8. Revista BECKHOFF. EtherCAT- el bus de campo Ethernet.
9. Manal BECKHOFF. Comunicación entre TwinCAT y Microsoft Azure IoT Hub
10. “Smart Buildings systems for Architecs, Owners and builders”, Elsevier 2010, James Sinopoli.
11. Siemens Building Technologies

ANEXOS

ÍNDICE ANEXOS

MÉTODO DE LAS JERARQUIAS ANALÍTICAS.	90
ESQUEMAS CENTRO DE SALUD.	92
ZONAS DEL CENTRO DE SALUD.	97
CÓDIGO DEL PROGRAMA.....	103

1. MÉTODO DE LAS JERARQUIAS ANALÍTICAS.

1-DATOS DE PARTIDA		
IMPORTANCIA DEL CRITERIO		
	1	1
	PRECIO	CARACTERÍSTICAS
EL3423	171,84	*
EL3443	211,94	**
EL3453	279,24	***
EL3483	121,72	****

Tabla 14. Datos de partida del método de las jerarquías analíticas.

2- NORMALIZACIÓN DE LA VALORACIÓN DE ALTERNATIVAS		
CRITERIOS NORMALIZADOS		
	1	1
	PRECIO	CARACTERÍSTICAS
EL3423	0,67	0,67
EL3443	0,33	1
EL3453	0	0,67
EL3483	1	0

Tabla 15. Normalización de la valoración de las alternativas

3- MÉTODO DE LAS JERARQUÍAS ANALÍTICAS				
	PRECIO	CARACTERÍSTICAS	TOTAL	NORMALIZADO
PRECIO	1	1	2	0,50
CARACTERÍSTICAS	1	1	2	0,50
			4	1,00

Tabla 16. Método de las jerarquías analíticas.

4- DECISIÓN			
	PRECIO	CARACTERÍSTICAS	TOTAL
EL3423	0,33	0,33	0,67
EL3443	0,17	0,50	0,67
EL3453	0,00	0,33	0,33
EL3483	0,50	0,00	0,50

Tabla 17. Decisión del método de las jerarquías analíticas.

*Características EL3423	**Características EL3443	***Características EL3453	****Características EL3483
0x1A00 (L1 Status)	0x1A00 (L1 Status)	0x1A00 (L1 Status)	0x1A00 (L1 Status)
0x1A03 (L1 Energy)	0x1A01 (L1 Basic)	0x1A01 (L1 Basic)	
	0x1A02 (L1 Power)	0x1A02 (L1 Power)	0x1A0A (L2 Status)
0x1A0A (L2 Status)	0x1A04 (L1 Timing)		
0x1A0D (L2 Energy)		0x1A0C (L2 Status)	0x1A14 (L3 Status)
	0x1A0A (L2 Status)	0x1A0D (L2 Basic)	
0x1A14 (L3 Status)	0x1A0B (L2 Basic)	0x1A0E (L2 Power)	0x1A1E (Total Total Status)
0x1A17 (L3 Energy)	0x1A0C (L2 Power)		0x1A20 (Total Total Advanced)
	0x1A0E (L2 Timing)	0x1A18 (L3 Status)	
0x1A1E (Total Total Status)		0x1A19 (L3 Basic)	
0x1A21 (Total Total Active)	0x1A14 (L3 Status)	0x1A1A (L3 Power)	
0x1A22 (Total Total Apparent)	0x1A15 (L3 Basic)		
	0x1A16 (L3 Power)	0x1A24 (Total Status)	
0x1A23 (Total Total Reactive)	0x1A18 (L3 Timing)	0x1A25 (Total Basic)	
	0x1A1E (Total Total Status)		
	0x1A1F (Total Total Basic)		
	0x1A21 (Total Total Active)		
	0x1A24 (Total Total L-L Voltage)		

Tabla 18. Características de las tarjetas a estudiar en el método.

2. ESQUEMAS CENTRO DE SALUD.

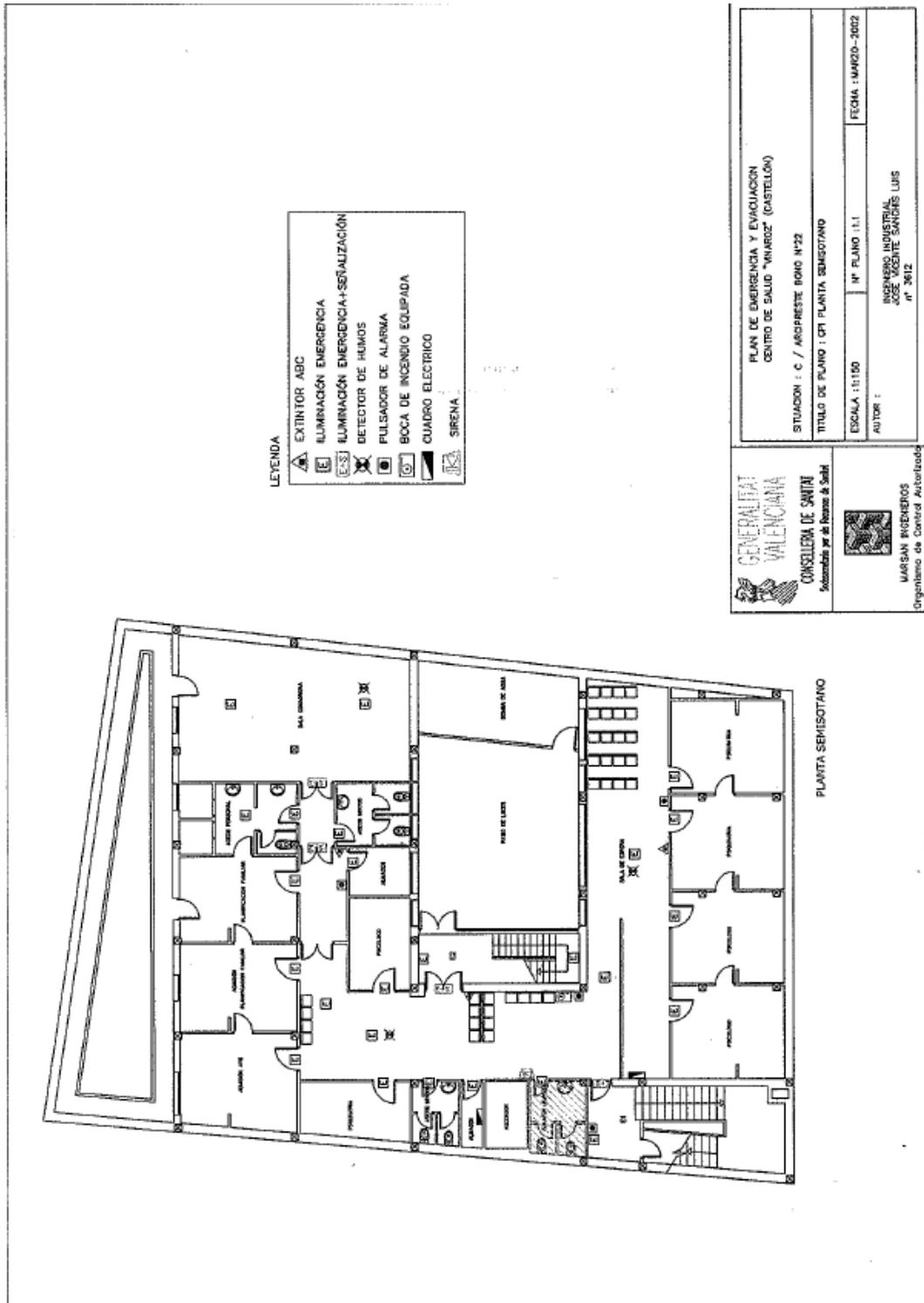


Imagen 27. Esquema planta sótano del centro de salud de Vinaroz.

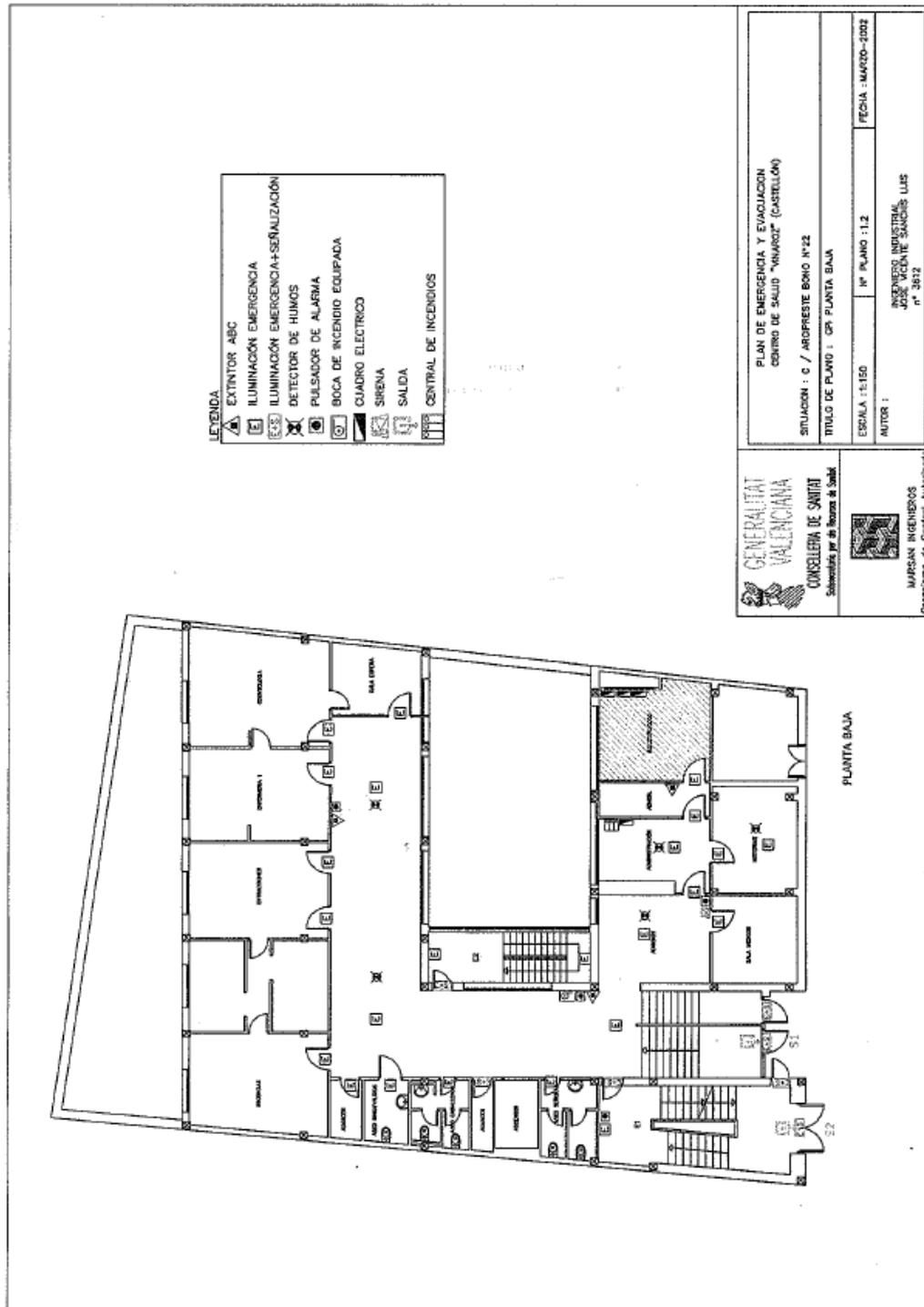


Imagen 28. Esquema planta baja del centro de salud de Vinaroz.

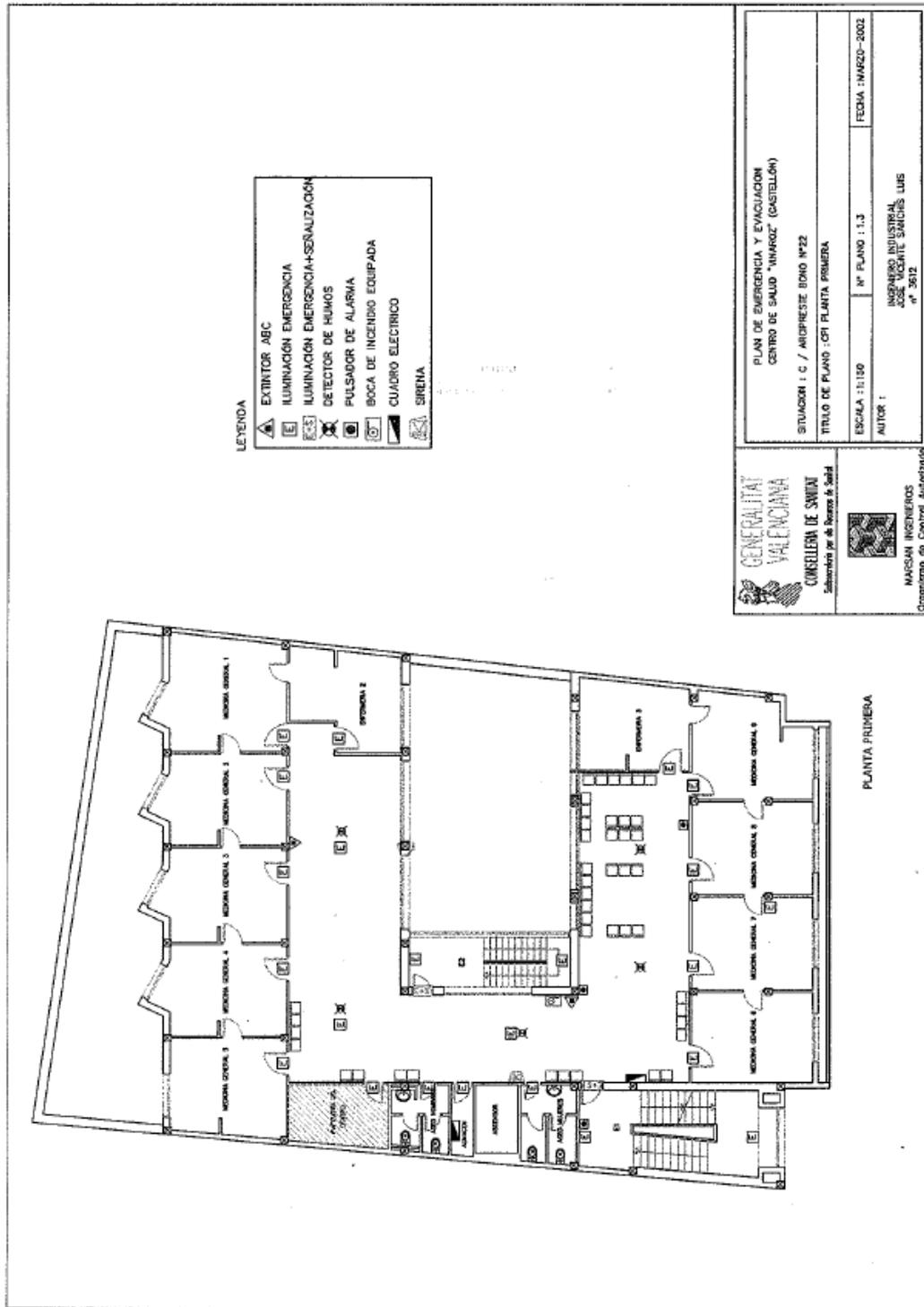


Imagen 29. Esquema planta primera del centro de salud de Vinaroz.

3. ZONAS DEL CENTRO DE SALUD.

<i>DISTRIBUCIÓN ZONAS ILUMINACIÓN CENTRO SALUD VINARÒS</i>							
		<i>LUMINARIAS DO</i>	<i>S. PRESENCIA DI</i>	<i>SENSORES LUZ AI</i>	<i>PERSIANAS AO</i>	<i>GRUPO CONTROL</i>	<i>CUADRO ELECTRICO</i>
<i>PLANTA 2ª</i>							
<i>IL.P2.Z1</i>	<i>ILUMINACIÓN ALA INTERIOR ZONA 1: SALA JUNTAS, ADMINISTACIÓN DIRECCIÓN, INSPECCIÓN + PASILLO INTERIOR</i>	<i>7 + 2</i>				<i>GRUPO 1</i>	<i>ALMACEN</i>
<i>IL.P2.ZP.Z1</i>	<i>ILUMINACIÓN SALAS ZONA PATIO ALA INTERIOR: REGISTRO, ALMACEN, FARMACIA, ARCHIVO</i>	<i>4</i>	<i>4</i>	<i>4</i>		<i>GRUPO 4</i>	<i>ALMACEN</i>
<i>IL.P2.PA</i>	<i>ILUMINACIÓN PASILLO P2 + ESCALERAS PATIO</i>	<i>4 + 1</i>		<i>4</i>		<i>GRUPO 3</i>	<i>PASILLO</i>
<i>IL.P2.ZP.Z2</i>	<i>ILUMINACIÓN SALAS ZONA PATIO ALA EXTERIOR: PEDIATRIA 2</i>	<i>1</i>	<i>1</i>	<i>1</i>		<i>GRUPO4</i>	<i>ALMACEN</i>

<i>IL.P2.Z2</i>	<i>ILUMINACIÓN ALA FACHADA EXTERIOR ZONA 2: ENFERMERIAS, T. SOCIAL</i>	3	3	3		<i>GRUPO 4</i>	<i>ALMACEN</i>
<i>IL.ESCALERAS</i>	<i>ILUMINACIÓN ESCALERAS PRINCIPALES</i>	2		1		<i>GRUPO 3</i>	<i>PASILLO</i>
<i>IL.P2.AUXILIARES</i>	<i>ILUMINACIÓN ZONAS AUXILIARES: ALMACEN, ASEOS PÚBLICOS</i>	3				<i>GRUPO 0</i>	<i>ALMACEN</i>
<i>PE.P2.PT</i>	<i>LÁMINAS PERSIANA PATIO P2 (1) (2) (3)</i>			3	3		<i>ALMACEN</i>
<i>PE.P2.FE</i>	<i>LÁMINAS PERSIANA FACHADA EXTERIOR P2 (4) (5)</i>			2	2		<i>ALMACEN</i>
<i>PLANTA 1ª</i>							
<i>IL.P1.Z1</i>	<i>ILUMINACIÓN ALA INTERIOR ZONA 1: CONSULTAS MEDICINA GENERAL 1-2-3-4-5</i>	5	5			<i>GRUPO 2</i>	<i>ALMACEN</i>
<i>IL.P1.ZP.Z1</i>	<i>ILUMINACIÓN SALAS ZONA PATIO ALA INTERIOR : ENFERMERIA 2</i>	1		1		<i>GRUPO 3</i>	<i>ALMACEN</i>
<i>IL.P1.PA</i>	<i>ILUMINACIÓN PASILLO P1 + ESCALERAS PATIO</i>	5 + 1		4		<i>GRUPO 3</i>	<i>PASILLO</i>
<i>IL.P1.ZP.Z2</i>	<i>ILUMINACIÓN SALAS ZONA PATIO ALA EXTERIOR : ENFERMERIA 3</i>	1	1	1		<i>GRUPO 4</i>	<i>ALMACEN</i>

<i>IL.PI.Z2</i>	<i>ILUMINACIÓN ALA FACHADA EXTERIOR ZONA 2: CONSULTAS MEDICINA GENERAL 6-7-8-9</i>	<i>4</i>	<i>4</i>	<i>4</i>		<i>GRUPO 4</i>	<i>ALMACEN</i>
<i>IL.PI.ESCALERAS</i>	<i>ILUMINACIÓN ESCALERAS PRINCIPALES</i>	<i>2</i>		<i>1</i>		<i>GRUPO 3</i>	<i>PASILLO</i>
<i>IL.PI.AUXILIARES</i>	<i>ILUMINACIÓN ZONAS AUXILIARES: ASEOS PÚBLICOS, ALMACEN, SALA CENTRO</i>	<i>4</i>				<i>GRUPO 0</i>	<i>ALMACEN</i>
<i>PE.PI.PT</i>	<i>LÁMINAS PERSIANA PATIO P2 (1) (2) (3)</i>			<i>3</i>	<i>3</i>		<i>ALMACEN</i>
<i>PE.PI.FE</i>	<i>LÁMINAS PERSIANA FACHADA EXTERIOR P2 (4) (5)</i>			<i>2</i>	<i>2</i>		<i>ALMACEN</i>
<i>PLANTA BAJA</i>							
<i>IL.PB.ZI</i>	<i>ILUMINACIÓN ALA INTERIOR ZONA 1: URGENCIAS, EXTRACCIONES, ENFERMERIA 1, ODONTOLOGIA</i>	<i>4</i>	<i>4</i>			<i>GRUPO 2</i>	
<i>IL.PB.ZP.ZI</i>	<i>ILUMINACIÓN SALAS ZONA PATIO ALA INTERIOR: SALA ESPERA</i>	<i>1</i>		<i>1</i>		<i>GRUPO 3</i>	<i>S.CONTROL</i>
<i>IL.PB.PA</i>	<i>ILUMINACIÓN PASILLO PB + ESCALERA PATIO</i>	<i>4 + 1</i>		<i>4</i>		<i>GRUPO 3</i>	<i>S.CONTROL</i>
<i>IL.PB.ZP.Z2</i>	<i>ILUMINACIÓN SALAS ZONA PATIO ALA EXTERIOR :ADMINISIÓN, ADMINISTRACIÓN</i>	<i>3</i>	<i>3</i>	<i>3</i>		<i>GRUPO 4</i>	<i>S.CONTROL</i>

<i>IL.PB.Z2</i>	<i>ILUMINACIÓN ALA FACHADA EXTERIOR ZONA 2: SALA MÉDICOS, HISTORIALES...</i>	3	3	3		GRUPO 4	S.CONTROL
<i>IL.PB.ESCALERAS</i>	<i>ILUMINACIÓN ESCALERAS PRINCIPALES</i>	2		1		GRUPO 3	S.CONTROL
<i>IL.PB.HL</i>	<i>ILUMINACIÓN ENTRADA CS</i>	1				GRUPO 1	S.CONTROL
<i>IL.PB.AUX</i>	<i>ILUMINACIÓN ZONAS AUXILIARES: ALMACEN, ASEOS PÚBLICOS, SALA DE CONTROL</i>	4				GRUPO 0	S.CONTROL
<i>PE.PB.PT</i>	<i>LAMINAS PERSIANA PATIO P2 (1) (2) (3)</i>			3	3		S.CONTROL
PLANTA SÓTANO							
<i>IL.PS.Z1</i>	<i>ILUMINACIÓN ALA INTERIOR ZONA 1: ADMISIÓN ATS, ADMISIÓN PLANIF. FAMILIAR, PLANIF. FAMILIAR</i>	3	3			GRUPO 2	ALMACEN
<i>IL.PS.ZP.Z1</i>	<i>ILUMINACIÓN SALAS ZONA PATIO ALA INTERIOR: PSICÓLOGO, SALA COMADRONA</i>	3	2	2		GRUPO 4	ALMACEN
<i>IL.PS.PA</i>	<i>ILUMINACIÓN PASILLO PS + ESCALERAS PATIO + SALAS ESPERA</i>	5 + 1		4		GRUPO 3	PASILLO
<i>IL.PS.Z2</i>	<i>ILUMINACIÓN ALA FACHADA EXTERIOR, ZONA 2: CONSULTAS PSICÓLOGOS/PSIQUIRAS</i>	4	4	4		GRUPO 4	ALMACEN

<i>IL.PS.ESCALERAS</i>	<i>ILUMINACIÓN ESCALERAS PRINCIPALES</i>	<i>2</i>		<i>1</i>		<i>GRUPO 3</i>	<i>PASILLO</i>
<i>IL.PS.AUX</i>	<i>ILUMINACIÓN ZONAS AUXILIARES: ALMACENES, ASEOS PERSONAL, ASEOS MIXTOS,BOMBA AGUA...</i>	<i>4</i>				<i>GRUPO 0</i>	<i>ALMACEN</i>
<i>PE.PS.PT</i>	<i>LÁMINAS PERSIANA PATIO P2 (1) (2) (3)</i>			<i>3</i>	<i>3</i>		<i>ALMACEN</i>

Tabla 19. Distribución zonas iluminación centro salud Vinaoz.

DISTRIBUCIÓN ZONAS CLIMATIZACIÓN Y CLIMATIZADORES CENTRO SALUD VINARÒS		
PLANTA 2ª		
CL.P2.Z1	CLIMATIZADOR ALA INTERIOR, ZONA 1 PLANTA 2ª	ALMACEN
CL.P2.PA	CLIMATIZADOR PASILLO PLANTA 2ª	ALMACEN
CL.P2.Z2	CLIMATIZADOR ALA FACHADA EXTERIOR, ZONA 2, PLANTA 2ª	ALMACEN
EX.P2	EXTRACTOR P2	ALMACEN
PLANTA 1ª		
CL.P1.Z1	CLIMATIZADOR ALA INTERIOR, ZONA 1 PLANTA 1ª	ALMACEN
CL.P1.PA	CLIMATIZADOR PASILLO PLANTA 1ª	ALMACEN
CL.P1.Z2	CLIMATIZADOR ALA FACHADA EXTERIOR, ZONA 2, PLANTA 1ª	ALMACEN
EX.P1	EXTRACTOR P1	ALMACEN
PLANTA BAJA		
CL.PB.Z1	CLIMATIZADOR ALA INTERIOR, ZONA 1, PLANTA BAJA	S.CONTROL
CL.PB.Z2	CLIMATIZADOR ALA FACHADA EXTERIOR Y PASILLO, ZONA 2, PLANTA BAJA	S.CONTROL
EX.PB	EXTRACTOR PB	S.CONTROL
PLANTA SÓTANO		
CL.PS.Z1	CLIMATIZADOR ALA INTERIOR, ZONA 1, PLANTA SOTANO	ALMACEN
CL.PS.PA	CLIMATIZADOR PASILLO PLANTA SOTANO	ALMACEN
CL.PS.Z2	CLIMATIZADOR ALA FACHADA EXTERIOR, ZONA 2, PLANTA SOTANO	ALMACEN
EX.PS	EXTRACTOR PS	ALMACEN

Tabla 20. Distribución zonas climatización y climatizadores centro salud Vinaroz

DISTRIBUCIÓN SEGURIDAD CENTRO SALUD VINARÒS			
		ENT.DIG (DI)	SAL.DIG (DO)
PLANTA 2ª			
SG.DH.P2	DETECTORES HUMOS P2	5	
SG.PALM.P2	PULSADORES ALARMA P2	1	
SG.SIRENA.P2	SIRENAS		1
<i>SG.DP.P2</i>	<i>DETECTORES PRESENCIA (PASILLO + ESCALERAS)</i>	4 + 2	
PLANTA 1ª			
SG.DH.P1	DETECTORES HUMOS	5	
SG.PALM.P1	PULSADORES ALARMA	1	
SG.SIRENA.P1	SIRENAS		1
<i>SG.DP.P1</i>	<i>DETECTORES PRESENCIA (PASILLO + ESCALERAS)</i>	4+2	
PLANTA BAJA			
SG.DH.PB	DETECTORES HUMOS P2	4	
SG.PALM.PB	PULSADORES ALARMA P2	1	
SG.SIRENA.PB	SIRENAS		1
<i>SG.DP.PB</i>	<i>DETECTORES PRESENCIA (PASILLO + ESCALERAS)V</i>	4+2	

PLANTA SOTANO			
SG.DH.PS	DETECTORES HUMOS P2	3	
SG.PALM.PS	PULSADORES ALARMA P2	1	
SG.SIRENA.PS	SIRENA		1
SG.DP.PS	DETECTORES PRESENCIA (PASILLO + ESCALERAS)	4 + 2	

Tabla 21. Distribución seguridad centro salud Vinaroz.

4. CÓDIGO DEL PROGRAMA

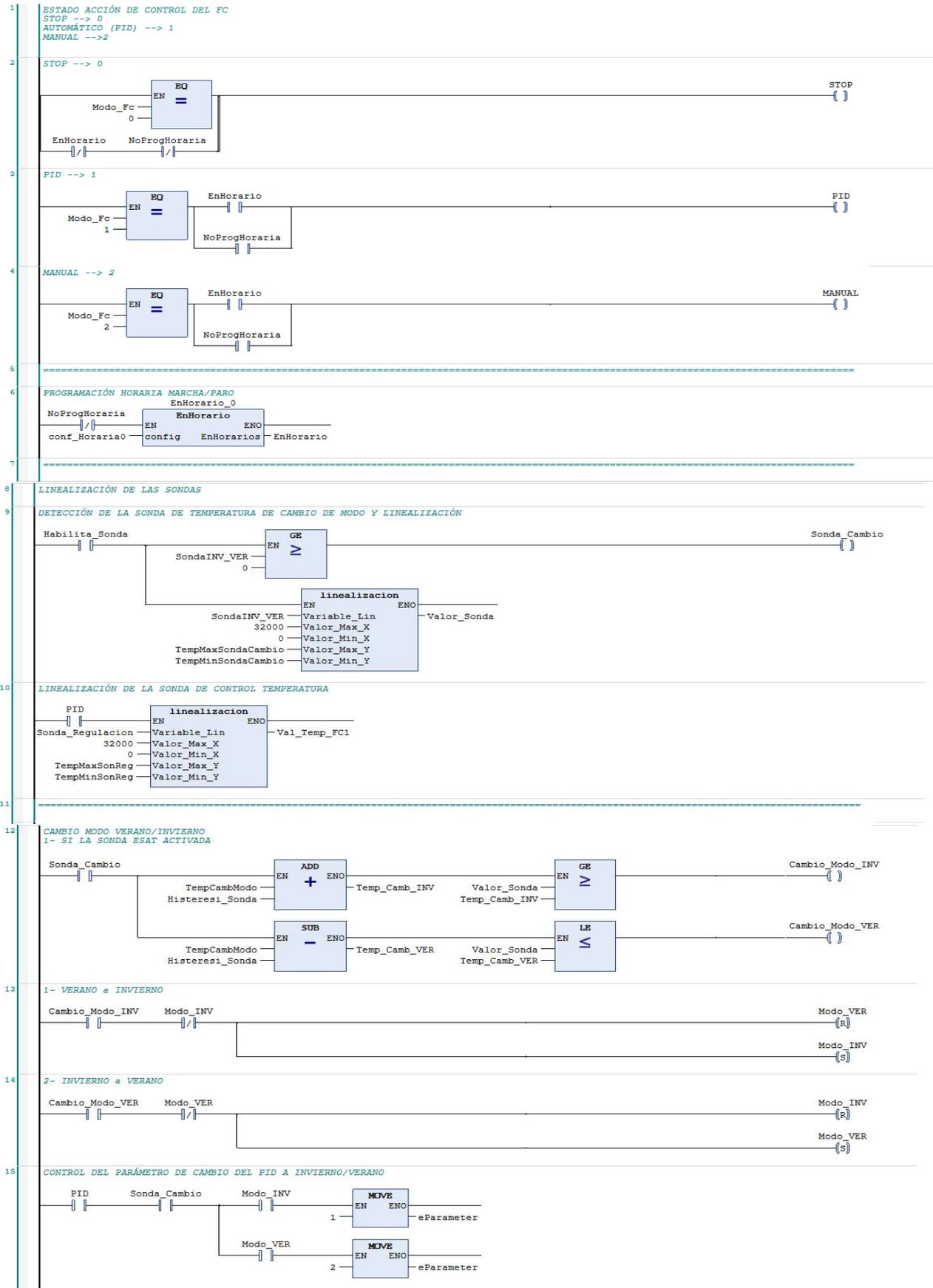
POU_CLIMATIZACION

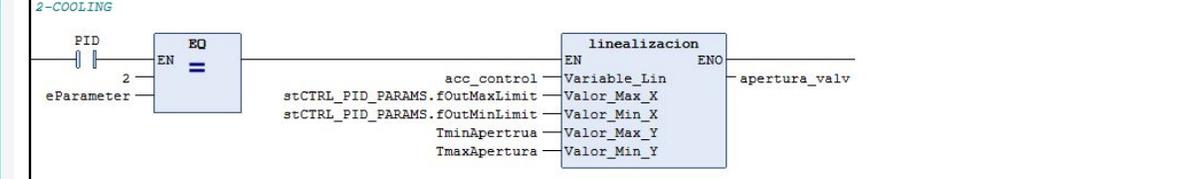
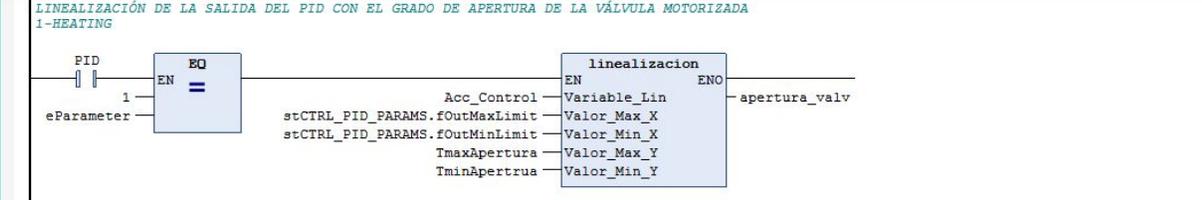
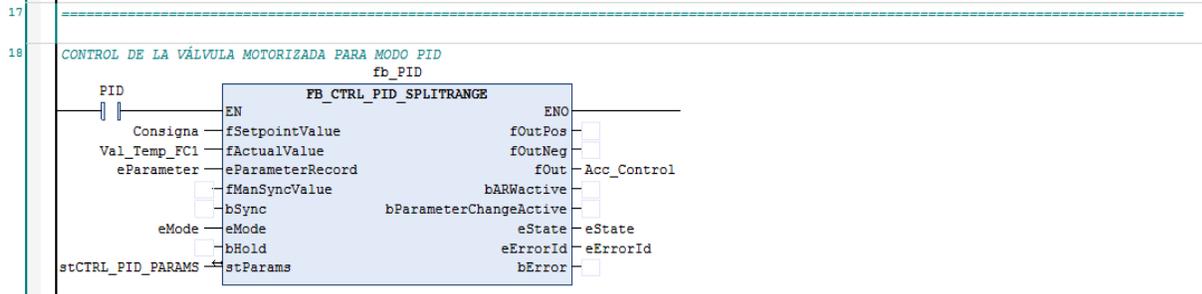
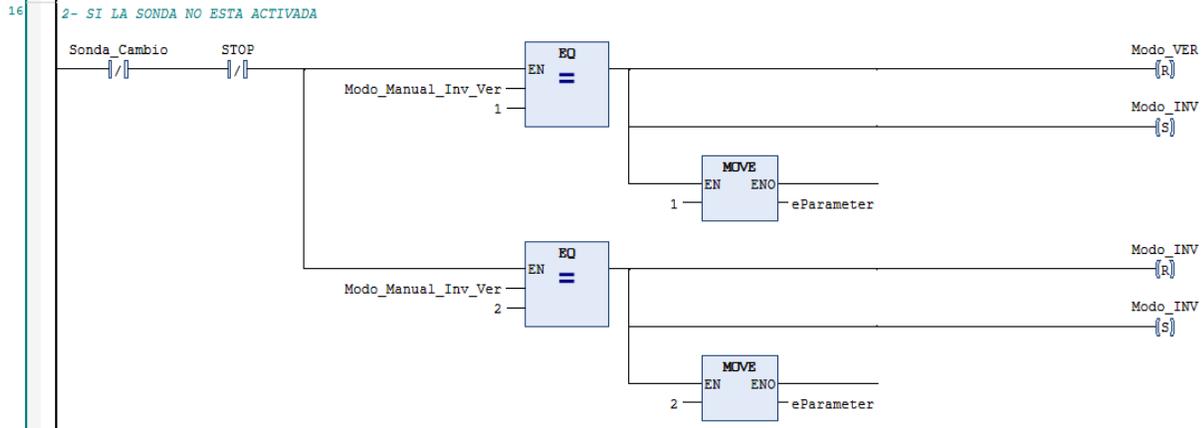
```
1 PROGRAM CLIMATIZACION
2 VAR_INPUT
3
4 //Variables de la Sonda de Temperatura de Regulación
5 TempMaxSonReg:LREAL:=45;
6 TempMinSonReg:LREAL:=0;
7 Sonda_Regulacion:INT;//Analog Input Sonda
8
9 //Variables para el control de la válvula motorizada
10 doFC_VMABRIR:BOOL;
11 doFC_VMCERRAR:BOOL;
12 doFC_MARCHA:BOOL;
13 diFC_EnMarcha:BOOL;
14
15 //Parámetros Sonda Cambio Verano/Invierno
16 SondaINV_VER:INT; //Analog Input Sonda
17 TempMaxSondaCambio: REAL;
18 TempMinSondaCambio: REAL;
19 Habilita_Sonda: BOOL;
20 TempCambModo:REAL;
21 Histeresi_Sonda:REAL;
22 Modo_Manual_Inv_Ver:INT;
23
24 //Variables de entrada del PID
25 Consigna: LREAL;
26
27 //Variable modos del Fan Coil
28 Modo_Fc: INT;
29
30 //Variable para la generación de pulsos para la apertura de la válvula motorizada
31 TApertura:REAL;
32
33 //Variable apertura de la válvula en el modo manual
34 Apertura_Deseada: INT;
35
36 //Auxiliar para saber si la programación está activada
37 NoProgHoraria:BOOL;
38 //Estructura para el control de la Programación Horaria
39 conf_Horaria0:ConfiguracionHoraria;
40
41 //Estructura para el control del PID
42 stCTRL_PID_PARAMS: ST_CTRL_PID_SPLITRANGE_PARAMS;
43
44 END_VAR
45
46 VAR_OUTPUT
47 HAY_ALM: BOOL;
48 END_VAR
49
50 VAR
51 //Parametros de control del PID
52 fb_PID: FB_CTRL_PID_SPLITRANGE;
53 eMode: E_CTRL_MODE:= eCTRL_MODE_ACTIVE;
54 eState: E_CTRL_STATE;
55 eErrorId: E_CTRL_ERRORCODES;
56 eParameter: E_CTRL_PARAMETER_RECORD:=eCTRL_PARAMETER_RECORD_HEATING;
57
58 //Variables de salida del PID
59 Acc_Control: LREAL;
60
61
```

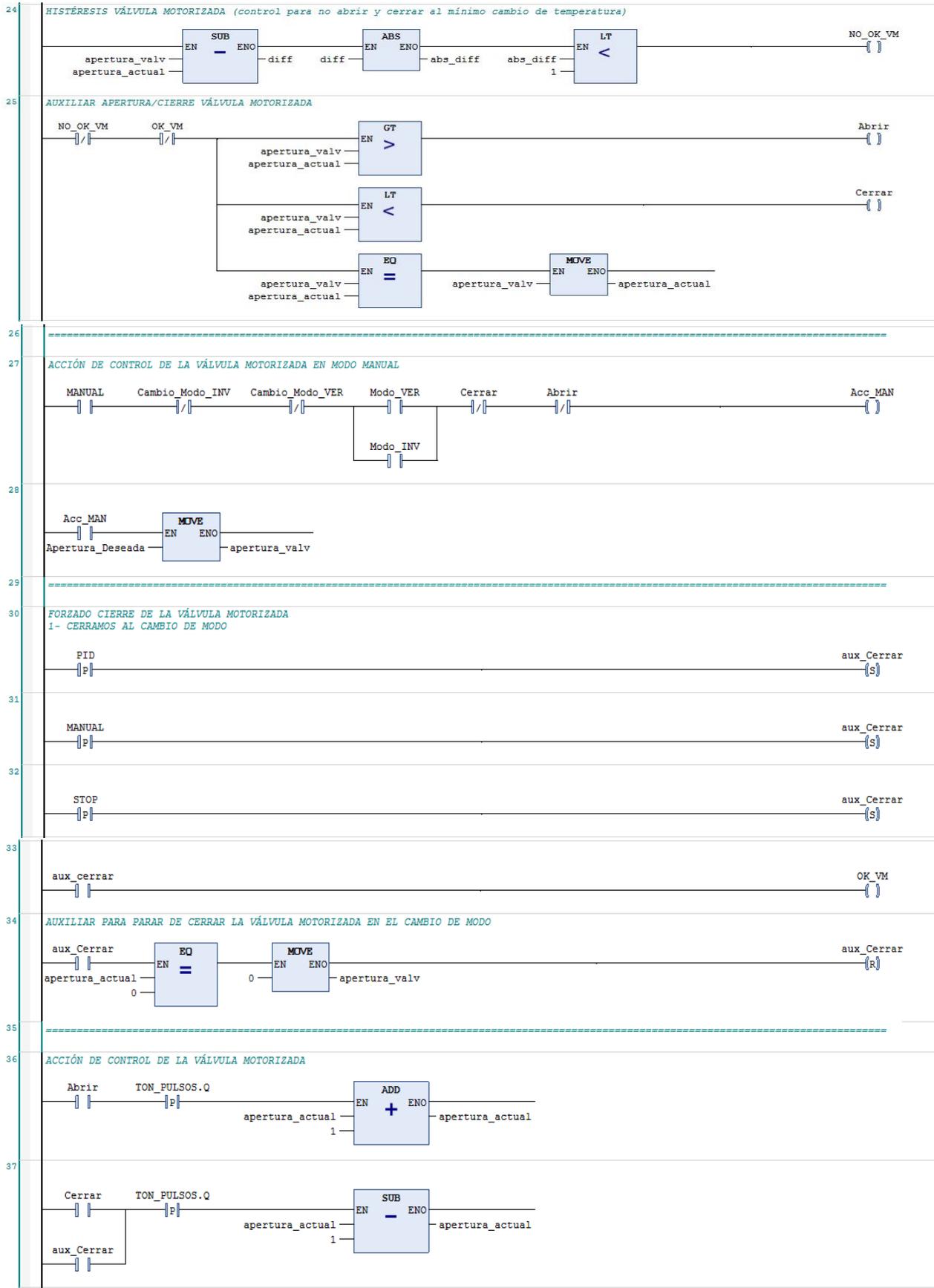
```

62 //Variable para la generación de pulsos para la apertura de la válvula motorizada
63 TPulso: TIME:=T#100MS;
64 TPulso1: REAL;
65 TPulso2:INT;
66 TON_PULSOS: TON;
67 TmaxApertura: REAL:=100;
68 TminApertrua: REAL:=0;
69
70 //Variables para el control de la válvula motorizada
71 Abrir: BOOL;
72 Cerrar: BOOL;
73 aux_Cerrar: BOOL;
74 Acc_MAN: BOOL;
75 apertura_valv: LREAL;
76 apertura_actual: LREAL:=0;
77 Val_Temp_FCl: LREAL; //Valor linealizado de la sonda analógica de temperatura
78
79 //Variables histeresis
80 diff: LREAL;
81 abs_diff: LREAL;
82 OK_VM: BOOL;
83 NO_OK_VM: BOOL;
84
85 //Variables Sonda cambio Inviern/Verano
86 Sonda_Cambio: BOOL;
87 Temp_Camb_INV: REAL;
88 Temp_Camb_VER: REAL;
89 Valor_Sonda: LREAL;
90 Cambio_Modo_INV: BOOL;
91 Cambio_Modo_VER: BOOL;
92 Modo_INV: BOOL;
93 Modo_VER: BOOL;
94
95
96 //Variables modos del Fan Coil
97 STOP: BOOL;
98 MANUAL: BOOL;
99 PID: BOOL;
100
101
102 //Variable para el control horario
103 EnHorario: BOOL;
104 EnHorario_0: EnHorario; //Función para evaluar los horarios
105
106 //Variables alarmas
107 Alm_FCl_No_EnMarcha: BOOL;
108 Alm_NoExtractorEnMarcha: BOOL;
109 Alm_Fallo_Termico: BOOL;
110 ALM: TON;
111
112 END_VAR

```









POU_CNFIGURACIONHORARIA

```

1  FUNCTION BLOCK CONFIGURACION_HORARIA
2  VAR_INPUT
3      Mes_Inicio: SINT;
4      Mes_Final: SINT;
5      Dia_Inicio: WORD;
6      Dia_Final: WORD;
7      Dia_Semana_0: BOOL; //Domingo
8      Dia_Semana_1: BOOL;
9      Dia_Semana_2: BOOL;
10     Dia_Semana_3: BOOL;
11     Dia_Semana_4: BOOL;
12     Dia_Semana_5: BOOL;
13     Dia_Semana_6: BOOL; //Sabado
14     Hora_Inicio: WORD;
15     Hora_Final: WORD;
16     Minuto_Inicio: WORD;
17     Minuto_Final: WORD;
18 END_VAR
19 VAR_OUTPUT
20     Config1:ConfiguracionHoraria;
21 END_VAR
22 VAR
23     i: SINT;
24     Dias_Semana: DWORD;
25 END_VAR
26
27 Dias_Semana:=0;
28 i:=0;
29
30     Config1.Mes_Inicio:=Mes_Inicio;
31     Config1.Mes_Final:=Mes_Final;
32     Config1.Dia_Inicio:=Dia_Inicio;
33     Config1.Dia_Final:=Dia_Final;
34     Config1.Hora_Inicio:=Hora_Inicio;
35     Config1.Hora_Final:=Hora_Final;
36     Config1.Minuto_Inicio:=Minuto_Inicio;
37     Config1.Minuto_Final:=Minuto_Final;
38
39     FOR i:=0 TO 6 DO
40         IF Dia_Semana_0 = TRUE AND i=0 THEN
41             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=0);
42         ELSIF Dia_Semana_1 = TRUE AND i=1 THEN
43             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=1);
44         ELSIF Dia_Semana_2 = TRUE AND i=2 THEN
45             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=2);
46         ELSIF Dia_Semana_3 = TRUE AND i=3 THEN
47             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=3);
48         ELSIF Dia_Semana_4 = TRUE AND i=4 THEN
49             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=4);
50         ELSIF Dia_Semana_5 = TRUE AND i=5 THEN
51             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=5);
52         ELSIF Dia_Semana_6 = TRUE AND i=6 THEN
53             Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=6);
54         END_IF
55     END_FOR
56     Config1.Dias_Semana:=Dias_Semana;
57

```

POU FB_EXTRACTOR_FALLOTERMICO

```

1  FUNCTION_BLOCK FB_EXTRACTOR_FALLOTERMICO
2  VAR_INPUT
3      FC1_EnMarcha: BOOL; //diFC1_EnMarcha
4      FC2_EnMarcha: BOOL; //diFC2_EnMarcha
5      FC3_EnMarcha: BOOL; //diFC3_EnMarcha
6      FC4_EnMarcha: BOOL; //diFC4_EnMarcha
7      ExtractorMarcha: BOOL; //diExtractroMarcha
8      FalloTermico: BOOL; //diFalloTermic
9
10 END_VAR
11
12 VAR_OUTPUT
13     Alm_NoExtractorEnMarcha: BOOL;
14     Alm_Fallo_Termico: BOOL;
15     HAY_ALM: BOOL;
16
17 END_VAR
18
19 VAR
20     ALM: TON;
21     Marcha: BOOL;
22
23     ALM_1: TON;
24 END_VAR

```



POU_PARAM_PID

```
1     FUNCTION_BLOCK PARAM_PID
2     VAR_INPUT
3         tCtrlCycleTime : TIME ;
4         tTaskCycleTime : TIME ;
5         fKp_heating : LREAL ;
6         tTn_heating : TIME ;
7         tTv_heating : TIME ;
8         tTd_heating : TIME ;
9         fKp_cooling : LREAL ;
10        tTn_cooling : TIME ;
11        tTv_cooling : TIME ;
12        tTd_cooling : TIME ;
13        nParameterChangeCycleTicks : UDINT ;
14        fOutMaxLimit : LREAL ;
15        fOutMinLimit : LREAL ;
16    END_VAR
17    VAR_OUTPUT
18        stCTRL_PID_PARAMS : ST_CTRL_PID_SPLITRANGE_PARAMS ;
19    END_VAR
20    VAR
21    END_VAR

1     stCTRL_PID_PARAMS . tCtrlCycleTime := tCtrlCycleTime ;
2     stCTRL_PID_PARAMS . tTaskCycleTime := tTaskCycleTime ;
3     stCTRL_PID_PARAMS . fKp_heating := fKp_heating ;
4     stCTRL_PID_PARAMS . tTn_heating := tTn_heating ;
5     stCTRL_PID_PARAMS . tTv_heating := tTv_heating ;
6     stCTRL_PID_PARAMS . tTd_heating := tTd_heating ;
7     stCTRL_PID_PARAMS . fKp_cooling := fKp_cooling ;
8     stCTRL_PID_PARAMS . tTn_cooling := tTn_cooling ;
9     stCTRL_PID_PARAMS . tTv_cooling := tTv_cooling ;
10    stCTRL_PID_PARAMS . tTd_cooling := tTd_cooling ;
11    stctrl_pid_params . nParameterChangeCycleTicks :=
nParameterChangeCycleTicks ;
12    stCTRL_PID_PARAMS . fOutMaxLimit := fOutMaxLimit ;
13    stCTRL_PID_PARAMS . fOutMinLimit := fOutMinLimit ;
14
```

POU_LINEALIZACION

```
1  FUNCTION linealizacion : int
2  VAR_INPUT
3      Variable_Lin : LREAL ;
4      Valor_Max_X : LREAL ; //Valor máximo que nos puede dar nuestra entrada
analógica
5      Valor_Min_X : LREAL ; //Valor mínimo que nos puede dar nuestra entrada
anaalógica
6      Valor_Max_Y : LREAL ; //Valor máximo al que queremos limitar
7      Valor_Min_Y : LREAL ; //Valor mínimo oal que queremos limitar
8
9  END_VAR
10 VAR
11     A : LREAL ;
12     B : LREAL ;
13     Valor : LREAL ;
14 END_VAR
..

1  //ys = ymin+((xs-xmin)/(xmax-xmin)*(ymax-ymin))
2  Valor := Valor_Min_Y + ( Variable_Lin - Valor_Min_X ) / ( Valor_Max_X - Valor_Min_X
) * ( Valor_Max_Y - Valor_Min_Y ) ;
3  linealizacion := LREAL_TO_INT ( Valor ) ;
```

POU_DIAMES

```
1     FUNCTION_BLOCK FB_DiaMes
2     VAR_INPUT
3
4         Dia_Mes_Inicio : DWORD ;
5         Dia_Mes_Final : DWORD ;
6         Dia_Actual : WORD ;
7
8     END_VAR
9
10    VAR_OUTPUT
11        Dia_Inicio : BOOL ;
12        Dia_Final : BOOL ;
13    END_VAR
14
15    VAR
16        Timer1 : TON ;
17        NT_Get_Time1 : NT_GetTime ;
18        miTimeStruct : TIMESTRUCT ;
19
20        bitRevisar : SINT ;
21
22    END_VAR
```

```
1     //Dias del mes en el rango de bits 1...31,
2     //Ejemplo dia 18 es 16#00040000 porque va del dit 1 al 31 no escribe en e
    bit 0.
3
4     //Hacemos comparaciones para evaluar si está en dia del mes
5     Dia_Inicio := ( Dia_Actual >= Dia_Mes_Inicio ) ;
6     Dia_Final := ( Dia_Actual <= Dia_Mes_Final ) ;
```


POU_ENHORARIO

```

1  FUNCTION_BLOCK EnHorario
2  VAR_INPUT
3      config : ConfiguracionHoraria ;
4
5  END_VAR
6
7  VAR_OUTPUT
8      EnHorarios : BOOL ;
9
10 END_VAR
11
12 VAR
13     EnMes : FB_Mes ;
14     MesOk : BOOL ;
15     meses_Activo : DWORD := 0 ;
16     bitI : SINT ;
17     bitF : SINT ;
18     EnDiaSemana : FB_DiaSemana ;
19     DiaSemanaOk : BOOL ;
20     EnHora : FB_Hora ;
21     HoraOK : BOOL ;
22
23     Timer1 : TON ;
24     NT_Get_Timed : NT_GetTime ;
25     miTimeStruct : TIMESTRUCT ;
26
27
28     DiaMesOk : BOOL ;
29     EnDiaMes : FB_DiaMes ;
30     Dia_InicioOK : BOOL ;
31     Dia_FinalOK : BOOL ;
32     Dia_activo : INT ;
33 END_VAR
..
1  meses_Activo := 0 ;
2
3  //Temporizador para activar periodicamente la FB NT_GetTimed
4  Timer1 ( IN := NOT Timer1.Q , PT := T#2S , Q => , ET => ) ;
5
6
7  //Obtenemos tiempo de Windows
8  NT_Get_Timed (
9      NETID := ,
10     START := Timer1.Q ,
11     TMOUT := ,
12     BUSY => ,
13     ERR => ,
14     ERRID => ,
15     TIMESTR => miTimeStruct ) ;
16
17

```

```

18      //Comprovamos que estamos en mes
19      bitI := ( config . Mes_Inicio );
20      bitF := ( config . Mes_Final );
21      meses_Activo := SETBIT32 ( inVal32 := meses_Activo , bitNo := bitI );

22      meses_Activo := SETBIT32 ( inVal32 := meses_Activo , bitNo := bitF );
23      IF miTimeStruct . wMonth > bitI AND miTimeStruct . wMonth < bitF THEN
24          MesOk := TRUE;
25      ELSIF miTimeStruct . wMonth > bitI THEN
26          MesOk := FALSE;
27      ELSE
28          EnMes ( meses_Activo := meses_Activo , mes_actual := miTimeStruct . wMonth
29              , Mes => );
30          MesOk := EnMes . Mes;
31      END_IF

32      //Comporvamos el día de la semana
33      EnDiaSemana (
34          Dias_Activo := config . Dias_Semana ,
35          Dia_Actual := miTimeStruct . wDayOfWeek ,
36          Dias_Semana => );
37      DiaSemanaOk := EnDiaSemana . Dias_Semana;

38      //Comprovamos las horas de funcionamiento
39      EnHora (
40          hora_inicio := config . Hora_Inicio ,
41          hora_final := config . Hora_Final ,
42          minuto_inicial := config . Minuto_Inicio ,
43          minuto_final := config . Minuto_Final ,
44          hora_actual := miTimeStruct . wHour ,
45          minutos_actual := miTimeStruct . wMinute ,
46          EnHora => );
47      HoraOK := EnHora . EnHora;

48      //Comprobamos que estemos en día de mes para mes de inicio y final
49      EnDiaMes (
50          Dia_Mes_Inicio := config . Dia_Inicio ,
51          Dia_Mes_Final := config . Dia_Final ,
52          Dia_Actual := miTimeStruct . wDay ,
53          Dia_Inicio => ,
54          Dia_Final => );
55      Dia_InicioOK := EnDiaMes . Dia_Inicio;
56      Dia_FinalOK := EnDiaMes . Dia_Final;

57      //Acción de control de la programación horaria
58      IF ( miTimeStruct . wMonth = config . Mes_Inicio ) THEN
59          EnHorarios := MesOk AND DiaSemanaOk AND HoraOK AND Dia_InicioOK;
60      ELSIF ( miTimeStruct . wMonth = config . Mes_Final ) THEN
61          EnHorarios := MesOk AND DiaSemanaOk AND HoraOK AND Dia_FinalOK;
62      ELSE
63          EnHorarios := MesOk AND DiaSemanaOk AND HoraOK;
64      END_IF
65      EnHorarios := MesOk AND DiaSemanaOk AND HoraOK;
66      END_IF
67
68

```

POU_HORA

```
1     FUNCTION_BLOCK FB_Hora
2     VAR_INPUT
3         hora_inicio : WORD ;
4         hora_final : WORD ;
5         minuto_inicial : WORD ;
6         minuto_final : WORD ;
7         hora_actual : WORD ;
8         minutos_actual : WORD ;
9
10    END_VAR
11
12    VAR_OUTPUT
13        EnHora : BOOL ;
14    END_VAR
15
16    VAR
17
18        minicio : UINT ;
19        mFinal : UINT ;
20        mActual : UINT ;
21        hSystem : UINT ;
22    END_VAR
23
24    //Pasamos todo a minutos
25    minicio := hora_inicio * 60 + minuto_inicial ;
26    mFinal := hora_final * 60 + minuto_final ;
27    mActual := hora_actual * 60 + minutos_actual ;
28
29    //Comparamos para evaluar si esté en horario
30    EnHora := ( (mActual >= minicio) AND (mActual <= mFinal) ) ;
```

POU_MES

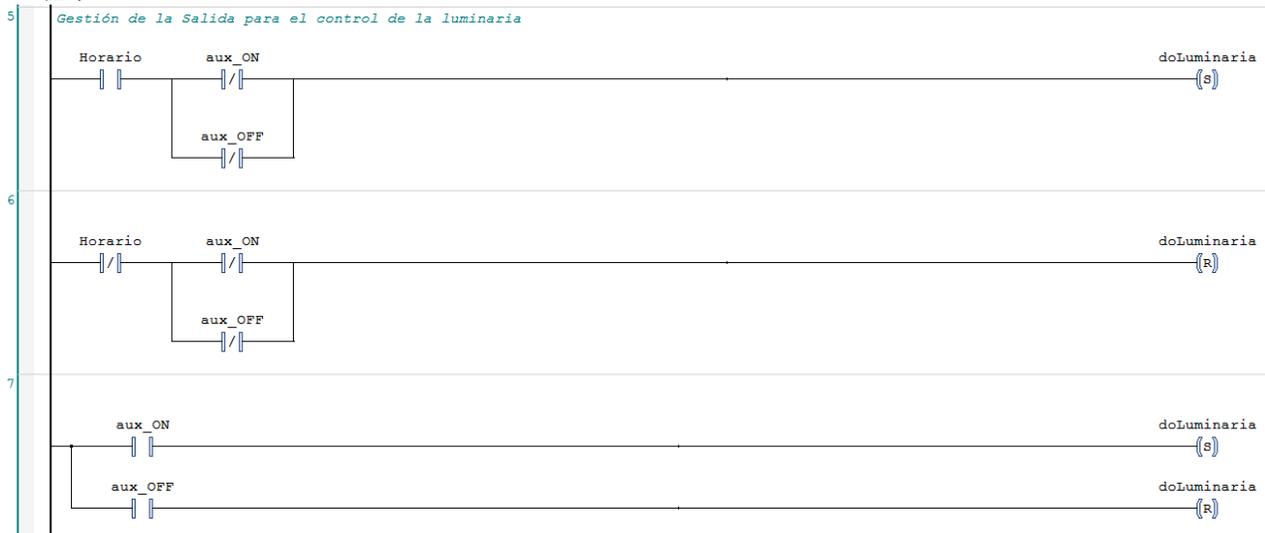
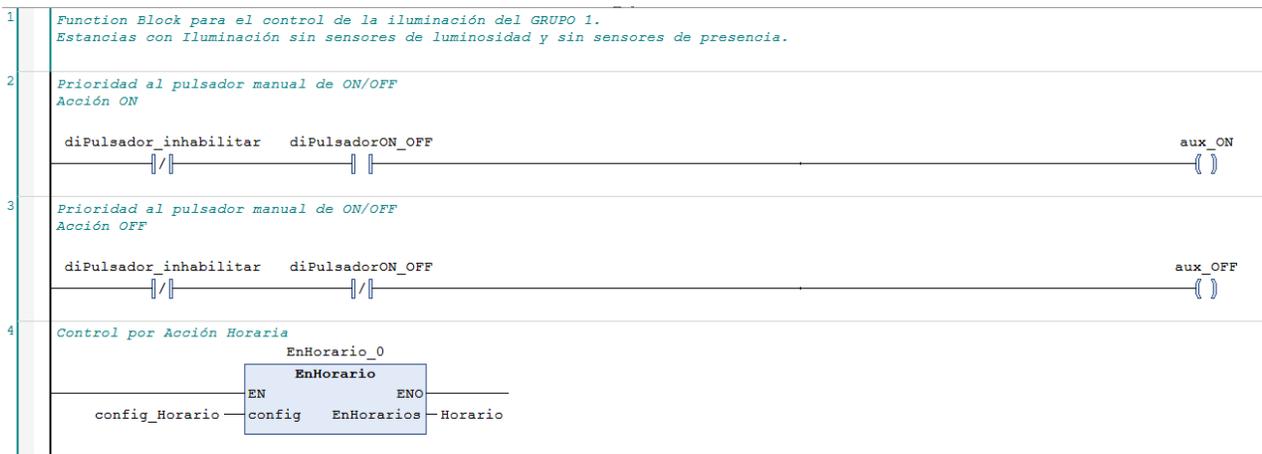
```
1     FUNCTION_BLOCK FB_Mes
2     VAR_INPUT
3
4         meses_Activo : DWORD ;
5         mes_actual : WORD ;
6
7     END_VAR
8
9     VAR_OUTPUT
10        Mes : BOOL ;
11    END_VAR
12
13    VAR
14        bitRevisar : SINT ;
15    END_VAR
16    ~
17
18    //Pasamos todo a minutos
19    inicio := hora_inicio * 60 + minuto_inicial ;
20    mFinal := hora_final * 60 + minuto_final ;
21    mActual := hora_actual * 60 + minutos_actual ;
22
23    //Comparamos para evaluar si está en horario
24    EnHora := ( ( mActual >= inicio ) AND ( mActual <= mFinal ) ) ;
25    ~
```

POU_GRUPO1

```

1 FUNCTION_BLOCK GRUPO1
2 VAR_INPUT
3     diPulsador_inhabilitar:BOOL; //Pulsador para inhabilitar el pulsador manual de encendido y apagado
4     diPulsadorON_OFF:BOOL; //Pulsador manual de encendido y apagado
5     config_Horario: ConfiguracionHoraria;
6 END_VAR
7 VAR_OUTPUT
8     doLuminaria:BOOL; //Salida para dar el control a las luminarias
9 END_VAR
10 VAR
11     EnHorario_0: EnHorario;
12     Horario: BOOL;
13     aux_ON: BOOL;
14     aux_OFF: BOOL;
15 END_VAR

```

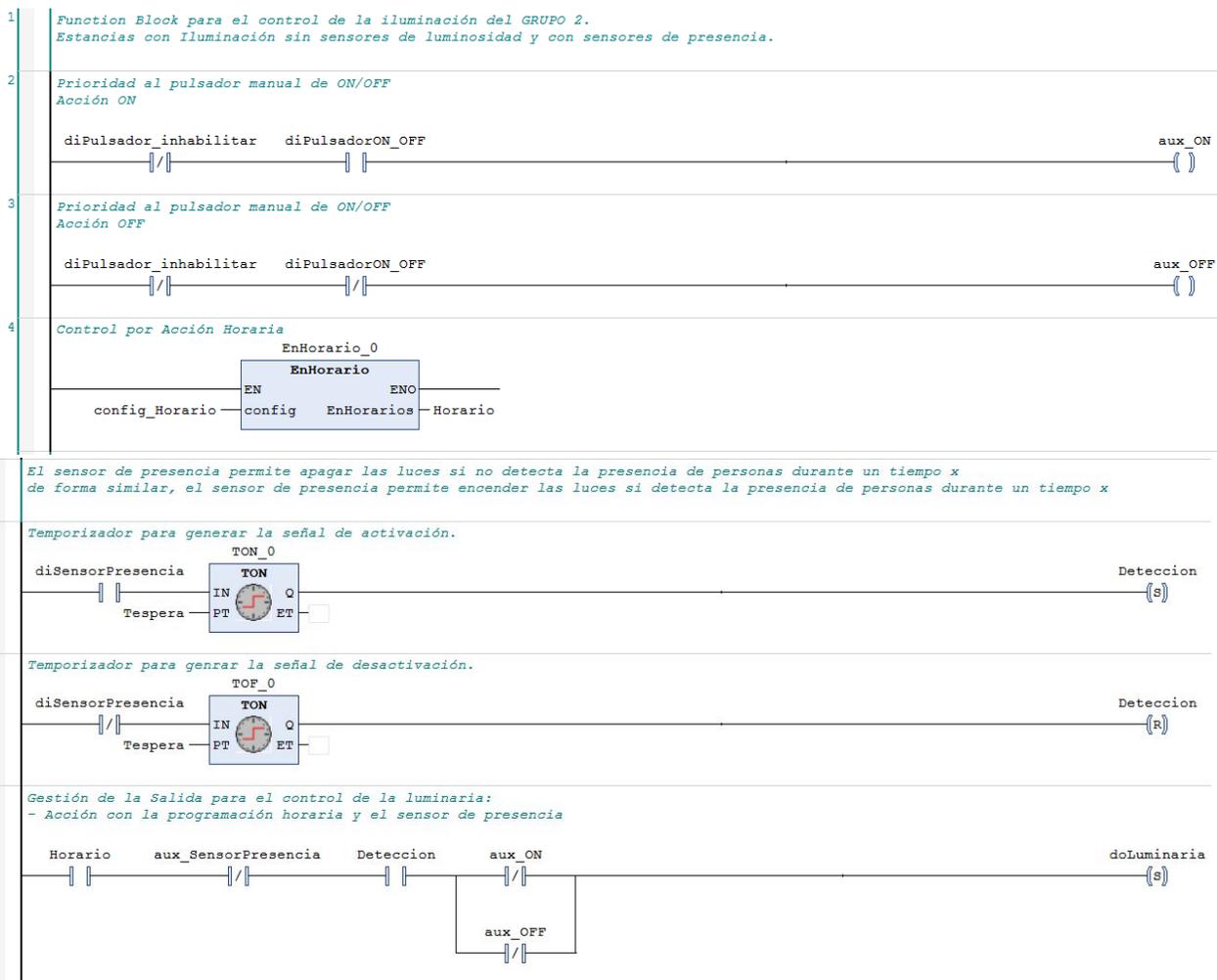


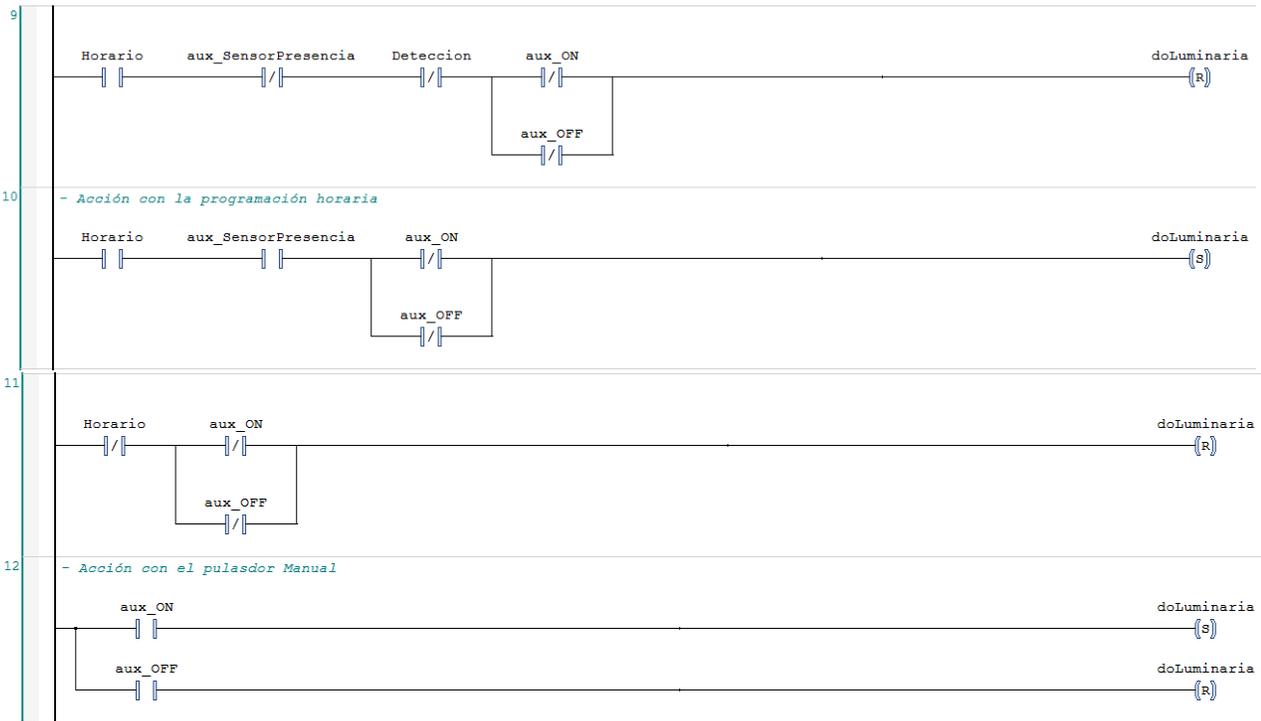
POU_GRUPO2

```

1 FUNCTION_BLOCK GRUPO2
2 VAR_INPUT
3     diPulsador_inhabilitar:BOOL; //Pulsador para inhabilitar el pulsador manual de encendido y apagado
4     diPulsadorON_OFF:BOOL; //Pulsador manual de encendido y apagado
5     diSensorPresencia:BOOL; //Señal de detección del sensor de Presencia
6     aux_SensorPresencia:BOOL; //Boton en la pantalla de control que permite deshabilitar el sensor de presencia
7     Tespera:TIME; //Tiempo espera para los temporizadores del Sensor de Presencia
8     config_Horario: ConfiguracionHoraria;
9 END_VAR
10 VAR_OUTPUT
11     doLuminaria:BOOL; //Salida para dar el control a las luminarias
12 END_VAR
13 VAR
14     EnHorario_0: EnHorario;
15     Horario: BOOL;
16     aux_ON: BOOL;
17     aux_OFF: BOOL;
18     TON_0: TON;
19     TOF_0: TON;
20     Deteccion:BOOL;
21 END_VAR
22

```



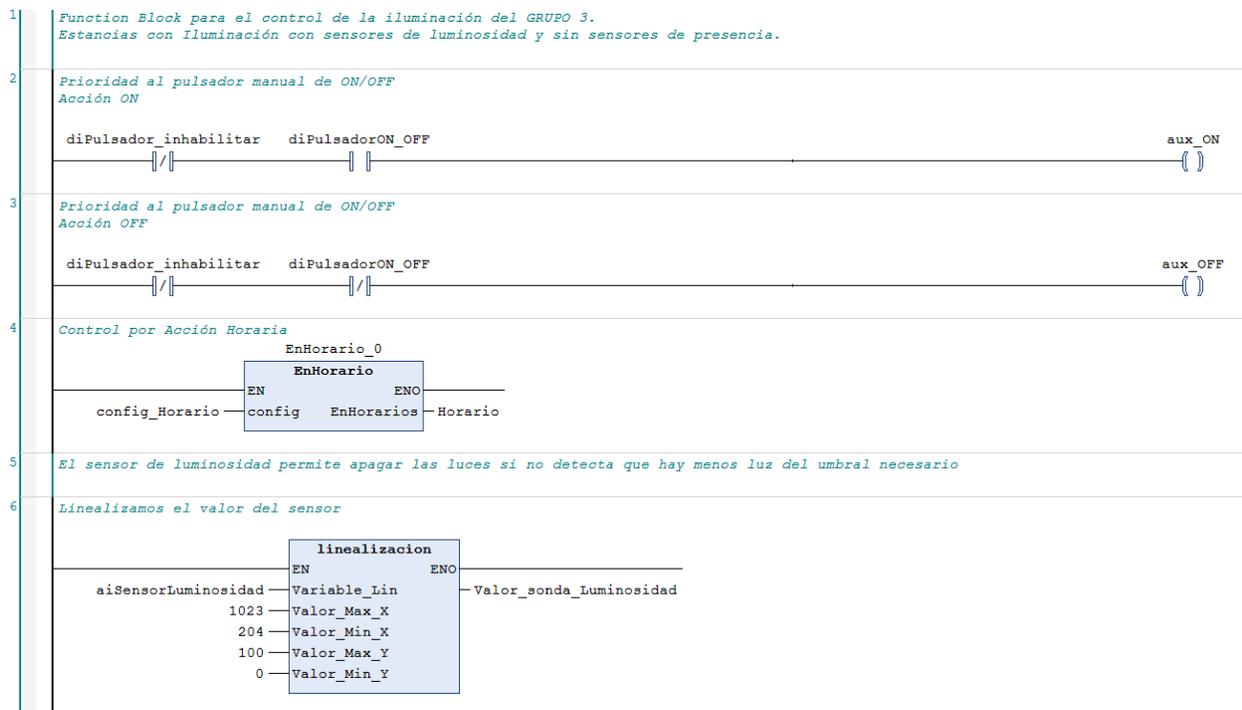


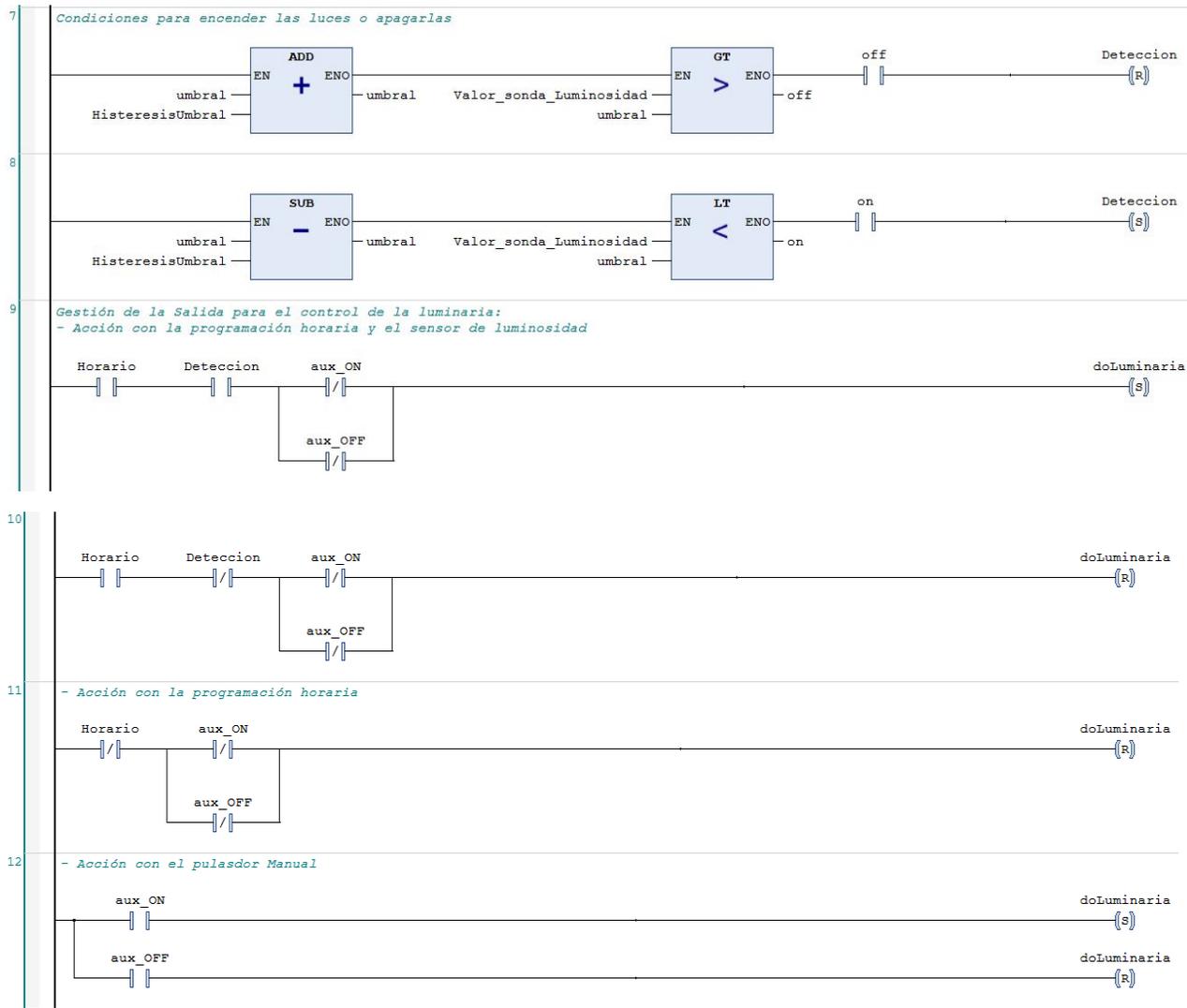
POU_GRUPO3

```

1 FUNCTION_BLOCK GRUPO3
2 VAR_INPUT
3     diPulsador_inhabilitar:BOOL; //Pulsador para inhabilitar el pulsador manual de encendido y apagado
4     diPulsadorON_OFF:BOOL; //Pulsador manual de encendido y apagado
5     aiSensorLuminosidad:INT; //Señal analógica del sensor de Luminosidad
6
7     umbral:INT; //Variable que nos indica cuando es necesario encender las luces o apagarlas
8     HisteresisUmbral:int; //Variable para no regular el encendido y apagado no deseado
9     config_Horario: ConfiguracionHoraria;
10 END_VAR
11 VAR_OUTPUT
12     doLuminaria:BOOL; //Salida para dar el control a las luminarias
13 END_VAR
14 VAR
15     EnHorario_0: EnHorario;
16     Horario: BOOL;
17     aux_ON: BOOL;
18     aux_OFF: BOOL;
19     Deteccion:BOOL; //auxiliar para indicar si se detecta mas luz o menos para encender o apagar las luces
20     Valor_sonda_Luminosidad: INT;
21     off: BOOL;
22     on: BOOL;
23 END_VAR

```



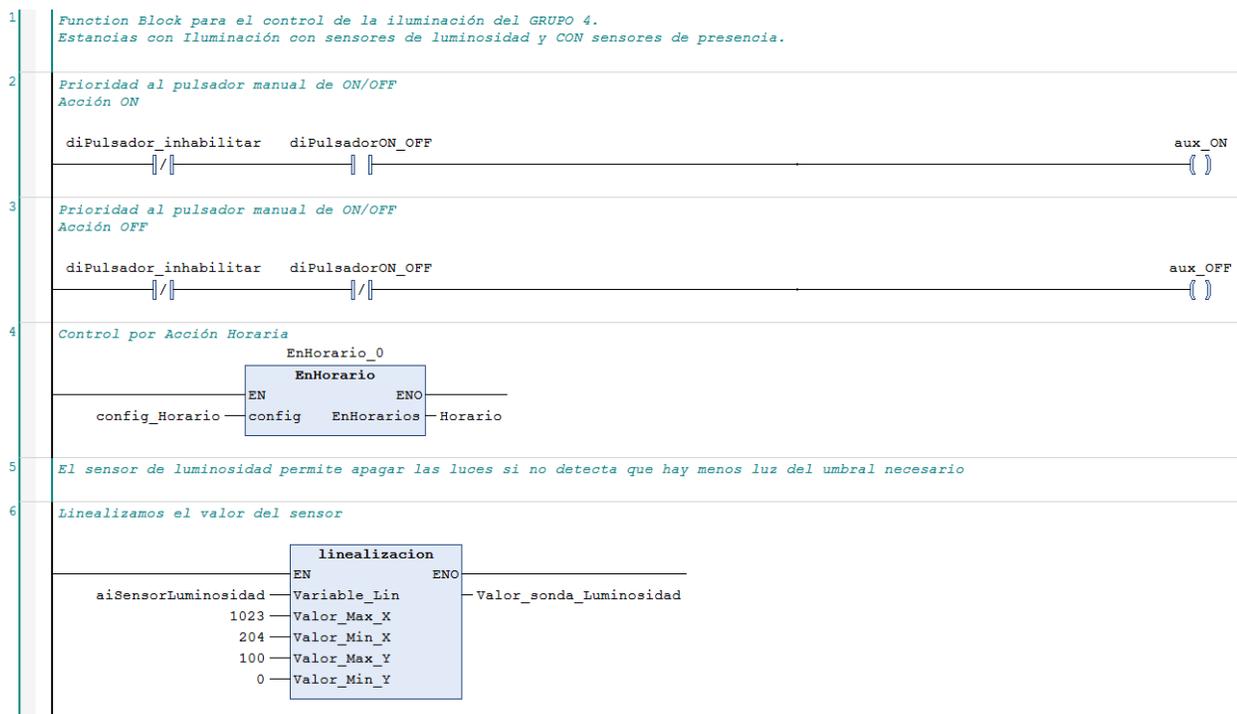


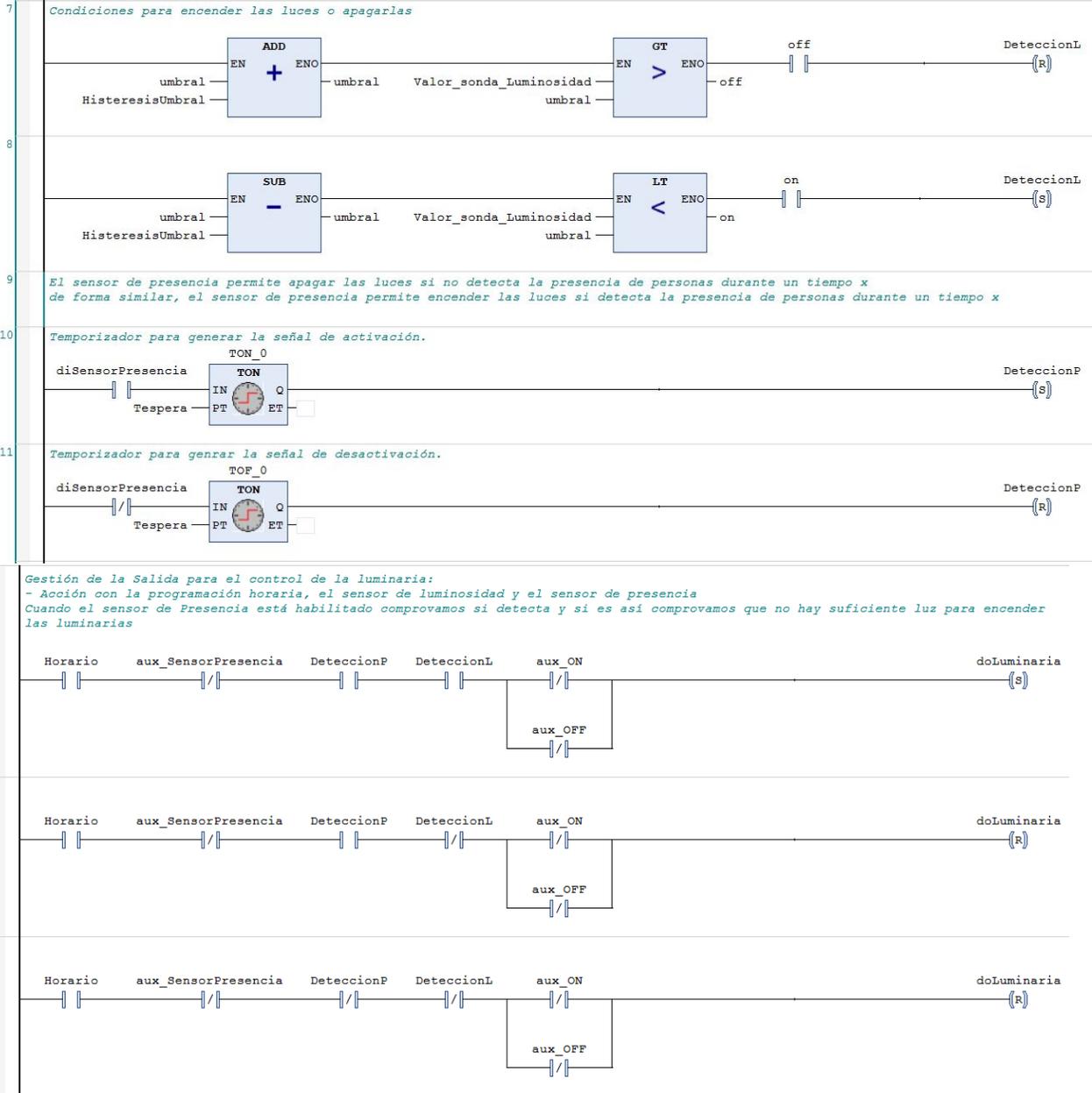
POU_GRUPO4

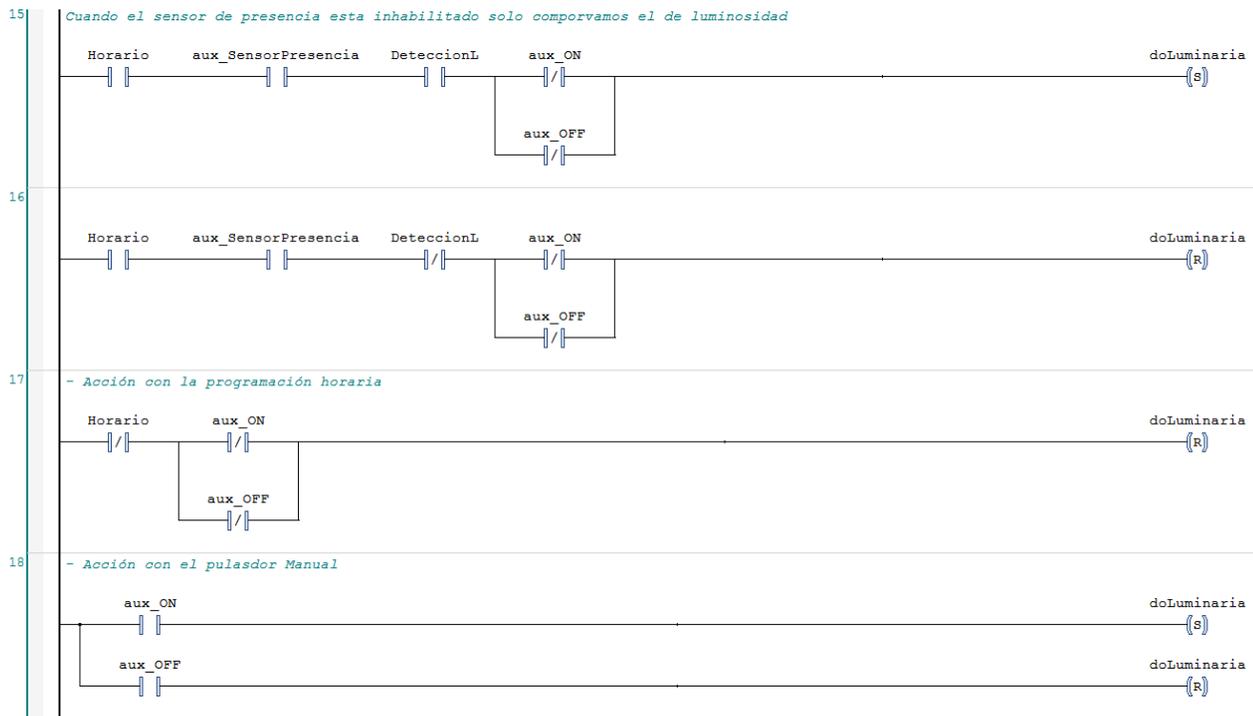
```

1 FUNCTION_BLOCK GRUPO4
2 VAR_INPUT
3     diPulsador_inhabilitar:BOOL; //Pulsador para inhabilitar el pulsador manual de encendido y apagado
4     diPulsadorON_OFF:BOOL; //Pulsador manual de encendido y apagado
5
6     aiSensorLuminosidad:INT; //Señal analógica del sensor de Luminosidad
7     umbral:INT; //Variable que nos indica cuando es necesario encender las luces o apagarlas
8     HisteresisUmbral:INT; //Variable para no regular el encendido y apagado no deseado
9
10    diSensorPresencia:BOOL; //Señal de detección del sensor de Presencia
11    aux_SensorPresencia:BOOL; //Boton en la pantalla de control que permite deshabilitar el sensor de presencia
12    Tespera:TIME; //Tiempo espera para los temporizadores del Sensor de Presencia
13
14    config_Horario: ConfiguracionHoraria; //Estructura para la configuración horaria
15 END_VAR
16 VAR_OUTPUT
17     doLuminarias:BOOL; //Salida para dar el control a las luminarias
18 END_VAR
19 VAR
20     EnHorario_0: EnHorario;
21     Horario: BOOL;
22     aux_ON: BOOL;
23     aux_OFF: BOOL;
24     TON_0: TON;
25     TOF_0: TON;
26     DeteccionP:BOOL;
27     DeteccionL:BOOL;
28     Valor_sonda_Luminosidad: INT;
29     off: BOOL;
30     on: BOOL;
31 END_VAR

```







POU_PERSIANAS

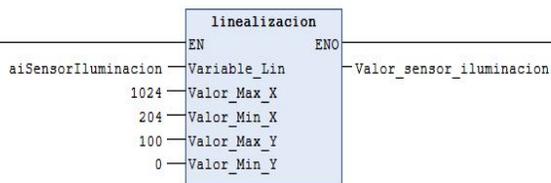
```

1  FUNCTION_BLOCK PERSIANAS
2  VAR_INPUT
3      aiSensorIluminacion: INT;
4      maxFranja1: INT;
5      maxFranja2: INT;
6      maxFranja3: INT;
7      maxFranja4: INT;
8      maxFranja5: INT;
9
10     HoraFinal1:WORD;
11     HoraFinal2:WORD;
12     HoraFinal3:WORD;
13     HoraFinal4:WORD;
14     HoraFinal5:WORD;
15
16     PrioridadHorario:BOOL;
17     PrioridadSensor:BOOL;
18
19     Apertura:INT;
20     MANUAL: BOOL;
21
22
23 END_VAR
24 VAR_OUTPUT
25     aoSistemaMotorizado:INT;
26 END_VAR
27 VAR
28     Valor_sensor_iluminacion: INT;
29     AperturaPersianas: INT;
30     Default:BOOL;
31
32     TON_0: ton;
33     NT_GetTime_0: NT_GetTime;
34     miTimeStruct: TIMESTRUCT;
35
36 END_VAR

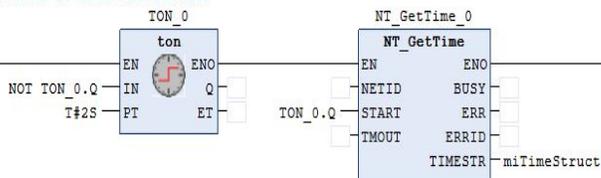
```

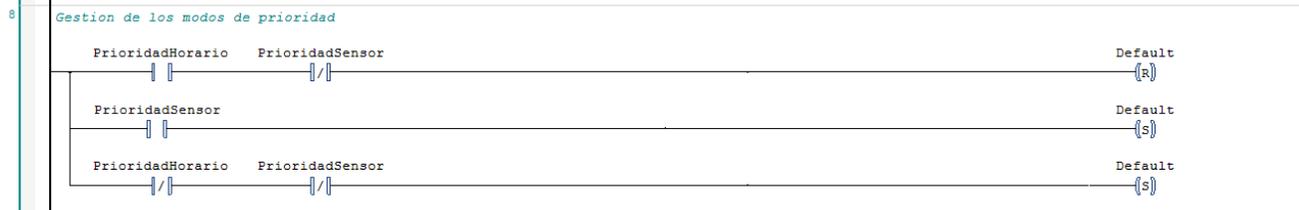
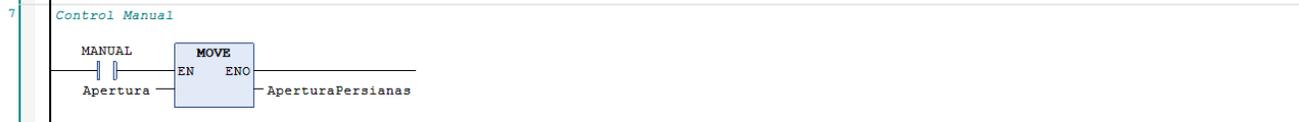
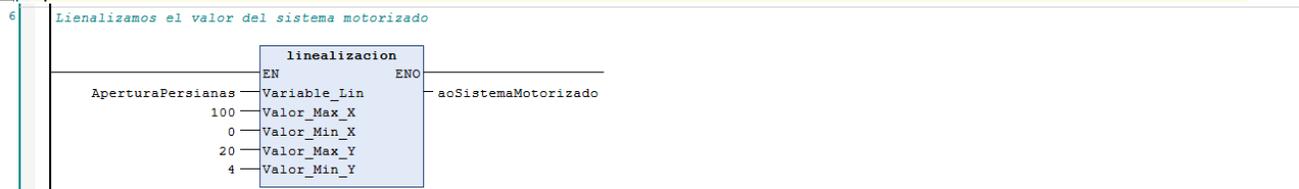
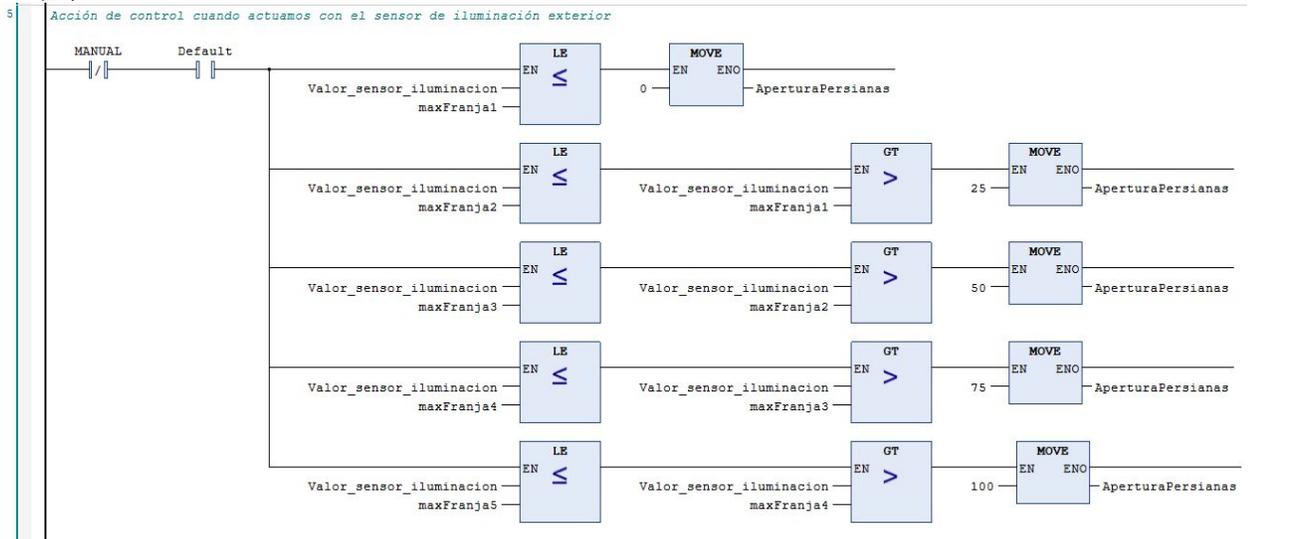
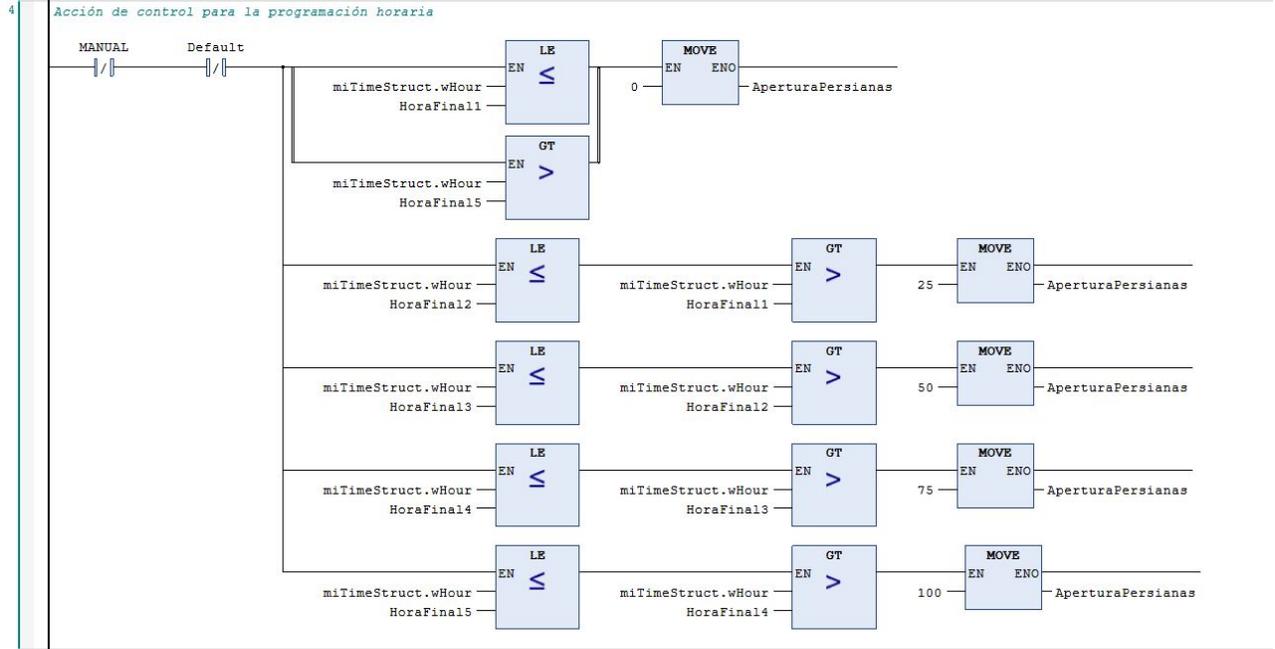
1 | *Function BLock para el control del sistema motorizado de las persianas*

2 | *Linealizamos el valor del sensor*



3 | *Obtenemos la hora del sistema*





POU_HORARIOS_SEMANA

```

1     FUNCTION_BLOCK HORARIOS_SEMANALES
2     VAR_INPUT
3     Dia_Semana_0 : BOOL ; //Domingo
4         Dia_Semana_1 : BOOL ;
5         Dia_Semana_2 : BOOL ;
6         Dia_Semana_3 : BOOL ;
7         Dia_Semana_4 : BOOL ;
8         Dia_Semana_5 : BOOL ;
9         Dia_Semana_6 : BOOL ; //Sabado
10    Hora_Inicio : WORD ;
11    Hora_Final : WORD ;
12    Minuto_Inicio : WORD ;
13    Minuto_Final : WORD ;
14    END_VAR
15    VAR_OUTPUT
16        Config1 : ConfiguracionHoraria ;
17    END_VAR
18    VAR
19        i : SINT ;
20        Dias_Semana : DWORD ;
21    END_VAR
1     Dias_Semana:=0;
2     i:=0;
3         Config1.Mes_Inicio:=1;
4         Config1.Mes_Final:=12;
5         Config1.Dia_Inicio:=1;
6         Config1.Dia_Final:=31;
7         Config1.Hora_Inicio:=Hora_Inicio;
8         Config1.Hora_Final:=Hora_Final;
9         Config1.Minuto_Inicio:=Minuto_Inicio;
10        Config1.Minuto_Final:=Minuto_Final;
11
12        FOR i:=0 TO 6 DO
13            IF Dia_Semana_0 = TRUE AND i=0 THEN
14                Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=0);
15            ELSIF Dia_Semana_1 = TRUE AND i=1 THEN
16                Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=1);
17            ELSIF Dia_Semana_2 = TRUE AND i=2 THEN
18                Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=2);
19            ELSIF Dia_Semana_3 = TRUE AND i=3 THEN
20                Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=3);
21            ELSIF Dia_Semana_4 = TRUE AND i=4 THEN
22                Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=4);
23            ELSIF Dia_Semana_5 = TRUE AND i=5 THEN
24                Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=5);
25            ELSIF Dia_Semana_6 = TRUE AND i=6 THEN
26                Dias_Semana:=SETBIT32(inVal32:=Dias_Semana , bitNo:=6);
27            END_IF
28        END_FOR
29        Config1.Dias_Semana:=Dias_Semana;

```

POU_IoT

```
1  FUNCTION_BLOCK IoT
2  VAR_INPUT
3      Energ_Activa : REAL ;
4      Energ_Reactiva : REAL ;
5      Energ_Aparente : REAL ;
6
7  END_VAR
8
9  VAR_OUTPUT
10 END_VAR
11
12 VAR
13     bInicializacion : BOOL := TRUE ;
14     bConnectar : BOOL := TRUE ;
15     // Cliente MQTT
16     fbMqttClient : FB_IotMqttClient ;
17     // Mensaje JSON
18     sJson : STRING ( 255 ) ;
19     fbCrearJson : FB_JsonSaxWriter ;
20     // Variables para publicar mensajes de dispositivo a nube
21     sTopicAlQuePublicar : STRING ( 255 ) ;
22     nMensajesEnviados : UDINT ;
23     fbTemporizador : TON ;
24
25     fbMessageQueue : FB_IotMqttMessageQueue ;
26     fbMessage : FB_IotMqttMessage ;
27     // Decodificar JSON
28     jsonDoc : SJsonValue ;
29     fbDecodificarJson : FB_JsonDomParser ;
30     jsonProp : SJsonValue ;
31
32     // Obtener fecha del sistema
33     Fecha : STRING ;
34     Ticks : T_FILETIME ;
35     fbObtenerTicks : GETCPUOUNTER ;
36
37 END_VAR
38
39 VAR CONSTANT
40     sDeviceId : STRING := 'PC_PLANTA_2' ;
41     sRutaCertificadoAzure : STRING ( 255 ) :=
42     'C:\Users\PCDisinel\Desktop\TFG\Documentos TFG\Azure.cer' ;
43     sAzureSasToken : STRING ( 511 ) :=
44     'HostName=IoTHubAndresDisinel.azure-devices.net;DeviceId=PC_PLANTA_2;SharedAccess:
45
46     sr=IoTHubAndresDisinel.azure-devices.net%2Fdevices%2FPC_PLANTA_2&sig=xldYT0%2BNh7:
47 ;
48 END_VAR
49 ..
```

```

2  IF bInicializacion THEN
3      bInicializacion := FALSE ;
4      // Preparar topics a los que publicar
5      sTopicAlQuePublicar := 'devices/PC_PLANTA_2/messages/events/readpipe/' ;
6      // Datos para la conexion a Azure IoT Hub por MQTT
7      fbMqttClient.stTLS.sCA := sRutaCertificadoAzure ;
8      fbMqttClient.stTLS.sAzureSas := sAzureSasToken ;
9      fbMqttClient.ipMessageQueue := fbMessageQueue ;
10  END_IF
11  //Comprobar si esta conectado
12  fbMqttClient.Execute ( bConnectar ) ;
13
14  // Leer hora del sistema
15  fbObtenerTicks ( cpuCntLoDW => Ticks.dwLowDateTime , cpuCntHiDW => Ticks.
dwHighDateTime ) ;
16  Fecha := SYSTEMTIME_TO_STRING ( FILETIME_TO_SYSTEMTIME ( Ticks ) ) ;
17  Fecha [ 10 ] := F_ToASC ( 'T' ) ;
18
19  // Si hemos conectado con el Dispositivo tendremos una realimentación TRUE.
20  IF fbMqttClient.bConnected THEN
21      // Dar la orden de envio de datos cada minuto
22      fbTemporizador ( IN := TRUE , PT := T#1M ) ;
23      IF fbTemporizador.Q THEN
24          fbTemporizador ( IN := FALSE ) ;
25          // Crear telegrama JSON a enviar
26          // Vacía el objeto JSON
27          fbCrearJson.ResetDocument ( ) ;
28          // Añade la llave de apertura
29          fbCrearJson.StartObject ( ) ;
30          // Añade las parejas clave/valor al mensaje
31          fbCrearJson.AddKey ( 'Energía Activa' ) ;
32          fbCrearJson.AddReal ( Energ_Activa ) ;
33          fbCrearJson.AddKey ( 'Energía Reactiva' ) ;
34          fbCrearJson.AddReal ( Energ_Reactiva ) ;
35          fbCrearJson.AddKey ( 'Energía Aparente' ) ;
36          fbCrearJson.AddReal ( Energ_Aparente ) ;
37          fbCrearJson.AddKey ( 'Fecha' ) ;
38          fbCrearJson.AddString ( Fecha ) ;
39          // Añade la llave de clausura
40          fbCrearJson.EndObject ( ) ;
41          // Obtiene el STRING con el mensaje JSON
42          sJson := fbCrearJson.GetDocument ( ) ;
43          // Publicar el telegrama JSON a la cuenta de Azure
44          fbMqttClient.Publish ( sTopic := sTopicAlQuePublicar ,
45                                pPayload := ADR ( sJson ) ,
46                                nPayloadSize := TO_UDINT ( LEN ( sJson ) ) ,
47                                eQoS := IcIotMqttQos.AtMostOnceDelivery ,
48                                bRetain := FALSE ,
49                                bQueue := FALSE ) ;
50      END_IF
51  END_IF

```

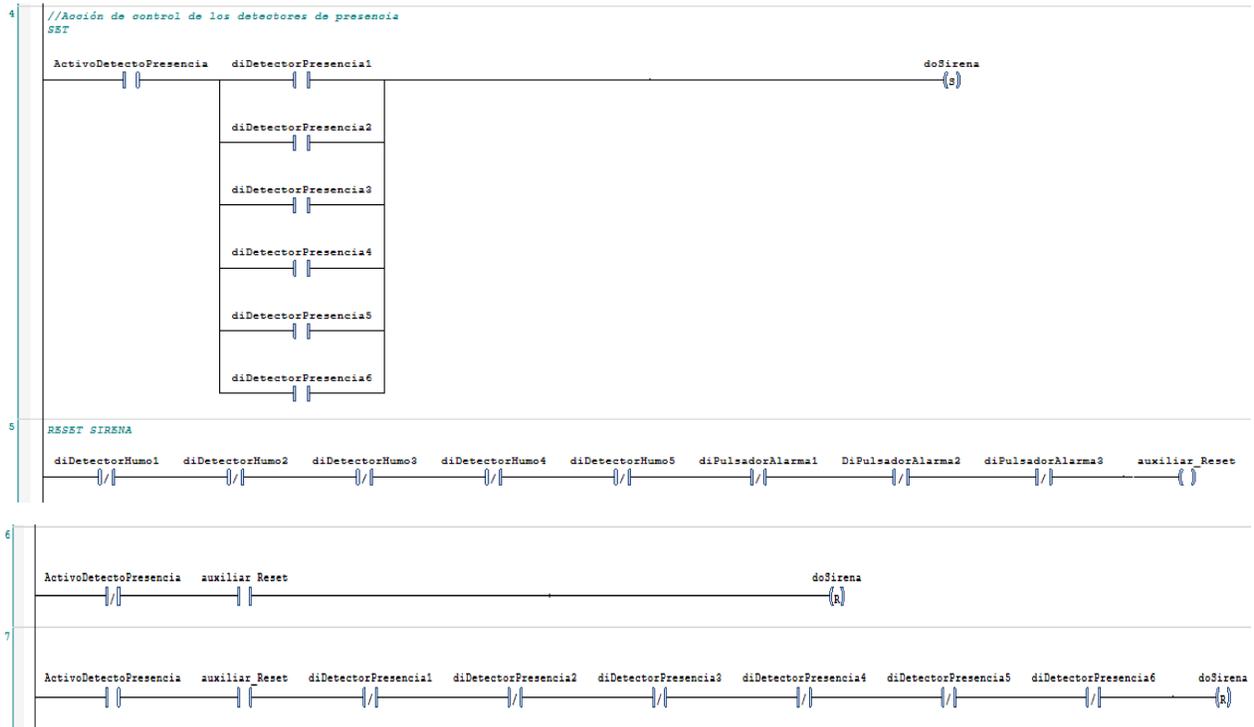
POU_SEGURIDAD_P1

```

1  FUNCTION_BLOCK SEGURIDAD_P1
2  VAR_INPUT
3      //Detectores de humo
4      diDetectorHumo1:BOOL;
5      diDetectorHumo2:BOOL;
6      diDetectorHumo3:BOOL;
7      diDetectorHumo4:BOOL;
8      diDetectorHumo5:BOOL;
9      //Pulsadores de alarma
10     diPulsadorAlarma1:BOOL;
11     //Detectores de presencia
12     diDetectorPresencia:BOOL;
13     diDetectorPresencia2: BOOL;
14     diDetectorPresencia3:BOOL;
15     diDetectorPresencia4:BOOL;
16     diDetectorPresencia5:BOOL;
17     diDetectorPresencia6:BOOL;
18     //Auxiliar para activar los detectores de presencia
19     ActivoDetectoPresencia:BOOL;
20
21 END_VAR
22 VAR_OUTPUT
23     doSirena:BOOL;
24 END_VAR
25 VAR
26     auxiliar_Reset: BOOL;
27     DiPulsadorAlarma2: BOOL;
28     diPulsadorAlarma3: BOOL;
29 END_VAR

```





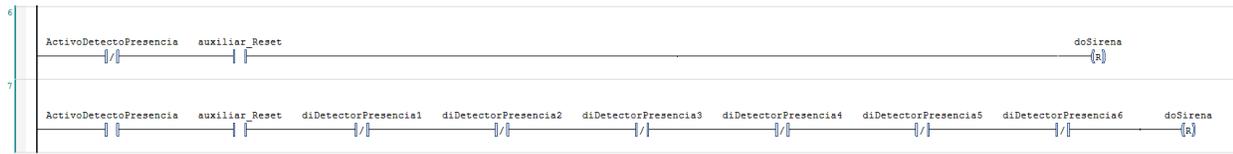
POU_SEGURIDAD_P2

```

1  FUNCTION_BLOCK SEGURIDAD_P2
2  VAR_INPUT
3      //Detectores de humo
4      diDetectorHumo1:BOOL;
5      diDetectorHumo2:BOOL;
6      diDetectorHumo3:BOOL;
7      diDetectorHumo4:BOOL;
8      diDetectorHumo5:BOOL;
9      //Pulsadores de alarma
10     diPulsadorAlarma1:BOOL;
11     diPulsadorAlarma2: BOOL;
12     diPulsadorAlarma3: BOOL;
13     diPulsadorAlarma4: BOOL;
14     //Detectores de presencia
15     diDetectorPresencia1:BOOL;
16     diDetectorPresencia2: BOOL;
17     diDetectorPresencia3:BOOL;
18     diDetectorPresencia4:BOOL;
19     diDetectorPresencia5:BOOL;
20     diDetectorPresencia6:BOOL;
21     //Auxiliar para activar los detectores de presencia
22     ActivoDetectoPresencia:BOOL;
23
24 END_VAR
25 VAR_OUTPUT
26     doSirena:BOOL;
27 END_VAR
28 VAR
29     auxiliar_Reset: BOOL;
30
31 END_VAR

```



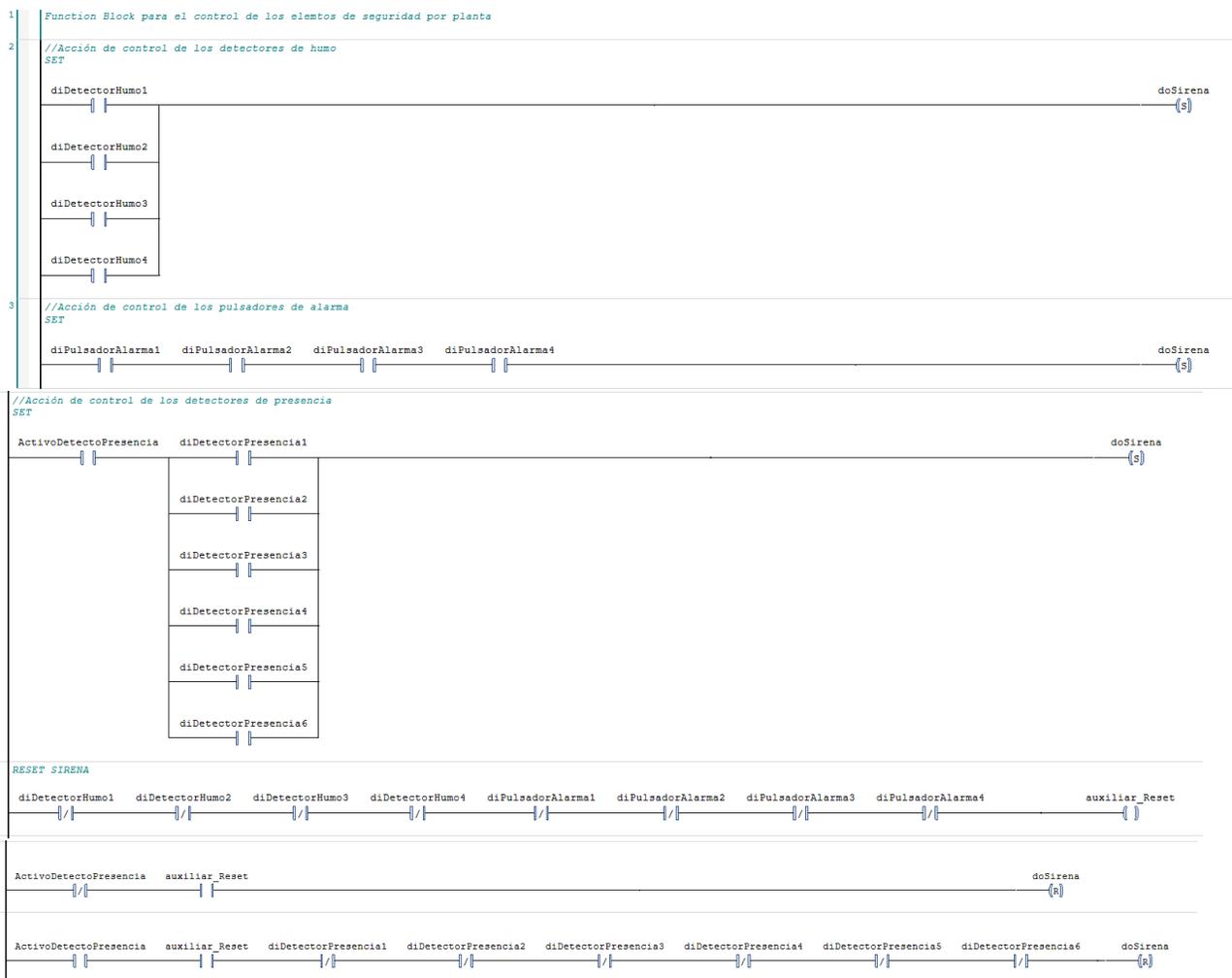


POU_SEGURIDAD_PBAJA

```

1 FUNCTION_BLOCK SEGURIDAD_PBAJA
2 VAR_INPUT
3     //Detectores de humo
4     diDetectorHumo1:BOOL;
5     diDetectorHumo2:BOOL;
6     diDetectorHumo3:BOOL;
7     diDetectorHumo4:BOOL;
8
9     //Pulsadores de alarma
10    diPulsadorAlarma1:BOOL;
11    //Detectores de presencia
12    diDetectorPresencial:BOOL;
13    diDetectorPresencia2: BOOL;
14    diDetectorPresencia3:BOOL;
15    diDetectorPresencia4:BOOL;
16    diDetectorPresencia5:BOOL;
17    diDetectorPresencia6:BOOL;
18    //Auxiliar para activar los detectores de presencia
19    ActivoDetectoPresencia:BOOL;
20
21 END_VAR
22 VAR_OUTPUT
23     doSirena:BOOL;
24 END_VAR
25 VAR
26     auxiliar_Reset: BOOL;
27     diPulsadorAlarma2: BOOL;
28     diPulsadorAlarma3: BOOL;
29     diPulsadorAlarma4: BOOL;
30 END_VAR

```



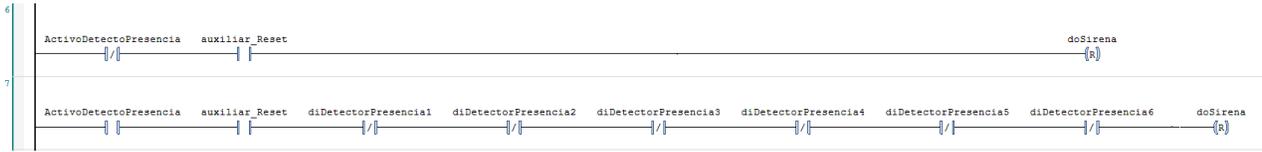
POU_SEGURIDAD_PSOTANO

```

1  FUNCTION_BLOCK SEGURIDAD_PSOTANO
2  VAR_INPUT
3      //Detectores de humo
4      diDetectorHumo1:BOOL;
5      diDetectorHumo2:BOOL;
6      diDetectorHumo3:BOOL;
7      //Pulsadores de alarma
8      diPulsadorAlarma1:BOOL;
9      //Detectores de presencia
10     diDetectorPresencia1:BOOL;
11     diDetectorPresencia2:BOOL;
12     diDetectorPresencia3:BOOL;
13     diDetectorPresencia4:BOOL;
14     diDetectorPresencia5:BOOL;
15     diDetectorPresencia6:BOOL;
16     //Auxiliar para activar los detectores de presencia
17     ActivoDetectoPresencia:BOOL;
18
19 END_VAR
20 VAR_OUTPUT
21     doSirena:BOOL;
22 END_VAR
23 VAR
24     auxiliar_Reset: BOOL;
25     diPulsadorAlarma2: BOOL;
26     diPulsadorAlarma3: BOOL;
27     diPulsadorAlarma4: BOOL;
28 END_VAR

```





POU_MAIN

```
1     PROGRAM MAIN
2     VAR
3
4         EXT_FALLO_TERMICO : FB_EXTRACTOR_FALLOTERMICO ;
5
6         Hay_Alarma : BOOL ;
7         ConfHor1 : CONFIGURACION_HORARIA ;
8         stPID : PARAM_PID ;
9
10        tTn_cool : TIME ;
11        tTv_cool : TIME ;
12        tTn_heat : TIME ;
13        tTv_heat : TIME ;
14        horario : EnHorario ;
15        ConfHor2 : CONFIGURACION_HORARIA ;
16        PasilloP2_G1 : GRUPO1 ;
17
18        Consulta1_P1 : GRUPO2 ;
19        Enfermeria2_P1 : GRUPO3 ;
20        Enfermeria3_P1 : GRUPO4 ;
21        PatioP2 : PERSIANAS ;
22        seguridad_p2 : SEGURIDAD_P2 ;
23        Horario_Semanal1 : HORARIOS_SEMANALES ;
24        Horario_Iluminacion : ConfiguracionHoraria ;
25        Conf_G4_P1_Z1 : INT := 1 ;
26        Pc_Planta_2 : IoT ;
27        Energia_Activa_P2 : REAL ;
28        Energia_Reactiva_P2 : REAL ;
29        Energia_Aparente_P2 : REAL ;
30    END_VAR
31
32    VAR PERSISTENT
33        stCTRL_PID_PARAMS : ST_CTRL_PID_SPLITRANGE_PARAMS ;
34    END_VAR
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```

16     Dia_Semana_3 := Dia_Semana_3 ,
17     Dia_Semana_4 := Dia_Semana_4 ,
18     Dia_Semana_5 := Dia_Semana_5 ,
19     Dia_Semana_6 := Dia_Semana_6 ,
20     Hora_Inicio := Hora_Inicio ,
21     Hora_Final := Hora_Final ,
22     Minuto_Inicio := Minuto_Inicio ,
23     Minuto_Final := Minuto_Final ,
24     Config1 => ) ;
25 (* Crearemos las otras configuraciones horarias.
26 ConfHor2 (
27     Mes_Inicio:= ,
28     Mes_Final:= ,
29     Dia_Inicio:= ,
30     Dia_Final:= ,
31     Dia_Semana_0:= ,
32     Dia_Semana_1:= ,
33     Dia_Semana_2:= ,
34     Dia_Semana_3:= ,
35     Dia_Semana_4:= ,
36     Dia_Semana_5:= ,
37     Dia_Semana_6:= ,
38     Hora_Inicio:= ,
39     Hora_Final:= ,
40     Minuto_Inicio:= ,
41     Minuto_Final:= ,
42     Config1=> );
43 *)
44
45 //Seleccionar la configuración Horaria
46 CASE Conf_FC1 OF
47     1: //Asignamos la configuración horaria 1
48         Configuración_Horaria := ConfHor1 . Config1 ;
49     (* Haríamos la asignación de la configuración horaria cuando
50     estuvieran creadas todas las configuraciones.
51     2: //Asignamos la configuración horaria 2
52         Configuración_Horaria:=ConfHor2.Config1;
53     3: //Asignamos la configuración horaria 3
54         Configuración_Horaria:=ConfHor.Config1;
55     4: //Asignamos la configuración horaria 4
56         Configuración_Horaria:=ConfHor.Config1;
57     *)
58 END_CASE
59
60 //Almacenamos los parámetros del PID
61 tIn_cool := INT_TO_TIME ( tIn_cooling ) * 1000 ;
62 tTv_cool := INT_TO_TIME ( tTv_cooling ) * 1000 ;
63 tIn_heat := REAL_TO_TIME ( tIn_heating ) * 1000 ;
64 tTv_heat := INT_TO_TIME ( tTv_heating ) * 1000 ;
65
66 stPID (

```

```

67     tCtrlCycleTime := tCtrlCycleTime ,
68     tTaskCycleTime := tTaskCycleTime ,
69     fKp_heating := fKp_heating ,
70     tTn_heating := tTn_heat ,
71     tTv_heating := tTv_heat ,
72     tTd_heating := tTd_heating ,
73     fKp_cooling := fKp_cooling ,
74     tTn_cooling := tTn_cool ,
75     tTv_cooling := tTv_cool ,
76     tTd_cooling := tTd_cooling ,
77     nParameterChangeCycleTicks := nParameterChangeCycleTicks ,
78     fOutMaxLimit := fOutMaxLimit ,
79     fOutMinLimit := fOutMinLimit ,
80     stCTRL_PID_PARAMS => ) ;
81 stCTRL_PID_PARAMS := stPID . stCTRL_PID_PARAMS ;
82 (*Repetiriamos la estructura anterior para los demás Fan Coils para los
parámetros del PID*)
83
84 //Almacenamos el valor de la temperatura a conseguir con el PID
85 IF Modo_FC1 = 1 THEN
86     Temp_Deseada_FC1 := Temp_Deseada_VER_FC1 ;
87     ELSE
88     Temp_Deseada_FC1 := Temp_Deseada_INV_FC1 ;
89 END_IF
90 (*Repetiriamos la estructura anterior para los demás Fan Coils*)
91
92 //Control de la Planta 1
93 // Programa para el control del FC1
94 CLIMATIZACION (
95     TempMaxSonReg := TempMaxSondaRegulacionFC1 ,
96     TempMinSonReg := TempMinSondaRegulacionFC1 ,
97     Sonda_Regulacion := aiFC1_STempReg ,
98     doFC_VMABRIR := doFC1_VMABRIR ,
99     doFC_VMCERRAR := doFC1_VMCERRAR ,
100    doFC_MARCHA := doFC1_MARCHA ,
101    diFC_EnMarcha := diFC1_EnMarcha ,
102    SondaINV_VER := aiSTempINV_VER ,
103    TempMaxSondaCambio := TempMaxSonLin ,
104    TempMinSondaCambio := TempMinSonLin ,
105    Habilita_Sonda := HabilitaSonda_P1 ,
106    TempCambModo := TempCambioModo ,
107    Histeresi_Sonda := HisteresiSonda ,
108    Modo_Manual_Inv_Ver := Modo_INVVER ,
109    Consigna := Temp_Deseada_FC1 ,
110    Modo_Fc := Modo_FC1 ,
111    TApertura := Tiempo_AperturaFC1 ,
112    Apertura_Deseada := Apertura_Deseada_FC1 ,
113    NoProgHoraria := OFFProg_Horaria ,
114    conf_Horaria0 := Configuracion_Horaria ,
115    stCTRL_PID_PARAMS := stCTRL_PID_PARAMS ,
116    HAY_ALM => ) ;

```

```

117 (*Repetiriamos la estructura anterior para el control de los Fan Coils*)
118
119 //Function Block para la gestión de las alarmas de la planta 1
120 EXT_FALLO_TERMICO (
121     FC1_EnMarcha := diFC1_EnMarcha ,
122     FC2_EnMarcha := diFC2_EnMarcha ,
123     FC3_EnMarcha := diFC3_EnMarcha ,
124     FC4_EnMarcha := diFC4_EnMarcha ,
125     ExtractorMarcha := diExtractoMarcha ,
126     FalloTermico := diFalloTermico ,
127     Alm_NoExtractorEnMarcha => ,
128     Alm_Fallo_Termico => ,
129     HAY_ALM => );
130 // Señal para activar la alarma de la pantalla principal
131 Hay_Alarma := CLIMATIZACION . HAY_ALM OR EXT_FALLO_TERMICO . HAY_ALM ;
132
133 (*Repetiriamos la estructura de la Planta para las otras 2*)
134
135 (*=====
136 =====*)
137 //Código para el control de la ILUMINACIÓN.
138 //Ejemplos de los usos de los Function Blocks de los diferentes Grupos.
139
140 //Código para la creación de las configuraciones horarias
141 Horario_Semanal1 (
142     Dia_Semana_0 := Dia_Semana_0I ,
143     Dia_Semana_1 := Dia_Semana_1I ,
144     Dia_Semana_2 := Dia_Semana_2I ,
145     Dia_Semana_3 := Dia_Semana_3I ,
146     Dia_Semana_4 := Dia_Semana_4I ,
147     Dia_Semana_5 := Dia_Semana_5I ,
148     Dia_Semana_6 := Dia_Semana_6I ,
149     Hora_Inicio := Hora_InicioI ,
150     Hora_Final := Hora_FinalI ,
151     Minuto_Inicio := Minuto_InicioI ,
152     Minuto_Final := Minuto_FinalI ,
153     Config1 => );
154
155 //Seleccionar la configuración Horaria para el Grupo 4, Plantal y Zonal
156 CASE Conf_G4_P1_Z1 OF
157     1 : //Asignamos la configuración horaria 1
158         Horario_Iluminacion := Horario_Semanal1 . Config1 ;
159     (* Haríamos la asignación de la configuración horaria cuando
160     estuvieran creadas todas las configuraciones.
161     2 : //Asignamos la configuración horaria 2
162         Horario_Iluminacion:=Horario_Semanal1.Config2;
163     3 : //Asignamos la configuración horaria 3
164         Horario_Iluminacion:=Horario_Semanal1.Config3;
165     4 : //Asignamos la configuración horaria 4
166         Horario_Iluminacion:=Horario_Semanal1.Config4;
167     *)

```

```

168 //Para los demás casos sería repetir el código anterior.
169 END_CASE
170
171 //Grupo 1
172 //Pasillo interior Planta 2, uso de 2 luminarias.
173 PasilloP2_G1 (
174     diPulsador_inhabilitar := diPulsadorInhabilitar ,
175     diPulsadorON_OFF := diPulsadorONOFF ,
176     config_Horario := Horario_Iluminacion ,
177     doLuminaria => ) ;
178 //Control de las dos luminarias
179 doLuminaria1 := PasilloP2_G1 . doLuminaria ;
180 doLuminaria2 := PasilloP2_G1 . doLuminaria ;
181
182 //Grupo 2
183 //Consultas médicas 1 Planta 1, uso de 1 luminaria y 1 sensor de presencia
184
185 Consulta1_P1 (
186     diPulsador_inhabilitar := diPulsadorInhabilitar2 ,
187     diPulsadorON_OFF := diPulsadorONOFF2 ,
188     diSensorPresencia := diSensorPresencia ,
189     aux_SensorPresencia := inhabilitarSensorPresencia ,
190     Tespera := Tespera1 ,
191     config_Horario := Horario_Iluminacion ,
192     doLuminaria => ) ;
193 //Control de las luminarias
194 doLuminaria3 := Consulta1_P1 . doLuminaria ;
195
196 //Grupo 3
197 //Iluminación Enfermería 2 Planta 1, uso de 1 luminaria y 1 sensor de
Luminosidad
198 Enfermeria2_P1 (
199     diPulsador_inhabilitar := diPulsadorInhabilitar3 ,
200     diPulsadorON_OFF := diPulsadorONOFF3 ,
201     aiSensorLuminosidad := aiSensorLuminosidad ,
202     umbral := umbral ,
203     HisteresisUmbral := histeresiLuminosidad ,
204     config_Horario := Horario_Iluminacion ,
205     doLuminaria => ) ;
206 //Control de las luminarias
207 doLuminaria4 := Enfermeria2_P1 . doLuminaria ;
208
209 //Grupo 4
210 //Iluminación Enfermería 3 Planta 1, uso de 1 luminaria, 1 sensor de luminosidad
y 1 sensor de presencia
211 Enfermeria3_P1 (
212     diPulsador_inhabilitar := diPulsadorInhabilitar4 ,
213     diPulsadorON_OFF := diPulsadorONOFF4 ,
214     aiSensorLuminosidad := aiSensorLuminosidad2 ,
215     umbral := umbral2 ,
216     HisteresisUmbral := histeresiLuminosidad2 ,

```

```

217     diSensorPresencia := diSensorPresencia2 ,
218     aux_SensorPresencia := inhabilitarSensorPresencia2 ,
219     Tespera := Tespera2 ,
220     config_Horario := Horario_Iluminacion ,
221     doLuminaria => );
222 //Control de las luminarias
223 doLuminaria5 := Enfermeria3_P1 . doLuminaria ;
224
225 //Control Persianas
226 //Laminas Persianas patio P2
227 PatioP2 (
228     aiSensorIluminacion := aiSensorIluminacion ,
229     maxFranja1 := maxFranja1 ,
230     maxFranja2 := maxFranja2 ,
231     maxFranja3 := maxFranja3 ,
232     maxFranja4 := maxFranja4 ,
233     maxFranja5 := maxFranja5 ,
234     HoraFinal1 := HoraFinal1 ,
235     HoraFinal2 := HoraFinal2 ,
236     HoraFinal3 := HoraFinal3 ,
237     HoraFinal4 := HoraFinal4 ,
238     HoraFinal5 := HoraFinal5 ,
239     PrioridadHorario := PrioridadHoraria ,
240     PrioridadSensor := PrioridadSensor ,
241     Apertura := Apertura ,
242     MANUAL := Manual ,
243     aoSistemaMotorizado => ) ;
244 //Control de las láminas de las persianas
245 aoSistemaMotorizado := PatioP2 . aoSistemaMotorizado ;
246
247
248 (*=====
249 =====*)
250 //Código para el control de la SEGURIDAD.
251 //Ejemplo del uso del function block de seguridad.
252
253 seguridad_p2 (
254     diDetectorHumo1 := diDetectorHumo1 ,
255     diDetectorHumo2 := diDetectorHumo2 ,
256     diDetectorHumo3 := diDetectorHumo3 ,
257     diDetectorHumo4 := diDetectorHumo4 ,
258     diDetectorHumo5 := diDetectorHumo5 ,
259     diPulsadorAlarma1 := diPulsadorAlarma1 ,
260     diPulsadorAlarma2 := diPulsadorAlarma2 ,
261     diPulsadorAlarma3 := diPulsadorAlarma3 ,
262     diPulsadorAlarma4 := diPulsadorAlarma4 ,
263     diDetectorPresencial := diDetectorPresencial ,
264     diDetectorPresencia2 := diDetectorPresencia2 ,
265     diDetectorPresencia3 := diDetectorPresencia3 ,
266     diDetectorPresencia4 := diDetectorPresencia4 ,
267     diDetectorPresencia5 := diDetectorPresencia5 ,

```

```

268     diDetectorPresencia6 := diDetectorPresencia6 ,
269     ActivoDetectoPresencia := ActivoDetectoPresencia ,
270     doSirena => );
271
272 //Control de la sirena de alarma
273 doSirena := seguridad_p2 . doSirena ;
274
275
276 (*=====
277 =====*)
278 //Código para el control de IoT.
279 //Ejemplo del uso del function block de IoT.
280 Energia_Activa_P2 := EL3423_P1 + EL3423_P2 + EL3423_P3 + EL3423_P4 + EL3423_P5 ;
281 Energia_Reactiva_P2 := EL3423_Q1 + EL3423_Q2 + EL3423_Q3 + EL3423_Q4 + EL3423_Q5 ;
282 Energia_Aparente_P2 := EL3423_S1 + EL3423_S2 + EL3423_S3 + EL3423_S4 + EL3423_S5 ;
283
284 Pc_Planta_2 ( Energ_Activa := Energia_Activa_P2 ,
285             Energ_Reactiva := Energia_Reactiva_P2 ,
286             Energ_Aparente := Energia_Aparente_P2 );
287

```

```

1  //attribute 'qualified_only'
2  VAR_GLOBAL
3  //PARÁMETROS GENERALES CLIMATIZACIÓN
4  NTotFC : INT := 3 ;
5  TempMinSonLin : REAL := - 50.0 ;
6  TempMaxSonLin : REAL := 100.0 ;
7  MaxConsignaTempCambio : REAL := 28.0 ;
8  MinConsignaTempCambio : REAL := 18.0 ;
9  ZonaMuerta : REAL := 2.0 ;
10 TempCambioModo : REAL := 20.0 ;
11 HisteresiSonda : REAL := 1.0 ;
12
13
14 //Planta 1 CLIMNTIZACIÓN
15 //Entradas Digitales
16 diFC1_EnMarcha AT %I* : BOOL ;
17 diFC2_EnMarcha AT %I* : BOOL ;
18 diFC3_EnMarcha AT %I* : BOOL ;
19 diFC4_EnMarcha AT %I* : BOOL ;
20 diExtractoMarcha AT %I* : BOOL ;
21 diFalloTermico AT %I* : BOOL ;
22 //Entradas analógicas
23 aiFC1_STempReg AT %I* : INT ; //Sondas de Regulación
24 aiFC2_STempReg AT %I* : INT ;
25 aiFC3_STempReg AT %I* : INT ;
26 aiFC4_STempReg AT %I* : INT ;
27 aiSTempINV_VER AT %I* : INT ; //Sonda de cambia de invierno a verano
28
29 //Temperaturas para asignar al PID (deberian ir en persistent para no tener
que reprogramarlas en caida de tensión)
30 Temp_Deseada_FC2 : LREAL ;
31 Temp_Deseada_FC3 : LREAL ;
32 Temp_Deseada_FC4 : LREAL ;
33 //Variables para asignar la apertura de la rejilla en modo manual
34 Apertura_Deseada_FC1 : INT ;
35 Apertura_Deseada_FC2 : INT ;
36 Apertura_Deseada_FC3 : INT ;
37 Apertura_Deseada_FC4 : INT ;
38 Modo_FC1 : INT ; //STOP/PID/MANUAL
39 Modo_FC2 : INT ;
40 Modo_FC3 : INT ;
41 Modo_FC4 : INT ;
42 Modo_INVVER : INT ;
43 HabilitaSonda_P1 : BOOL ; //Bit para habilitar la sonda de cambio invierno a
verano
44
45 OFFProg_Horaria : BOOL ; //Bit para desactivar la programación horaria de la
Climatización
46
47 //Variables para poder generar las programaciones horarias
48 Config_Actual : INT ;
49

```

```

50      //Salidas digitales
51      doFC1_VMABRIR AT %Q* : BOOL ;
52      doFC1_VMCERRAR AT %Q* : BOOL ;
53      doFC1_MARCHA AT %Q* : BOOL ;
54      doFC2_VMABRIR AT %Q* : BOOL ;
55      doFC2_VMCERRAR AT %Q* : BOOL ;
56      doFC2_MARCHA AT %Q* : BOOL ;
57      doFC3_VMABRIR AT %Q* : BOOL ;
58      doFC3_VMCERRAR AT %Q* : BOOL ;
59      doFC3_MARCHA AT %Q* : BOOL ;
60      doFC4_VMABRIR AT %Q* : BOOL ;
61      doFC4_VMCERRAR AT %Q* : BOOL ;
62      doFC4_MARCHA AT %Q* : BOOL ;
63      doExtractorMarcha AT %Q* : BOOL ;
64      doExtractorInvierno_Verano AT %Q* : BOOL ; //Variable para indicar el modo
1-INVIERNO, 0-VERANO

65
66
67      //EJEMPLOS ILUMINACIÓN
68      //EJEMPLO Grupo 1
69      //Pasillo interior Planta 2, uso de 2 luminarias.
70      //ENTRADAS DIGITALES
71      diPulsadorInhabilitar AT %I* : BOOL ; //inhabilita el pulsador manual de
ON/OFF
72      diPulsadorONOFF AT %I* : BOOL ;
73
74      //SALIDAS DIGITALES
75      doLuminaria1 AT %Q* : BOOL ;
76      doLuminaria2 AT %Q* : BOOL ;
77
78      //EJEMPLO Grupo 2
79      //Consultas médicas 1 Planta 1, uso de 1 luminaria y un sensor de presencia
80      //ENTRADAS DIGITALES
81      diPulsadorInhabilitar2 AT %I* : BOOL ; //inhabilita el pulsador manual de
ON/OFF
82      diPulsadorONOFF2 AT %I* : BOOL ;
83      diSensorPresencia AT %I* : BOOL ; //Variable para la entrada del sensor de
presencia

84
85      //SALIDAS DIGITALES
86      doLuminaria3 AT %Q* : BOOL ;
87
88      //Tiempo espera para los Sensores de Presencia
89      Tesperal : TIME ;
90      //Variable para el boton de inhabilitar el sensor de presencia
91      inhabilitarSensorPresencia : BOOL ;
92
93      // EJEMPLO Grupo 3
94      //Iluminación Enfermería 2 Planta 1, uso de 1 luminaria y 1 sensor de
Luminosidad
95      //ENTRADAS DIGITALES

```

```

96     diPulsadorInhabilitar3 AT %I* : BOOL ; //inhabilita el pulsador manual de
ON/OFF
97     diPulsadorONOFF3 AT %I* : BOOL ;
98     //ENTRADAS ANALÓGICAS
99     aiSensorLuminosidad AT %I* : INT ; //Variable para la entrada del sensor de
luminosidad
100
101     //SALIDAS DIGITALES
102     doLuminaria4 AT %Q* : BOOL ;
103     //Variable para definir el umbral de luminosidad y su histéresis
104     umbral : INT ;
105     histeresiLuminosidad : INT ;
106
107     // EJEMPLO Grupo 4
108     //Iluminación Enfermería 2 Planta 1, uso de 1 luminaria, 1 sensor de
Luminosidad y 1 sensor de presencia
109     //ENTRADAS DIGITALES
110     diPulsadorInhabilitar4 AT %I* : BOOL ; //inhabilita el pulsador manual de
ON/OFF
111     diPulsadorONOFF4 AT %I* : BOOL ;
112     diSensorPresencia2 AT %I* : BOOL ; //Variable para la entrada del sensor de
presencia
113     //ENTRADAS ANALÓGICAS
114     aiSensorLuminosidad2 AT %I* : INT ; //Variable para la entrada del sensor de
luminosidad
115
116     //SALIDAS DIGITALES
117     doLuminaria5 AT %Q* : BOOL ;
118     //Variable para definir el umbral de luminosidad y su histéresis
119     umbral2 : INT ;
120     histeresiLuminosidad2 : INT ;
121
122     //Tiempo espera para los Sensores de Presencia
123     Tespera2 : TIME ;
124     //Variable para el boton de inhabilitar el sensor de presencia
125     inhabilitarSensorPresencia2 : BOOL ;
126
127
128     // EJEMPLO Persianas
129     //Laminas Persianas patio P2
130     //ENTRADAS ANALÓGICAS
131     aiSensorIluminacion AT %I* : INT ;
132     //SALIDAS ANALÓGICAS
133     aoSistemaMotorizado AT %Q* : INT ;
134     //Franjas Horarias
135     maxFranja1 : INT ;
136     maxFranja2 : INT ;
137     maxFranja3 : INT ;
138     maxFranja4 : INT ;
139     maxFranja5 : INT ;
140     //Franjas Luminosidad

```

```

141 HoraFinal1 : WORD ;
142 HoraFinal2 : WORD ;
143 HoraFinal3 : WORD ;
144 HoraFinal4 : WORD ;
145 HoraFinal5 : WORD ;
146 //Auxiliares para la prioridad del control
147 PrioridadHoraria : BOOL ;
148 PrioridadSensor : BOOL ;
149 //Auxiliar para el control manual de la apertura de las persianas
150 Apertura : INT ;
151 Manual : BOOL ;
152
153 // EJEMPLO seguridad
154 //ENTRADAS DIGITALES
155 //Detectores de humo
156 diDetectorHumo1 AT %I* : BOOL ;
157 diDetectorHumo2 AT %I* : BOOL ;
158 diDetectorHumo3 AT %I* : BOOL ;
159 diDetectorHumo4 AT %I* : BOOL ;
160 diDetectorHumo5 AT %I* : BOOL ;
161 //Pulsadores de alarma
162 diPulsadorAlarma1 AT %I* : BOOL ;
163 diPulsadorAlarma2 AT %I* : BOOL ;
164 diPulsadorAlarma3 AT %I* : BOOL ;
165 diPulsadorAlarma4 AT %I* : BOOL ;
166 //Detectores de presencia
167 diDetectorPresencial AT %I* : BOOL ;
168 diDetectorPresencia2 AT %I* : BOOL ;
169 diDetectorPresencia3 AT %I* : BOOL ;
170 diDetectorPresencia4 AT %I* : BOOL ;
171 diDetectorPresencia5 AT %I* : BOOL ;
172 diDetectorPresencia6 AT %I* : BOOL ;
173 //SALIDAS DIGITALES
174 doSirena AT %Q* : BOOL ;
175 //Auxiliar para activar los detectores de presencia
176 ActivoDetectoPresencia : BOOL ;
177
178 //ANALIZADORES DE REDES
179 //PLANTA 2
180 //ANALIZADOR 1
181 EL3423_P1 AT %I* : REAL ;
182 EL3423_Q1 AT %I* : REAL ;
183 EL3423_S1 AT %I* : REAL ;
184 //ANALIZADOR 2
185 EL3423_P2 AT %I* : REAL ;
186 EL3423_Q2 AT %I* : REAL ;
187 EL3423_S2 AT %I* : REAL ;
188 //ANALIZADOR 3
189 EL3423_P3 AT %I* : REAL ;
190 EL3423_Q3 AT %I* : REAL ;
191 EL3423_S3 AT %I* : REAL ;

```

```

192     //ANALIZADOR 4
193     EL3423_P4 AT %I* : REAL ;
194     EL3423_Q4 AT %I* : REAL ;
195     EL3423_S4 AT %I* : REAL ;
196     //ANALIZADOR 5
197     EL3423_P5 AT %I* : REAL ;
198     EL3423_Q5 AT %I* : REAL ;
199     EL3423_S5 AT %I* : REAL ;
200 END_VAR
201
202 VAR_GLOBAL PERSISTENT
203     Configuracion_Horaria : ConfiguracionHoraria ;
204     Conf_FCl : INT ;
205
206     //Variables programación Horaria 1 CLIMATIZACIÓN
207     Mes_Inicio : SINT ;
208     Mes_Final : SINT ;
209     Dia_Inicio : WORD ;
210     Dia_Final : WORD ;
211     Dia_Semana_0 : BOOL ; //Domingo
212     Dia_Semana_1 : BOOL ;
213     Dia_Semana_2 : BOOL ;
214     Dia_Semana_3 : BOOL ;
215     Dia_Semana_4 : BOOL ;
216     Dia_Semana_5 : BOOL ;
217     Dia_Semana_6 : BOOL ; //Sabado
218     Hora_Inicio : WORD ;
219     Hora_Final : WORD ;
220     Minuto_Inicio : WORD ;
221     Minuto_Final : WORD ;
222
223
224     //Variables de Control PID
225     Temp_Deseada_FCl : LREAL ;
226     Temp_Deseada_VER_FCl : LREAL ;
227     Temp_Deseada_INV_FCl : LREAL ;
228     //Hay que pasar a al struct del PID
229     tCtrlCycleTime : TIME := T#10MS ;
230     tTaskCycleTime : TIME := T#10MS ;
231     fKp_heating : LREAL ;
232     //tTn_heating:TIME;
233     //tTv_heating:TIME;
234     tTn_heating : REAL ;
235     tTv_heating : INT ;
236     tTd_heating : TIME := T#0MS ;
237     fKp_cooling : LREAL ;
238     //tTn_cooling:TIME;
239     //tTv_cooling:TIME;
240     tTn_cooling : INT ;
241     tTv_cooling : INT ;
242     tTd_cooling : TIME := T#0MS ;

```

```

243     nParameterChangeCycleTicks : UDINT := 100 ;
244     fOutMaxLimit : LREAL := 1023 ;
245     fOutMinLimit : LREAL := 204 ;
246
247     //Variables de la sonda de temperatura del Fan Coil
248     TempMaxSondaRegulacionFC1 : LREAL ;
249     TempMinSondaRegulacionFC1 : LREAL ;
250     //Variable del Tiempo de apertura del Fan Coil
251     Tiempo_AperturaFC1 : REAL ;
252
253     //Variables programación Horaria 1 ILUMINACIÓN
254     Dia_Semana_0I : BOOL ; //Domingo
255     Dia_Semana_1I : BOOL ;
256     Dia_Semana_2I : BOOL ;
257     Dia_Semana_3I : BOOL ;
258     Dia_Semana_4I : BOOL ;
259     Dia_Semana_5I : BOOL ;
260     Dia_Semana_6I : BOOL ; //Sabado
261     Hora_InicioI : WORD ;
262     Hora_FinalI : WORD ;
263     Minuto_InicioI : WORD ;
264     Minuto_FinalI : WORD ;
265 END_VAR
266

```

DUT_CONFIGURACIONHORARIA_

```
1     TYPE ConfiguracionHoraria :
2     STRUCT
3         Mes_Inicio : SINT ;
4         Mes_Final : SINT ;
5         Dia_Inicio : WORD ;
6         Dia_Final : WORD ;
7         Dias_Semana : DWORD ;
8         Hora_Inicio : WORD ;
9         Hora_Final : WORD ;
10        Minuto_Inicio : WORD ;
11        Minuto_Final : WORD ;
12
13    END_STRUCT
14    END_TYPE
15
```


PRESUPUESTO

El presupuesto se compone de las siguientes tablas:

CÓDIGO	PRECIO(€/U)	UNIDADES(U)	TOTAL (€)
PC			
CP6606-0001-0020	590,00	3,00	1770
CP6602-0000-0020	1382,00	1,00	1382
			3152

Tabla 22. Presupuesto de material 1: PC.

CÓDIGO	PRECIO(€/U)	UNIDADES(U)	TOTAL (€)
LICENCIAS			
TC1200	75,00	4,00	300
TF2000	160,00	4,00	640
TF4100	170,00	4,00	680
TF6710	50,00	4,00	200
			1820

Tabla 23. Presupuesto de material 2: Licencias de software.

CÓDIGO	PRECIO(€/U)	UNIDADES(U)	TOTAL (€)
TARJETAS			
EL1008	50,12	15,00	751,8
EL2008	55,13	16,00	882,08
EL3054	171,84	20,00	3436,8
EL3204/EL3314	264,92	4,00	1059,68
EL4024	189,02	6,00	1134,12
EL3423	171,84	19,00	3264,96
EK1100	140,34	7,00	982,38
CU2005	121,72	1,00	121,72
EL9011	4,01	7,00	28,07
			11661,61

Tabla 24. Presupuesto de material 3: Tarjetas de los dispositivos.

PRESUPUESTO DE EJECUCIÓN DE MATERIAL (PEM)	COSTE (€)
PC	3152
LICENCIAS	1820,00
TARJETAS	11661,61
16633,61	

Tabla 25. Presupuesto de ejecución de material.

PRESUPUESTO TOTAL		COSTE (€)
PEM		16633,61
20% CARGAS FISCALES Y GENERALES		3326,722
6% BENEFICIO INDUSTRIAL		998,0166
PRESUPUESTO DE EJECUCIÓN POR CONTRATA (PEC)		20958,3486
7% PROYECTO Y DIRECCIÓN DE OBRA		1467,0844
	TOTAL	22425,433
TOTAL + 21% IVA		27134,7739

Tabla 26. Presupuesto total.

El presupuesto total asciende a veintisiete mil ciento treinta y cuatro Euros y setenta y siete céntimos.

PLIEGO DE CONDICIONES



CU2005 | 5-port Fast Ethernet Switch

The Beckhoff Ethernet Switches offer five RJ45 Ethernet ports. Switches relay incoming Ethernet frames to the destination ports. In full duplex mode, they prevent collisions. They can be used universally in automation and office networks. User-friendly installation via integrated DIN rail adapter.

The switches meet the special requirements of real-time-capable Industrial Ethernet solutions through several outstanding features:

- compact design in full metal housing
- half or full duplex, with automatic baud rate detection
- 10/100 Mbits/s Ethernet
- cross-over detection: automatic detection and correction of crossover and straight-through Ethernet cables
- clear, quick diagnosis, two LEDs for each Ethernet port
- fast DIN rail mounting
- industrial design

Technical data	CU2005
Bus system	all Ethernet (IEEE 802.3)-based protocols, store and forward switching mode, unmanaged
Number of Ethernet ports	5
Ethernet Interface	10BASE-T/100BASE-TX Ethernet with 5 x RJ45
Cable length	up to 100 m twisted pair
Data transfer rates	IEEE 802.3u auto-negotiation, half or full duplex, automatic settings
Hardware diagnosis	2 LEDs per channel (activity, link)
Power supply	24 (18...30) V DC, 3-pin connection (+, -, PE)
Weight	approx. 260 g
Dimensions (W x H x D)	approx. 73 mm x 100 mm x 30 mm
Operating/storage temperature	0...+55 °C/-25...+85 °C
Protect. class/Installation pos.	IP 20/variable

System	
Ethernet TCP/IP	For further Ethernet TCP/IP products please see the system overview .

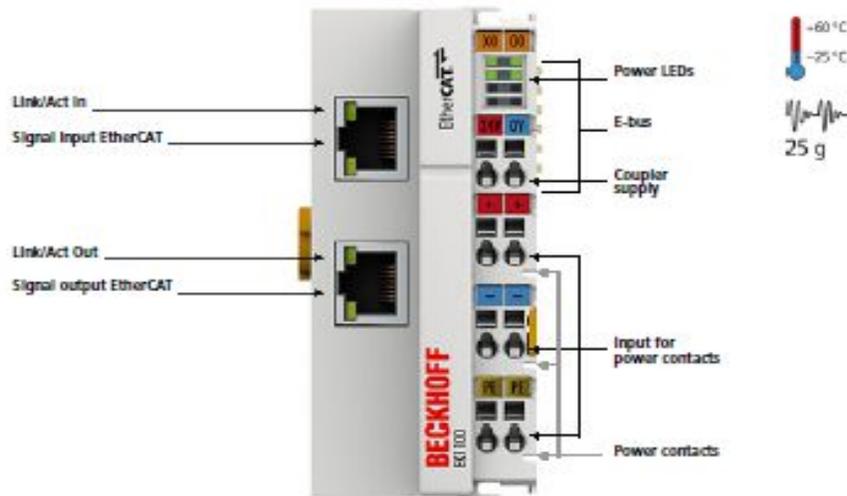
CP6606-0001-0020 | 7-inch "Economy" Panel PC

Variants	Processor	available
CP6606-0001-0020	ARM Cortex™-A8, 1 GHz (TC3: 30)	yes

*The TwinCAT 3 performance class defines the exact ordering number for the respective TwinCAT 3 product. Please see [here](#) for an overview of the TC3 performance classes.

CP6606-0001-0020	
Housing	<ul style="list-style-type: none"> aluminum front with steel sheet rear cover all connectors at the bottom of the rear side 1 slot for microSD flash card, accessible from the rear side lithium battery of the system clock, accessible from the rear side pull-out clamping levers for fast installation without loose parts protection class front side IP 54, rear side IP 20 operating temperature 0...55 °C
Front panel	<ul style="list-style-type: none"> 7-Inch TFT display, resolution 800 x 480 WVGA touch screen without keys
Features	<ul style="list-style-type: none"> processor ARM Cortex™-A8, 1 GHz (TC3: 30) 3½-Inch motherboard 1 GB DDR3 RAM graphic adapter Integrated Inside the processor 1 on-board Ethernet adapter 10/100BASE-T with RJ45 connector 1 on-board EtherCAT adapter with RJ45 connection 512 MB microSD flash card 1 serial RS232 ports and 2 USB 2.0 ports 24 V DC power supply Microsoft Windows Embedded Compact 7, English

Options	7-inch "Economy" Panel PC with ARM Cortex™-A8
C9900-H800	2 GB microSD card, instead of 512 MB microSD card
C9900-H801	4 GB microSD card, instead of 512 MB microSD card
C9900-H806	8 GB microSD card, instead of 512 MB microSD card
C9900-U213	Internal, capacitive 1-second UPS to ensure secure backup of persistent application data on the microSD flash card and 128 kbyte NOVRAM for fail-safe storage of TwinCAT process data, requires TwinCAT and Windows Embedded Compact 7
C9900-M317	wall mounting frame for building installation for 7" "Economy" built-in Panel PC CP6606
C9900-S706	TwinCAT 2 PLC runtime for Windows Embedded Compact 7
C9900-S707	TwinCAT 2 NC PTP runtime for Windows Embedded Compact 7
C9900-G070	<ul style="list-style-type: none"> Push-button extension for CP6606 with landscape 7-inch display – push-button extension on the bottom side – 3 push-button keys with signal lamp, type RAFI RAFIX 22FS+, round, 30 mm – 1 emergency stop key, type RAFI RAFIX 22FS+ – Labels for push-button caps for individual marking of each push-button can be ordered as an option. – The emergency stop key is wired with two normally-closed contacts, the red push-button with one normally-closed contact and the remaining push-buttons each with one normally-open contact to a terminal row. – Additionally, all push-buttons are transmitted with a normally-open contact via USB. – The LEDs of the push-buttons are controlled via USB only.
C9900-G071	<ul style="list-style-type: none"> Push-button extension for CP6606 with landscape 7-inch display – push-button extension on the bottom side – 3 push-button keys with signal lamp, type RAFI RAFIX 22FS+, round, 30 mm – 1 emergency stop key, type RAFI RAFIX 22FS+ – Labels for push-button caps for individual marking of each push-button can be ordered as an option. – The emergency stop key and the red push-button are wired each with two normally-closed contacts to a terminal row. The remaining push-buttons are wired each with two normally-open contacts to a terminal row. – The LEDs of the push-buttons are wired to a terminal row.



EK1100 | EtherCAT Coupler

The EK1100 coupler connects EtherCAT with the EtherCAT Terminals (ELxxxx). One station consists of an EK1100 coupler, any number of EtherCAT Terminals and a bus end terminal. The coupler converts the passing telegrams from Ethernet 100BASE-TX to E-bus signal representation.

The coupler is connected to the network via the upper Ethernet interface. The lower RJ45 socket may be used to connect further EtherCAT devices in the same strand. In the EtherCAT network, the EK1100 coupler can be installed anywhere in the Ethernet signal transfer section (100BASE-TX) – except directly at the switch. The couplers EK9000 and EK1000 are suitable for installation at the switch.

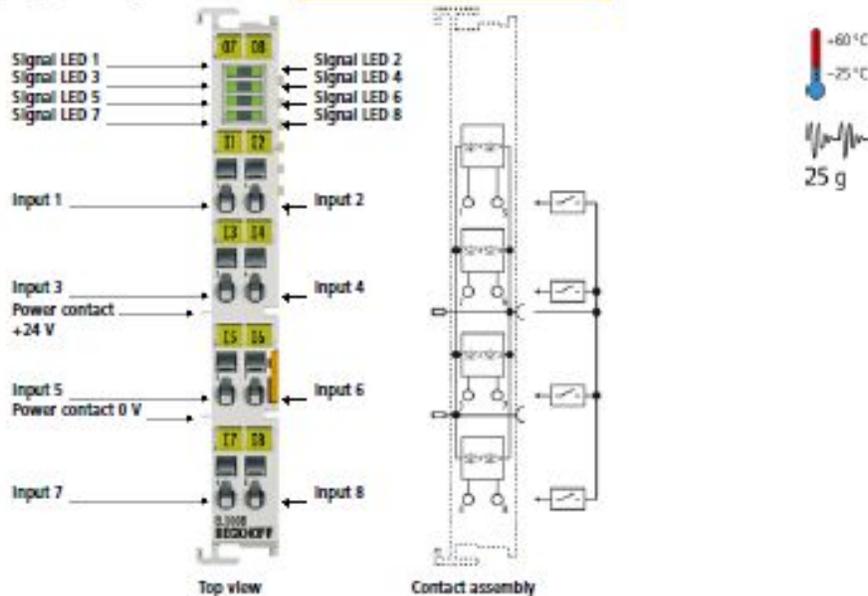
Technical data	EK1100
Task within EtherCAT system	coupling of EtherCAT Terminals (ELxxxx) to 100BASE-TX EtherCAT networks
Data transfer medium	Ethernet/EtherCAT cable (min. Cat. 5), shielded
Distance between stations	max. 100 m (100BASE-TX)
Number of EtherCAT Terminals	up to 65,534
Protocol	EtherCAT
Delay	approx. 1 µs
Data transfer rates	100 Mbit/s
Configuration	not required
Bus Interface	2 x RJ45
Power supply	24 V DC (-15 %/+20 %)
Current consumption from U _s	70 mA + (Σ E-bus current/4)
Current consumption from U _v	load
Current supply E-bus	2000 mA
Power contacts	max. 24 V DC/max. 10 A
Electrical Isolation	500 V (power contact/supply voltage/Ethernet)
Operating/storage temperature	-25...+60 °C/-40...+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC Immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protect. class/Installation pos.	IP 20/variable
Approvals	CE, UL, Ex, IECEx

Accessories

Cordsets cordsets and connectors

System

EtherCAT For further EtherCAT products please see the [system overview](#)



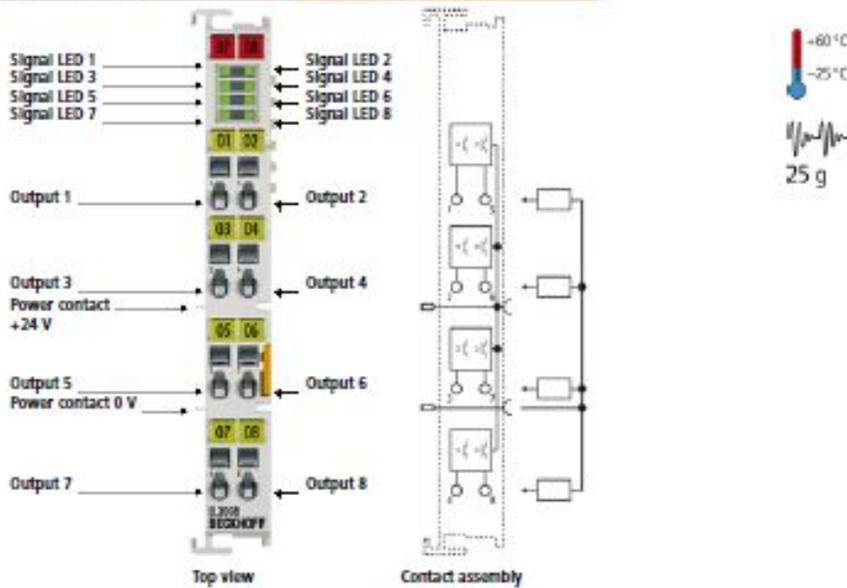
EL1008 | 8-channel digital input terminal 24 V DC, 3 ms

The EL1008 digital input terminal acquires the binary control signals from the process level and transmits them, in an electrically isolated form, to the higher-level automation unit. Digital input terminals from the EL100x series have a 3 ms input filter. The EtherCAT Terminals indicate their state via an LED.

Technical data	EL1008 ES1008
Connection technology	1-wire
Specification	EN 61131-2, type 1/3
Number of Inputs	8
Nominal voltage	24 V DC (-15 %/+20 %)
"0" signal voltage	-3...+5 V (EN 61131-2, type 3)
"1" signal voltage	11...30 V (EN 61131-2, type 3)
Input current	typ. 3 mA (EN 61131-2, type 3)
Input filter	typ. 3.0 ms
Distributed clocks	-
Current consumption power contacts	typ. 2 mA + load
Current consumption E-bus	typ. 90 mA
Electrical isolation	500 V (E-bus/field potential)
Bit width in the process image	8 Inputs
Configuration	no address or configuration setting
Special features	standard Input terminal for bouncing signals (filter 3 ms)
Weight	approx. 55 g
Operating/storage temperature	-25...+60 °C/-40...+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protect. class/Installation pos.	IP 20/see documentation
Pluggable wiring	for all ESxxxx terminals
Approvals	CE, UL, Ex, IECEx

Digital output

EL2008



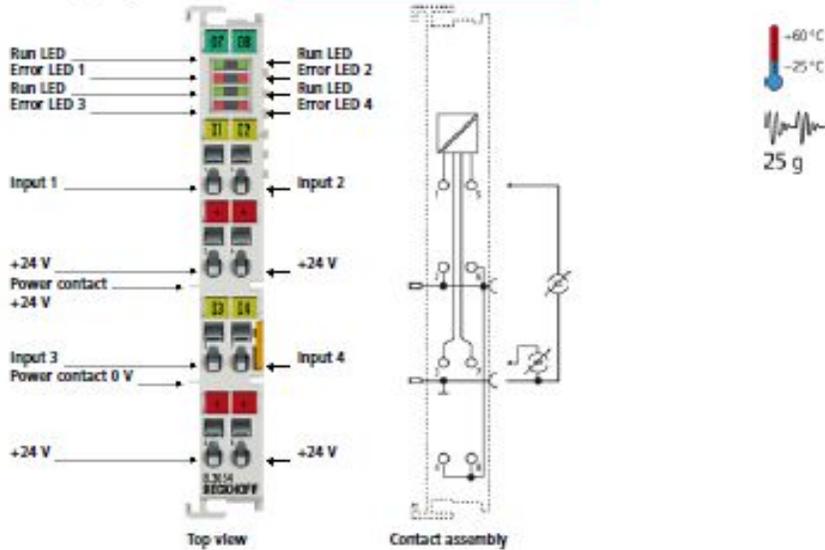
EL2008 | 8-channel digital output terminal 24 V DC, 0.5 A

The EL2008 digital output terminal connects the binary control signals from the automation unit on to the actuators at the process level with electrical isolation. The EtherCAT Terminal indicates its signal state via an LED.

Technical data	EL2008 ES2008
Connection technology	1-wire
Number of outputs	8
Nominal voltage	24 V DC (-15 %/+20 %)
Load type	ohmic, inductive, lamp load
Distributed clocks	-
Max. output current	0.5 A (short-circuit proof) per channel
Short-circuit current	typ. < 2 A
Reverse voltage protection	yes
Breaking energy	< 150 mJ/channel
Switching times	typ. T_{on} : 60 μ s, typ. T_{off} : 300 μ s
Current consumption E-bus	typ. 110 mA
Electrical isolation	500 V (E-bus/field potential)
Current consumption power contacts	typ. 15 mA + load
Bit width in the process image	8 outputs
Configuration	no address or configuration setting
Weight	approx. 55 g
Operating/storage temperature	-25...+60 °C/-40...+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protect. class/installation pos.	IP 20/see documentation
Pluggable wiring	for all ESxxxx terminals
Approvals/markings	CE, UL, ATEX, IECEx
Ex-Marking	II 3 G Ex nA IIC T4 Gc Ex nA IIC T4 Gc Ex tc IIIC T135 °C Dc

Analog input

EL3054



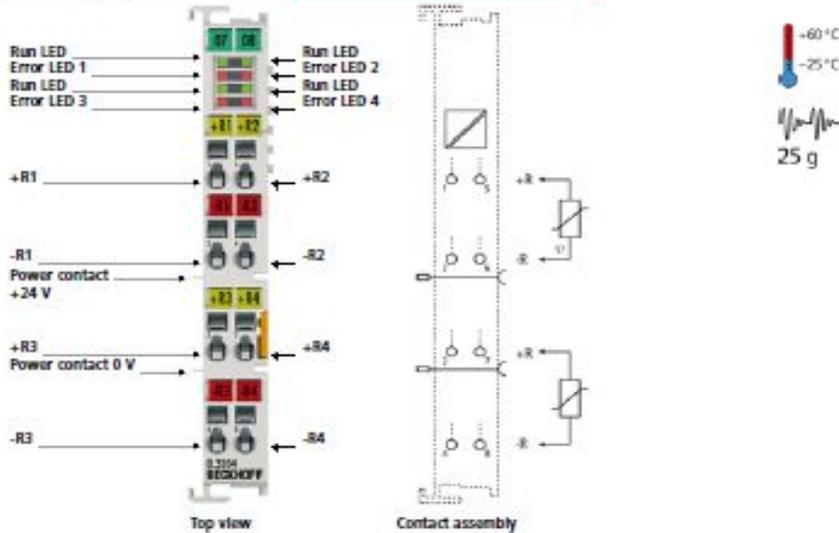
EL3054 | 4-channel analog input terminal 4...20 mA, single-ended, 12 bit

The EL3054 analog input terminal processes signals in the range between 4 and 20 mA. The current is digitised to a resolution of 12 bits and is transmitted (electrically isolated) to the higher-level automation device. The input electronics are independent of the supply voltage of the power contacts. In the EL3054 with four inputs, the 24 V power contact is connected to the terminal in order to enable connection of 2-wire sensors without external supply. The power contacts are connected through. The signal state of the EtherCAT Terminal is indicated by light emitting diodes. The error LEDs indicate an overload condition and a broken wire.

Technical data	EL3054 ES3054
Number of Inputs	4 (single-ended)
Technology	single-ended
Signal current	4...20 mA
Distributed clocks	–
Internal resistance	typ. 85 Ω
Input filter limit frequency	1 kHz
Dielectric strength	max. 30 V
Conversion time	0.625 ms default setting, configurable, multiplex
Resolution	12 bit (16 bit presentation incl. sign)
Measuring error	< ±0.3 % (relative to full scale value)
Electrical Isolation	500 V (E-bus/signal voltage)
Current consumption power contacts	–
Current consumption E-bus	typ. 130 mA
Bit width in the process image	Inputs: 16 byte
Configuration	no address or configuration setting required
Special features	standard and compact process image, activatable FIR/IR filters, limit value monitoring
Weight	approx. 60 g
Operating/storage temperature	-25...+60 °C/-40...+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC Immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protect. class/Installation pos.	IP 20/variable
Pluggable wiring	for all ESxxxx terminals
Approvals/markings	CE, UL, ATEX
Ex-Marking	II 3 G Ex nA IIC T4 Gc

Analog input

EL3204



EL3204 | 4-channel input terminal PT100 (RTD)

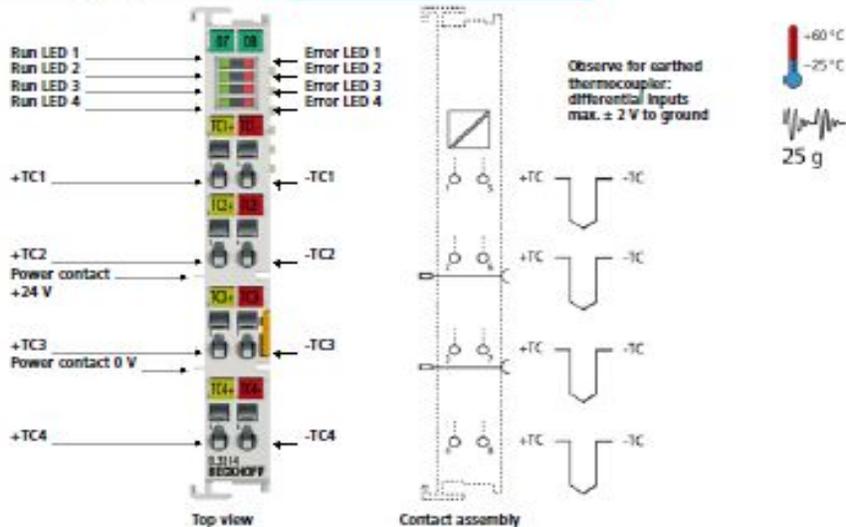
The EL3204 analog input terminal allows resistance sensors to be connected directly. The EtherCAT Terminal circuit can operate sensors using the 2-wire technique. A microprocessor handles linearisation across the whole temperature range, which is freely selectable. The EtherCAT Terminal's standard settings are: resolution 0.1 °C in the temperature range of PT100 sensors. Sensor malfunctions such as broken wires are indicated by error LEDs.

Technical data	EL3204 ES3204
Number of inputs	4
Power supply	via the E-bus
Distributed clocks	–
Input filter limit frequency	typ. 1 kHz
Sensor types	PT100, PT200, PT500, PT1000, NI100, NI120, NI1000 resistance measurement (e.g. potentiometer, 10 Ω...1/4 kΩ), KTY sensors (types see documentation)
Connection method	2-wire
Resolution	0.1 °C per digit
Conversion time	approx. 85 ms default setting, 2...800 ms configurable
Temperature range	-200...+850 °C (PT sensors); -60...+250 °C (NI sensors)
Measuring current	< 0.5 mA (load-dependent)
Measuring error	< ±0.5 °C for PT sensors
Electrical isolation	500 V (E-bus/signal voltage)
Current consumption power contacts	–
Current consumption E-bus	typ. 190 mA
Bit width in the process image	4 x 32 bit RTD Input
Special features	integrated digital filter, limit value monitoring
Weight	approx. 60 g
Operating/storage temperature	-25...+60 °C/-40...+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protect. class/Installation pos.	IP 20/variable
Pluggable wiring	for all ESxxxx terminals
Approvals/markings	CE, UL, ATEX
Ex-Marking	II 3 G Ex nA IIC T4 Gc

Special terminals

EL3204-0200

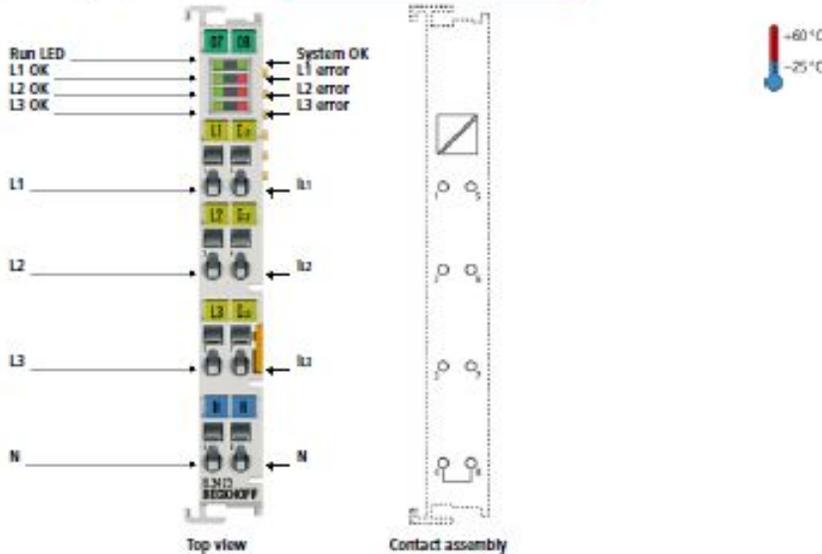
4-channel universal input terminal for RTD up to 240 kΩ, NTC 20 k, 16 bit



EL3314 | 4-channel thermocouple input terminal with open-circuit recognition

The EL3314 analog input terminal allows four thermocouples to be connected directly. The EtherCAT Terminal circuit can operate thermocouple sensors using the 2-wire technique. A microprocessor handles linearisation across the whole temperature range, which is freely selectable. The error LEDs indicate a broken wire. Compensation for the cold junction is made through an internal temperature measurement at the terminal. The EL3314 can also be used for mV measurement.

Technical data	EL3314
Number of inputs	4
Power supply	via the E-bus
Thermocouple sensor types	types K, J, L, E, T, N, U, B, R, S, C (default setting type K), mV measurement
Distributed clocks	–
Input filter limit frequency	typ. 1 kHz; depending on sensor length, conversion time, sensor type
Connection method	2-wire
Wiring fail indication	yes
Conversion time	approx. 2.5 s up to 20 ms, depending on configuration and filter setting, default: approx. 250 ms
Voltage measurement	± 30 mV... ± 75 mV
Temperature range	In the range defined in each case for the sensor (default setting: type K; $-200 \dots +1370$ °C); voltage measurement: ± 30 mV... ± 75 mV
Resolution	0.1 °C per digit
Measuring error	$< \pm 0.3$ % (relative to full scale value)
Electrical Isolation	500 V (E-bus/signal voltage)
Current consumption power contacts	–
Current consumption E-bus	typ. 200 mA
Bit width in the process image	4 x 32 bit TC input, 4 x 16 bit TC output
Configuration	no address setting, configuration via the controller
Special features	open-circuit recognition
Weight	approx. 60 g
Operating/storage temperature	$-25 \dots +60$ °C/ $-40 \dots +85$ °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protect. class/Installation pos.	IP 20/variable
Approvals/markings	CE, UL, ATEX, IIECEX
Ex-Marking	II 3 G Ex nA IIC T4 Gc Ex nA IIC T4 Gc Ex tc IIIC T135 °C Dc

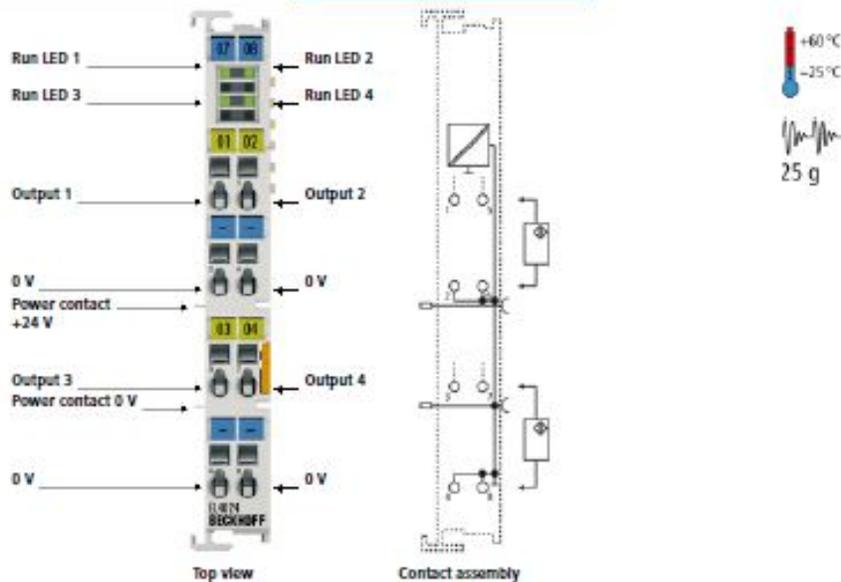


EL3423 | 3-phase power measurement terminal, Economy

The EL3423 EtherCAT Terminal enables the measurement of relevant data for an efficient energy management system. The voltage is measured internally via direct connection of L1, L2, L3 and N. The current of the three phases L1, L2 and L3 is fed via simple current transformers. The measured energy values are available separately as generated and accepted values. In the EL3423 Terminal, the effective power and the energy consumption for each phase are calculated. In addition, an internally calculated network quality factor provides information about the quality of the monitored voltage supply. The EL3423 offers basic functionality for mains analysis and energy management.

Technical data	EL3423
Number of inputs	3 x current, 3 x voltage
Technology	3-phase energy measurement
Oversampling factor	-
Distributed clocks	-
Update Interval	> 10 s
Update time	net-synchronous
Measured values	energy, power, power quality factor
Measuring voltage	max. 480 V AC 3~ (ULx-N: max. 277 V AC/240 V DC)
Measuring current	max. 1 A (AC/DC), via measuring transformers x A/1 A
Measuring error	0.5 % relative to full scale value (UI), 1 % calculated value
Monitoring function	phase order, phase failure, phase asymmetry, undervoltage/overvoltage (adjustable)
Electrical isolation	2500 V
Current consumption power contacts	-
Current consumption E-bus	typ. 120 mA
Special features	mains monitoring, single-phase operation possible
Weight	approx. 75 g
Operating/storage temperature	-25...+60 °C/-40...+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC Immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Approvals/markings	CE

Related products	
EL3443	3-phase power measurement terminal with extended functionality
EL3483	3-phase grid status terminal for voltage, frequency and phases



EL4024 | 4-channel analog output terminal 4...20 mA, 12 bit

The EL4024 analog output terminal generates signals in the range between 4 and 20 mA. The power is supplied to the process level with a resolution of 12 bits and is electrically isolated. Ground potential for the output channels of the EtherCAT Terminal is common with the 24 V DC supply. The output stages are powered by the 24 V supply. The EL4024 has four channels. The signal state of the EtherCAT Terminal is indicated by light emitting diodes.

Technical data	EL4024 ES4024
Connection technology	2-wire, single-ended
Number of outputs	4
Power supply	24 V DC via power contacts
Signal current	4...20 mA
Distributed clocks	yes
Distributed clock precision	<< 1 μs
Load	< 350 Ω (short-circuit proof)
Output error	< 0.1 % (relative to end value)
Resolution	12 bit
Electrical isolation	500 V (E-bus/signal voltage)
Conversion time	~ 250 μs
Current consumption power contacts	typ. 25 mA
Current consumption E-bus	typ. 140 mA
Bit width in the process image	4 x 16 bit AO output
Special features	Optional watchdog: user-specific output value with ramp; user synchronisation can be activated.
Weight	approx. 60 g
Operating/storage temperature	-25...+60 °C/-40...+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protect. class/installation pos.	IP 20/variable
Pluggable wiring	for all ESxxxx terminals
Approvals/markings	CE, UL, ATEX
Ex-Marking	II 3 G Ex nA IIC T4 Gc



EL9011 | Bus end cover

Each assembly must be terminated at the right hand end with an EL9011 bus end cap.

Technical data	EL9011
Technology	bus end cover
Current load	—
Power LED	—
Defect LED	—
Renewed Infeed	—
Current consumption E-bus	—
Integrated fine-wire fuse	—
Diagnostics in the process image	—
Reported to E-bus	—
PE contact	—
Shield connection	—
Connection facility to additional power contact	—
Bit width in the process image	0
Electrical connection to DIN rail	—
Housing width	5 mm
Side by side mounting on EtherCAT Terminals with power contact	yes
Side by side mounting on EtherCAT Terminals without power contact	yes
Special features	cover for E-bus contacts
Weight	approx. 10 g
Operating/storage temperature	-25...+60 °C/-40...+85 °C
Approvals/markings	CE, UL, ATEX
Ex marking	II 3 G Ex nA IIC T4 Gc