

Trabajo Fin de Grado

Grado en Ingeniería Electrónica, Robótica y
Mecatrónica.

Automatización de un parking mediante autómatas
programables y SCADA

Autor: Paloma Carrasco Fernández

Tutor: María del Mar Castilla Nieto

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

Sevilla, 2018



Trabajo Fin de Grado
Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Automatización de un parking mediante autómatas programables y SCADA

Autor:

Paloma Carrasco Fernández

Tutor:

María del Mar Castilla Nieto

Profesor Ayudante Doctor

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Trabajo Fin de Grado: Automatización de un parking mediante autómatas programables y SCADA

Autor: Paloma Carrasco Fernández

Tutor: María del Mar Castilla Nieto

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

A mi familia y amigos

Agradecimientos

En primer lugar, me gustaría agradecer a mi familia por haberme dado la oportunidad de estudiar esta carrera y formarme tanto profesionalmente como personalmente.

En segundo lugar, a los profesores, algunos por inculcarme la curiosidad de aprender y conseguir cada vez más cosas sólo por la satisfacción que produce y a otros por crearme la adicción al café tan necesaria para poder realizar los trabajos a tiempo y aprenderme el temario de los exámenes la noche antes.

Y, por último, a esas personas que han aparecido durante estos años y se han buscado un hueco para quedarse.

Paloma Carrasco Fernández

Sevilla, 2018

La automatización de procesos industriales ha dado lugar a avances significativos que han permitido a las compañías implementar procesos de producción más eficientes, seguros y competitivos gracias al desarrollo e innovación de las nuevas tecnologías.

La automatización industrial, es un conjunto de métodos y recursos que implican equipos físicos y programas destinados al control de dichos equipos con el propósito de llevar a cabo procesos industriales automáticos en un entorno de producción eficiente, que tiende a satisfacer la creciente demanda de bienes en intervalos de tiempo cada vez más reducidos.

Además, cabe destacar que en los anteriores años con la gran crisis que se ha sufrido y de la que aún se tienen restos, las PYMES prestadoras de servicios en el área de la Automatización manifestaron su adecuación a las condiciones de crisis manteniendo sus actividades incluso en algunos casos incrementándolas. Se ha presentado una capacidad innovadora, potenciando la flexibilidad para adaptarse y favoreciendo a los clientes el ahorro de energía, la integración de procesos de alta complejidad, integrando dispositivos de control que optimizaran procesos productivos y mejora de la gestión de la empresa, entre otros.

En los últimos años se ha estado desarrollando el sistema denominado SCADA, por medio del cual se pueden supervisar y controlar distintas variables que se presentan en un proceso o planta. Se deben usar periféricos, unidades remotas, etc., que le permiten al operador tener acceso completo al proceso mediante su visualización.

El objetivo de este proyecto es el diseño e implementación de la automatización para un parking. Asimismo, se ha desarrollado un sistema SCADA que permita supervisar e interactuar con él. Más concretamente, se propone la automatización del parking haciendo uso de dos PLCs de Siemens S7-1200 y dos pantallas SIMATIC HMI de Siemens también.

Para poder realizar dicha automatización, primeramente, se eligió un modelo de parking ya existente en la realidad tras un análisis de los diferentes tipos de parkings automatizados existentes. Posteriormente se ha propuesto una programación completa de los dispositivos para hacer un correcto funcionamiento de éste partiendo desde cero. La elaboración del SCADA, que permite al operario estar pendiente de todo lo que ocurre en el proceso e intervenir cuando lo considere necesario, obliga a estar comunicado con una interfaz propuesta para que el cliente elija lo que desea, de ahí el uso de las dos pantallas SIMATIC HMI. El uso de los PLCs es el que permite realizar en sí, el proceso de automatizar el parking. Por lo tanto, es evidente que, de esta forma se tienen que mantener comunicados tanto los PLCs como las pantallas HMI. Dicha comunicación se ha realizado utilizando PROFINET.

Finalmente, gracias a la disponibilidad de dichos dispositivos en el Laboratorio de Control de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla, se ha podido comprobar el correcto funcionamiento de la programación diseñada, y así proponer esta solución alternativa para la automatización de un parking que contenga un SCADA.

Abstract

The automation of industrial processes has led to significant advances that have enabled companies to implement more efficient, safe and competitive production processes, mainly due to the development and innovation of new technologies.

Industrial automation is a set of methods and resources that involve physical equipment and software designed to control such equipment in order to carry out automatic industrial processes in an efficient production environment, which tends to meet the growing demand for goods at ever shorter intervals of time.

In addition, it should be noted that in previous years, with the great crisis that has been suffered and from which there are still remains, SMEs providing services in the area of Automation expressed their adaptation to the crisis conditions by maintaining their activities even in some cases increasing them. It has presented an innovative capacity, enhancing flexibility to adapt and favoring energy saving, the integration of highly complex processes, integrating control devices that optimize production processes and improve the management of the company, among others.

In recent years, SCADA system concept has been developed, through which different variables that occur in a process or plant can be monitored and controlled. Peripherals, remote units, etc. must be used, which allow the operator to have full access to the process through its visualization.

The aim of this project is to design and implement an automation system for automation for a car park and to develop a SCADA system that allows monitoring and interacting with it. More specifically, to automate the car park using two Siemens S7-1200 PLCs and two Siemens SIMATIC HMI displays have been used.

Firstly, to be able to carry out this automation a parking model already existing in reality was chosen. Subsequently, a complete programming of the devices has been proposed in order to provide a correct functioning of the system from scratch. The development of SCADA, which allows the operator to keep an eye on everything that happens in the process and to intervene when he or she deems it necessary, requires communication with the proposed interface so that the customer can choose what he or she wants, hence the use of the two SIMATIC HMI screens. The use of PLCs is what allows the process of automating the car park itself to be carried out. Therefore, it is evident that, in this way, both PLCs and HMI screens must be kept in communication.

Finally, thanks to the availability of these devices in the Control Laboratory of the Superior Technical School of Engineering of the University of Seville, it has been possible to verify the correct functioning of the designed programming, and thus to propose this alternative solution for the automation of a car park containing a SCADA.

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Tablas	xvii
Índice de Figuras	xix
Notación	xxiii
1 Introducción	1
1.1. <i>Motivación</i>	1
1.2. <i>Objetivos del Trabajo Fin de Grado</i>	5
1.3. <i>Estructura del Trabajo Fin de Grado</i>	5
2 Hardware	7
2.1. <i>Componentes</i>	7
2.1.1. Pantallas HMI	7
2.1.2. PLC S7-1200	8
2.2. <i>Conexionado.</i>	9
2.3. <i>Función de cada dispositivo en el proyecto.</i>	10
3 Software	11
3.1. <i>Interfaz gráfica</i>	11
3.2. <i>Programación PLCs</i>	16
3.2.1. Lenguaje PLCs	17
3.2.2. Instrucciones empleadas PLCs	17
3.3. <i>Programación HMIs</i>	21
3.3.1. Comunicación entre HMI y PLC	23
4 Programación de los PLCs	25
4.1. <i>Configuración y conexión de la arquitectura implementada</i>	25
4.2. <i>Proyecto_FINAL PLCs</i>	26
4.2.1. PLC_usuario	26
4.2.2. PLC_administrador	38
5 Scada	45
5.1. <i>Concepto de Control</i>	45
5.2. <i>Sistemas de supervisión y mando</i>	46
5.2.1. Principios generales de diseño	47
5.2.2. Colores	48
5.3. <i>Principios generales de diseño</i>	49
5.3.1. Norma ISO 9241	49
5.3.2. Elementos gráficos	49
5.4. <i>Proyecto_FINAL pantallas HMI</i>	50
5.4.1. HMI_usuario	50

5.4.2. HMI_administrador	56
6 Mejoras implementadas	57
6.1. PLC_usuario	57
6.2. PLC_administrador	57
6.3. HMI_usuario	58
6.4. HMI_administrador	60
7 Conclusiones y futuros trabajos	61
Referencias	63
Anexos	65

ÍNDICE DE TABLAS

Tabla 3.1 Instrucciones básicas - General	17
Tabla 3.2 Instrucciones básicas - Operaciones lógicas con bits	17
Tabla 3.3 Instrucciones básicas - Contadores.	17
Tabla 3.4 Instrucciones básicas - Comparación.	18
Tabla 3.5 Instrucciones básicas - Temporizadores.	18
Tabla 3.6 Instrucciones básicas - Funciones matemáticas.	18
Tabla 3.7 Instrucciones avanzadas - String + Char.	18
Tabla 3.8 Comunicación – Open user communication	19
Tabla 3.9 Herramientas empleadas y eventos disponibles.	22
Tabla 4.1 Direcciones de los dispositivos del trabajo.	25
Tabla 4.2 Variables Segmento-1 PLC_usuario.	27
Tabla 4.3 Variables Segmento-2 PLC_usuario.	29
Tabla 4.4 Variables Segmento-2 PLC_usuario	33
Tabla 4.5 Tabla de codificación del envío.	34
Tabla 4.6 Variables Segmento-4 PLC_usuario.	35
Tabla 4.7 Variables Segmento-5 PLC_usuario.	38
Tabla 4.8 Variables Segmento-1 PLC_administrador.	39
Tabla 4.9 Variables Segmento-2 PLC_administrador.	41
Tabla 4.10 Variables Segmento-3 PLC_administrador.	41
Tabla 4.11 Variables Segmento-4 PLC_administrador.	42
Tabla 4.12 Variables Segmento-5 PLC_administrador.	43
Tabla 5.1 Colores según el estado.	49

ÍNDICE DE FIGURAS

Figura 1.1. Situación común en aparcamientos convencionales.	1
Figura 1.2 Tipo de parking con grúas.	2
Figura 1.3 Prototipo de parking con AGVs.	2
Figura 1.4 Tipo de parking simulando una noria.	3
Figura 1.5 Parking alemán.	3
Figura 1.6 Parking chino.	4
Figura 1.7 Boceto tipo de parking elegido.	4
Figura 2.1 Vista delantera SIMATIC HMI KTP700 Basic PN	7
Figura 2.2 Vista trasera SIMATIC HMI KTP700 Basic PN	8
Figura 2.3 PLC SIMATIC S7-1200 CPU 1214C AC/DC/Rly	8
Figura 2.4 Visualización real de la configuración del proyecto.	9
Figura 2.5 Conexionado total del proyecto.	9
Figura 3.1 Imagen inicial Tia Portal.	11
Figura 3.2 Vista del portal de Tia Portal.	12
Figura 3.3 Vista del proyecto de Tia Portal.	12
Figura 3.4 Agregar dispositivo.	13
Figura 3.5 Agregar PLC.	13
Figura 3.6 Vista previa del PLC.	14
Figura 3.7 Conexión de la SIMATIC HMI con el PLC.	14
Figura 3.8 Vista previa de la HMI.	15
Figura 3.9 Dispositivos y redes.	15
Figura 3.10 Iconos de compilar, transmitir e iniciar la simulación.	16
Figura 3.11 Icono de observación.	16
Figura 3.12 Vista del programa principal PLC (main).	16
Figura 3.13 TSEND_C.	19
Figura 3.14 TSEND_C Configuración.	19
Figura 3.15 Conexión entre PLCs.	19
Figura 3.16 Ejemplo parámetros de conexión TSEND_C.	20
Figura 3.17 Ejemplo parámetros del bloque TSEND_C.	20
Figura 3.18 Ejemplo parámetros del bloque TSEND_C.	20
Figura 3.19 Ejemplo parámetros de conexión TRCV_C.	21
Figura 3.20 Ejemplo parámetros del bloque TRCV_C.	21
Figura 3.21 Plantilla HMI.	22
Figura 3.22 Conexión de variables.	23
Figura 4.1 Conexión del Proyecto.	25

Figura 4.2 Programación Segmento-1 PLC_usuario.	26
Figura 4.3 Programación Segmento-2 PLC_usuario.	27
Figura 4.4 Programación Segmento-2 “Insertar matrícula” PLC_usuario.	28
Figura 4.5 Bloque de datos relacionados con la matrícula.	30
Figura 4.6 Aparcamientos por planta	30
Figura 4.7 Bloque de datos de la base de datos del parking.	31
Figura 4.8 Programación Segmento-3 PLC_usuario “Operaciones_base_datos”.	32
Figura 4.9 Programación Segmento-3 PLC_usuario.	32
Figura 4.10 Programación Segmento-4 PLC_usuario.	35
Figura 4.11 TSEND_C Parámetros de la conexión.	36
Figura 4.12 TSEND_C Parámetros del bloque.	36
Figura 4.13 Programación Segmento-5 PLC_usuario.	36
Figura 4.14 TRCV_C Parámetros de la conexión.	37
Figura 4.15 TRCV_C Parámetros del bloque.	37
Figura 4.16 Programación Segmento-1 PLC_administrador.	38
Figura 4.17 TRCV_C Parámetros de la conexión, administrador.	39
Figura 4.18 TRCV_C Parámetros del bloque, administrador.	39
Figura 4.19 Programación Segmento-2 PLC_administrador.	40
Figura 4.20 Programación Segmento-3 PLC_administrador.	41
Figura 4.21 Programación Segmento-4 PLC_administrador.	42
Figura 4.22 Programación Segmento-5 PLC_administrador	42
Figura 4.23 TSEND_C Parámetros de la conexión, administrador.	43
Figura 4.24 TSEND_C Parámetros del bloque, administrador.	43
Figura 5.1 Pirámide de Automatización.	45
Figura 5.2 Esquema imágenes HMI_usuario.	50
Figura 5.3 Plantilla HMI_usuario.	50
Figura 5.4 0_Imagen raíz.	51
Figura 5.5 Capa 0, campo entrada-salida de Insertar matrícula.	51
Figura 5.6 Capa 0, Insertar matrícula.	52
Figura 5.7 Capa 0 y 1, Insertar matrícula.	52
Figura 5.8 Comprobar matrícula.	53
Figura 5.9 Capa 0, 5_Aparcando.	54
Figura 5.10 Capa 1, 5_Aparcando.	54
Figura 5.11 Capa 2, 5_Aparcando.	54
Figura 5.12 Capa 0, 5_Sacando.	55
Figura 5.13 Capa 1, 5_Sacando.	55
Figura 5.14 Capa 2, 5_Sacando.	56
Figura 5.15 Única imagen HMI_administrador.	56
Figura 6.1 Modificación Segmento5 PLC_usuario.	57

Figura 6.2 Modificación Segmento5 PLC_administrador.	58
Figura 6.3 Nueva capa para errores, HMI_usuario.	59
Figura 6.4 Nueva capa para fallo, HMI_usuario.	59
Figura 6.5 Nueva imagen HMI_administrador	60

NOTACIÓN

AC	Corriente alterna
AGV	Vehículos de guiado automático
DC	Corriente continua
HMI	Interfaz humano-máquina
PLC	Controlador lógico programable
Rly	Relé
SCADA	Sistemas de supervisión, control y adquisición de datos
TFT	Transistor de películas finas
G	Giga- (prefijo= 10^9)
m	Mili-(prefijo= 10^{-3})
V	Voltios (Medida de tensión eléctrica)

1 INTRODUCCIÓN

We are just an advanced breed of monkeys on a minor planet of a very average star. But we can understand the Universe. That makes us something very special

- Stephen Hawking -

En este capítulo se trata de explicar porqué es importante este trabajo así como sus objetivos. También se podrá encontrar una estructura general del mismo.

1.1. Motivación

Encontrar aparcamiento, en numerosas ocasiones, se convierte en un ir y venir de vueltas hasta conseguir un hueco donde realizar numerosas maniobras para conseguir meter el coche.



Figura 1.1. Situación común en aparcamientos convencionales.

A medida que ha transcurrido el tiempo y con el fin de solucionar este problema se ha optado por la creación de parkings automatizados, donde lo único que tiene que hacer el usuario es introducir el coche e irse, encargándose el propio parking de aparcarlo o de sacarlo según se lo indique el usuario. Sin ir más lejos en Madrid, Barcelona y Vigo ya se encuentran parkings de este tipo, aunque en varias ocasiones se han encontrado problemas. Por ejemplo, en 2015, un parking automatizado localizado en Madrid obtuvo más de 90 averías en el mecanismo que baja y sube los coches. Incluso quedó paralizado dos días impidiendo la salida de los coches de su interior.

Aparte de automatizar el proceso de aparcamiento, otro de los objetivos de este tipo de parkings es la optimización del espacio disponible. Para ello, se han elaborado diferentes diseños con el fin de obtener la máxima capacidad posible, por eso hay numerosos tipos de parking. En general, se pueden encontrar aparcamientos independientes donde los vehículos se aparcen o recogen de forma independientes unos de otros, o aquellos en los que para introducir o sacar un vehículo hay que mover otro, denominados aparcamientos dependientes.

Sin embargo, cabe destacar también que hay numerosos tipos de parking porque hay varias formas de realizar el proceso de aparcarse o sacar un coche. Más concretamente, se pueden usar grúas parecidas a las convencionales encargadas de levantar el coche, AGVs que levanten los vehículos e incluso plataformas que se mueven simulando una noria, véase figuras 1.2 - 1.4.



Figura 1.2 Tipo de parking con grúas.



Figura 1.3 Prototipo de parking con AGVs.



Figura 1.4 Tipo de parking simulando una noria.

Cabe destacar que algunos de los parkings automáticos más futuristas son el de Alemania y el de China, véase figura 1.5 y 1.6, respectivamente. El alemán usa varias grúas para acceder a las distintas posiciones dónde se almacenan los coches, pudiendo mover más de un coche al mismo tiempo. Por otro lado, el parking el chino únicamente tiene una plataforma para mover el coche. Son similares en la idea de hacer una torre, aunque el primero es aéreo y el segundo subterráneo.



Figura 1.5 Parking alemán.



Figura 1.6 Parking chino.

En este proyecto, aunque se considera que el alemán es más útil por tener dos grúas independientes, se ha tomado como modelo el parking chino dado que el impacto visual es mucho menor al ser subterráneo y además es más compacto. En la siguiente figura se puede observar el boceto general del modelo elegido.



Figura 1.7 Boceto tipo de parking elegido.

Como se puede esperar en todos los parkings automatizados es necesario disponer de una serie de herramientas que permitan establecer una comunicación con el cliente. Asimismo, es de vital importancia comunicar y realizar en el menor tiempo posible las acciones que desea el cliente. Para poder tener una comunicación plena, lo más útil es usar un SCADA que permita supervisar y controlar el proceso.

De esta forma, se tendría una interfaz para que el cliente disponga su decisión y ésta se comunica con el parking para realizar el proceso avisando al sistema SCADA.

1.2. Objetivos del Trabajo Fin de Grado

El objetivo principal de este Trabajo Fin de Grado (TFG) es el diseño e implementación de un sistema de automatización que permita conseguir con éxito aparcar y sacar los vehículos de un parking tipo como el que se puede observar en la figura 1.7. Para ello, se han utilizado dos autómatas programables (PLC) con sus correspondientes pantallas táctiles que permiten interactuar con ellos. Más concretamente, se ha considerado una estructura distribuida con dos tipos de usuarios: cliente y operario del parking. Además, se ha diseñado una interfaz hombre-máquina (HMI) para el cliente del parking y un sistema SCADA que permite al operario supervisar todo el proceso. Para alcanzar la meta propuesta en este TFG ha sido necesario la consecución de cada uno de los objetivos que se detallan a continuación:

- Análisis de los diferentes parkings automatizados que se pueden encontrar en la literatura y selección de un parking tipo a automatizar.
- Diseño de la arquitectura que permita el control automático del parking así como la comunicación entre los diferentes elementos.
- Aprendizaje del programa, Tia Portal que va a permitir introducir el programa propuesto en los PLCs y en las HMIs. Dado que todos los dispositivos usados en el marco de este TFG son de Siemens, el uso de un programa de Siemens crea una facilidad a la hora de conectar y comunicar los dispositivos.
- Establecer una comunicación basada en Ethernet Industrial (PROFINET) entre los diferentes elementos que componen la arquitectura propuesta.
- Proponer un código para que uno de los PLCs junto con una de las pantallas HMI constituyan la interfaz del usuario capaz de comunicarse con el parking y que será utilizada por el cliente.
- Proponer otro código de PLC y programar otra HMI para realizar el proceso de automatización para aparcar y sacar los coches del parking así como crear el SCADA para que el operario, trabajador del parking, pueda realizar su trabajo.
- Comprobar el correcto funcionamiento total del conjunto.

En la realización del objetivo principal, conseguir la automatización completa con el SCADA, han aparecido otros objetivos, los cuales tratan de solucionar los problemas que han ido surgiendo. Por ejemplo, la creación de una base de datos para el parking o la correcta comunicación entre ambas partes del proyecto, cliente-operario.

1.3. Estructura del Trabajo Fin de Grado

A continuación, se muestra la estructura que conforma el Trabajo Fin de Grado.

- Capítulo 1. Introducción.
Dedicado a hacer una breve introducción sobre los parkings automatizados, indicando el elegido, los objetivos a realizar y este propio subapartado en el que se muestra la estructura del trabajo.
- Capítulo 2. Hardware.
Dedicado a describir brevemente los componentes que conforman el proyecto, así como la arquitectura que se ha implementado. También indica la función de los componentes principales dentro del trabajo.

- Capítulo 3. Software.
Dedicado explicar el uso del programa Tia Portal, la interfaz, la programación de los PLCs y pantallas HMI y funciones utilizadas durante el proyecto.
- Capítulo 4. Programación de los PLCs.
Encargado de mostrar la configuración y conexión llevada a cabo durante el trabajo y de explicar el programa propuesto para cada uno de los PLCs.
- Capítulo 5. Scada.
Breve introducción a los sistemas SCADA y la normativa existente para los mismos. Asimismo, se muestra la implementación realizada para la HMI del cliente y el SCADA del operario.
- Capítulo 6. Mejoras implementadas.
Modificación del proyecto para solucionar un error por falta de comunicación al cliente, se encarga de explicar dicho error y proponer los códigos de todos los elementos para la solución del mismo.
- Capítulo 7. Conclusiones y futuros trabajos.
Encargado de mostrar, como su nombre indica, la principal conclusión obtenida e indicar posibles mejoras y futuros trabajos relacionados.

2 HARDWARE

Este capítulo indica la configuración física que se ha realizado en el marco del proyecto. Para ello, se incluye una descripción general de los distintos componentes que lo conforman y el conexionado de éstos para su correcto funcionamiento. Además, se ha incluido una descripción de la función de los componentes principales.

2.1. Componentes

El proyecto consta de dos pantallas SIMATIC HMI de Siemens, dos PLC S7-1200 de Siemens, un switch SCALANCE XB005 de Siemens, 4 cables de red y una fuente de alimentación de 24V. En los apartados siguientes se va a realizar una breve descripción tanto de las pantallas como de los PLCs. Del resto de componentes solo cabe mencionar que el switch SCALANCE XB005 permite conectar los 4 dispositivos, así como conectarlo al router para que el ordenador pueda pasar los programas.

2.1.1 Pantallas HMI

Las pantallas de Siemens son las que han permitido crear una interfaz para que el usuario, cliente del parking, pueda decidir que opción desea realizar así como ver cuando se ha completado el proceso elegido. Además, estas pantallas también se han usado para crear el SCADA para que el operario o administrador, supervise todo el proceso y reaccione cuando considere necesario.

Más concretamente, se ha utilizado el modelo KTP700 Basic PN con referencia 6AV2123-2GB03-0AX0, la cual tiene una pantalla 7" TFT, 800 x 480 píxeles con 65536 colores. Cabe destacar como se puede observar en la figura 2.1 que se puede usar tanto de forma táctil como con los 8 botones que incluye. En la figura 2.2 se observa su entrada de alimentación, una entrada USB, y la conexión PROFINET¹. La entrada USB se usa como memoria y permite hasta 16 GB, pero no transmitir el programa mediante esta vía. La conexión PROFINET es la que permite comunicación entre la pantalla y otros dispositivos ya que es el protocolo usado. Para su conexionado con el resto de elementos es necesario saber que tiene que alimentarse a DC 24V.

Para mayor información ir al Anexo 1 Datasheet HMI KTP700 Basic PN.



Figura 2.1 Vista delantera SIMATIC HMI
KTP700 Basic PN

¹PROFINET: <https://www.profibus.com/technology/profinet/>



Figura 2.2 Vista trasera SIMATIC HMI KTP700 Basic PN

2.1.2 PLC S7-1200

Los PLCs son los que realizan todo el proceso. Por un lado, el PLC empleado para comunicarse con la pantalla que contiene la interfaz del usuario, será el encargado de recibir la matrícula e interactuar con ella en una base de datos y después mandar la información necesaria al otro PLC, que será el que avise de los errores al SCADA y realice toda la maniobra aparcar o sacar el coche. En el apartado 2.3 se explica su funcionamiento de forma más detallada.

El modelo de estos PLC es el CPU 1214C AC/DC/Rly, con referencia 6ES7214-1BG40-0XB0, véase la figura 2.3. Además, los PLCs utilizados llevan una Signal board AQ 1 x 12 bits con referencia 6ES7 232-4HA30-0XB0, aunque no se ha usado en el proyecto, así como una botonera como la que se puede observar en la figura 2.4 la cual sí que se ha usado para hacer la simulación de conexión de los sensores necesarios para aparcar o sacar el coche. Éstos, a diferencia de las pantallas que se alimentan con la fuente de alimentación, se conectan directamente a la red mediante un enchufe.

Para mayor información ir al Anexo 2 Datasheet S7-1200 CPU 1214C AC-DC-Rly.



Figura 2.3 PLC SIMATIC S7-1200 CPU 1214C
AC/DC/Rly

2.2. Conexionado.

Aunque en la figura 2.4 se tiene una visualización de los elementos del proyecto así como su conexionado, en la figura 2.5 se muestra un esquema de la arquitectura completa creada para este TFG. Más concretamente, se ha incluido la fuente de alimentación, junto con un mayor detalle en lo referente al cableado.

Por un lado, en **negro** se tienen las conexiones a la red eléctrica de los enchufes de los PLCs. También en **negro**, aunque mas fino, se tiene la tierra que se conecta desde la fuente de alimentación de 24V para alimentar las pantallas HMI y el switch. Así como el **rojo**, alimentación que sale desde la fuente. También se puede observar la conexión a red eléctrica de la fuente de alimentación.

Por otro lado, en **verde**, se encuentran los cables PROFINET que permiten la comunicación vía Ethernet Industrial. A partir de ahora para referirse al PLC que contiene el código del usuario y a la HMI que el cliente ve, se le denotará por PLC_usuario y HMI_usuario, respectivamente. De igual forma, para el operario, PLC_administrador y HMI_administrador.

Por ultimo, en **gris**, se encuentra el cable que conecta el switch al router permitiendo la comunicación entre el ordenador y los distintos elementos, dando así la opción de insertar los programas en los componentes.



Figura 2.4 Visualización real de la configuración del proyecto.

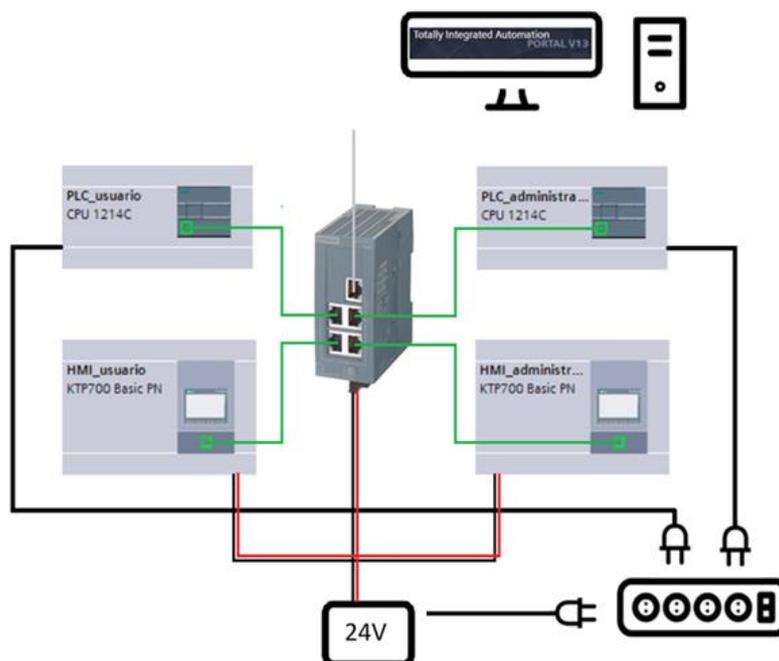


Figura 2.5 Conexionado total del proyecto.

2.3. Función de cada dispositivo en el proyecto.

Se va a usar este apartado para indicar de forma generalizada lo que van a hacer cada uno de los 4 componentes principales del proyecto, es decir, PLC_usuario, PLC_administrador, HMI_usuario y HMI_administrador.

- **PLC_usuario.**

Éste dispositivo es el encargado de, primeramente, almacenar la opción elegida por el usuario. Esto quiere decir que en sus salidas va a estar indicado si el cliente desea aparcar o sacar el vehículo. Además, también tiene que almacenar la matrícula indicada y realizar ciertas operaciones en la base de datos que tiene guardada en su interior. Esto sirve para que cuando se desee aparcar el coche, siempre sea en la primera plaza que este vacía empezando por la más próxima y así conseguir que se tarde lo menos posible. Y que cuando se desee sacar el coche, se busque la matrícula correspondiente y así se encuentre la plaza en la que se encuentra.

Se tiene que encargar de mandar la plaza al PLC_administrador realizando una codificación para saber si lo que se quiere es sacar o aparcar el coche. Y debe recibir por parte del PLC_administrador la variable que indique si ya se ha realizado la opción elegida por el usuario.

- **PLC_administrador.**

En primer lugar debe recibir la codificación enviada por el PLC_usuario, descodificarla para así saber la opción que debe realizar, y realizar el proceso deseado. En el caso de encontrar algún tipo de error, deberá asignarlo a las variables correspondientes, para que se quede constancia de ellos.

Por ultimo, cuando el proceso esté finalizado, enviará al PLC_usuario las variables correspondientes para indicar al cliente que ya puede abandonar el parking porque todo ha ido correctamente.

- **HMI_usuario.**

Esta pantalla contiene la interfaz para que el usuario elija la opción que desee e introduzca su matrícula. La comunicación con el PLC_usuario, hace posible que se almacene en la base de datos correctamente. Además cuando el PLC_usuario reciba que el proceso ha finalizado, ésta debe indicárselo al cliente mediante HMI_usuario.

- **HMI_administrador.**

Es la que realiza el SCADA que permite al operario interactuar al aparecer los errores. Indica mediante leds los procesos que se están realizando y la aparición de errores. Además permite parar el proceso y llevarlo a la posición inicial hasta solucionar los errores que han aparecido.

Se puede encontrar una mayor descripción de los programas que se han implementado en los PLCs en el capítulo 4 y de las pantallas HMI en el capítulo 5.

3 SOFTWARE

Siemens vende el software Tia Portal como la clave para liberar todo el potencial de Totally Integrated Automation. El software optimiza todos sus procedimientos de procesamiento, operación de máquinas y planificación. Asegurando su facilidad de uso gracias a la interfaz de usuario. Por ello, en este capítulo, se va a hacer una descripción de la interfaz, así como el lenguaje usado para el proyecto, funciones propias del programa y cómo realizar la conexión de los elementos.

Para que este capítulo sea lo más útil posible, se ha creado un nuevo proyecto (ProyectoPrueba) para poder adjuntar las imágenes paso a paso. El proyecto completo se explicará en los capítulos 4 y 5.

3.1. Interfaz gráfica

Cuando se abre por primera vez el programa Tia Portal, se encuentra una pantalla tal y como se muestra en la figura 3.1. Esta nos da opción a crear un nuevo proyecto, abrir uno ya creado anteriormente o migrar un proyecto.

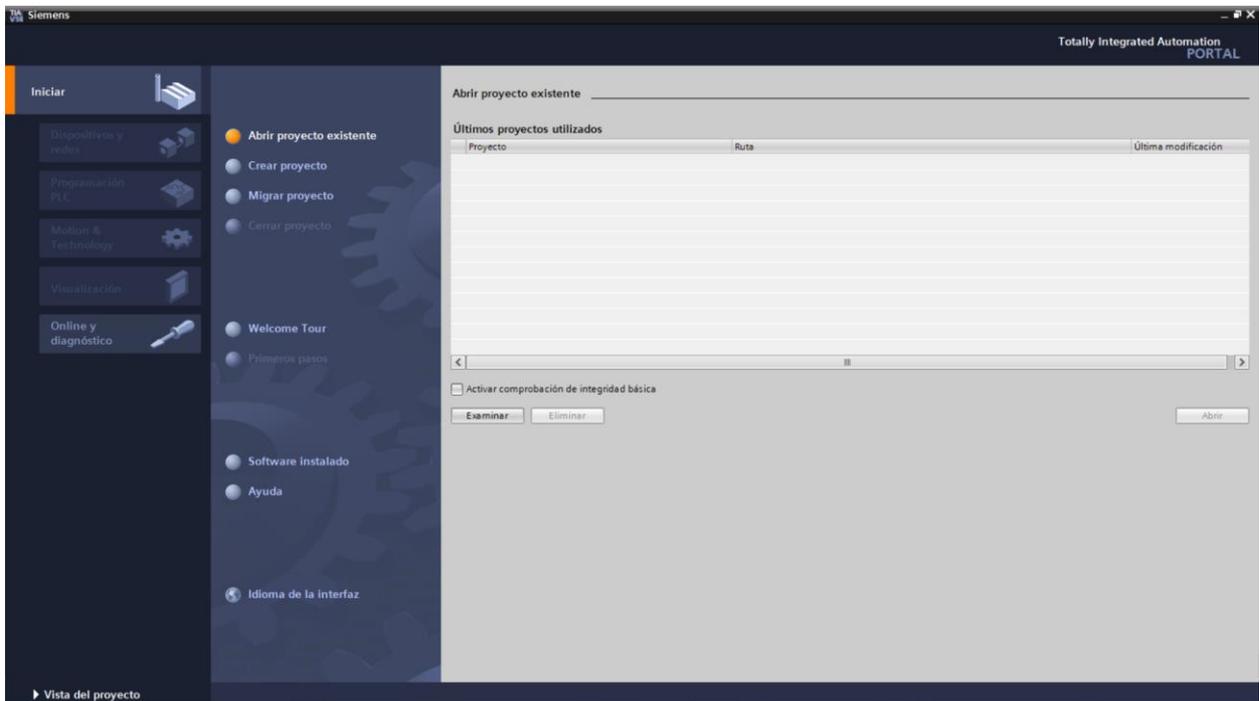


Figura 3.1 Imagen inicial Tia Portal.

La opción Welcome Tour redirige a la página de Siemens donde hay información y vídeos sobre el uso del programa. También en esta primera pantalla se puede ver la versión instalada del programa, mostrar la ayuda al usuario y cambiar el idioma de la interfaz si se desea.

Tanto eligiendo la opción de abrir proyecto existente como crear un proyecto, la pantalla que sale es la figura 3.2. Aunque todas las opciones que dan también son accesibles dando a la vista previa del proyecto, la cual se corresponde con la figura 3.3.

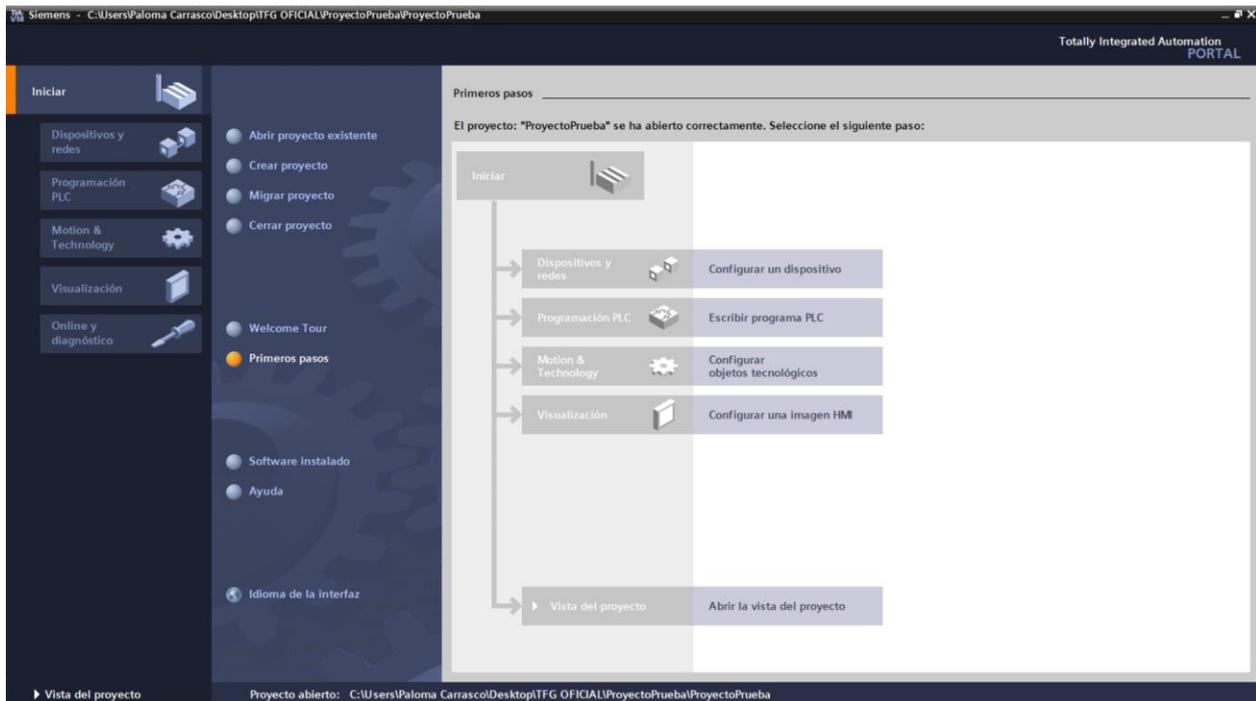


Figura 3.2 Vista del portal de Tia Portal.

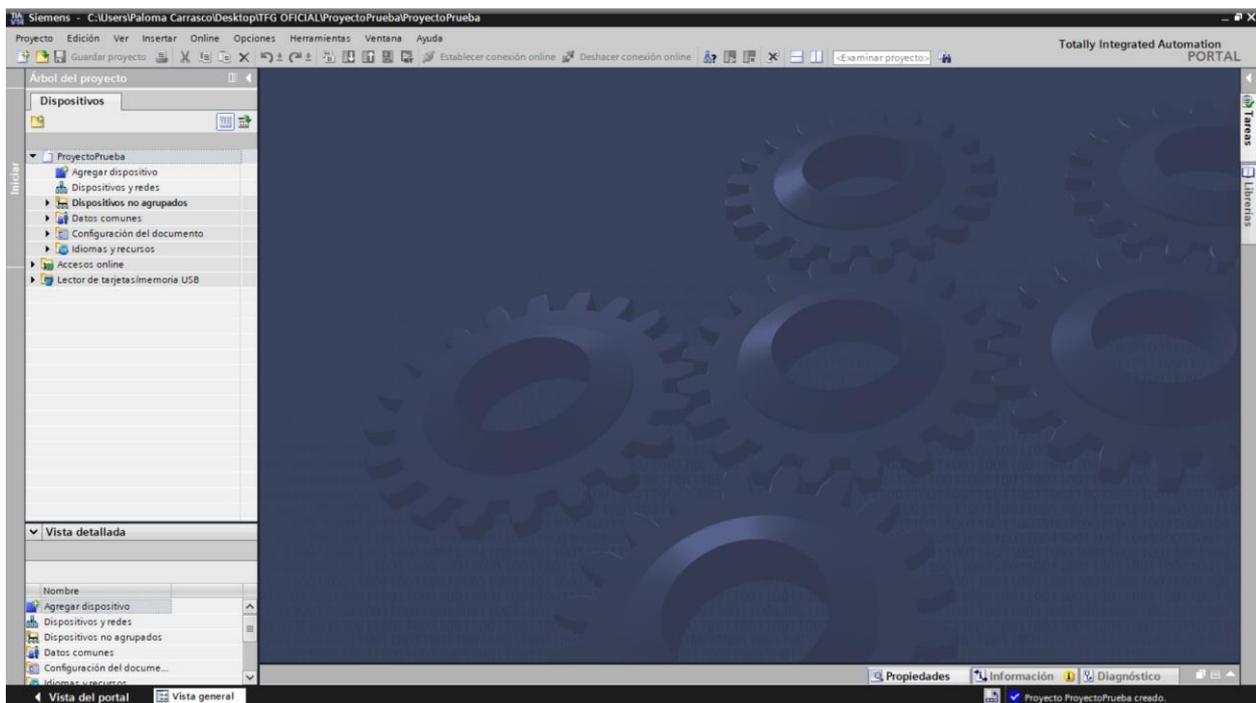


Figura 3.3 Vista del proyecto de Tia Portal.

Como se puede observar, una vez en la vista del proyecto, es posible añadir dispositivos y crear redes entre ellos. Para ilustrar el procedimiento seguido para la configuración utilizada en este trabajo fin de grado, se va a realizar una prueba para ver como se agregaría tanto un PLC como una pantalla HMI. Primero se va a agregar el PLC, ya que como se verá mas adelante, a la hora de añadir una pantalla SIMATIC HMI, ésta se enlaza desde el inicio al PLC estableciendo así una comunicación entre ellos.

Por lo tanto, comenzando con el PLC se tiene que clicar en agregar dispositivo. Aparecerá entonces una pantalla como la de la figura 3.4. Para añadir el PLC se tienen dos opciones, se puede añadir directamente con el modelo exacto o se puede no especificarlo y usar la determinación del propio programa para asignar directamente el

dispositivo que se encuentra realmente conectado. En el caso de este proyecto, resulta más interesante agregar el dispositivo sin especificar el modelo ya que la hacer la determinación incluye directamente la Signal Board que contienen los PLCs del laboratorio. Ahorrando el paso de tener que determinarlo de forma manual.

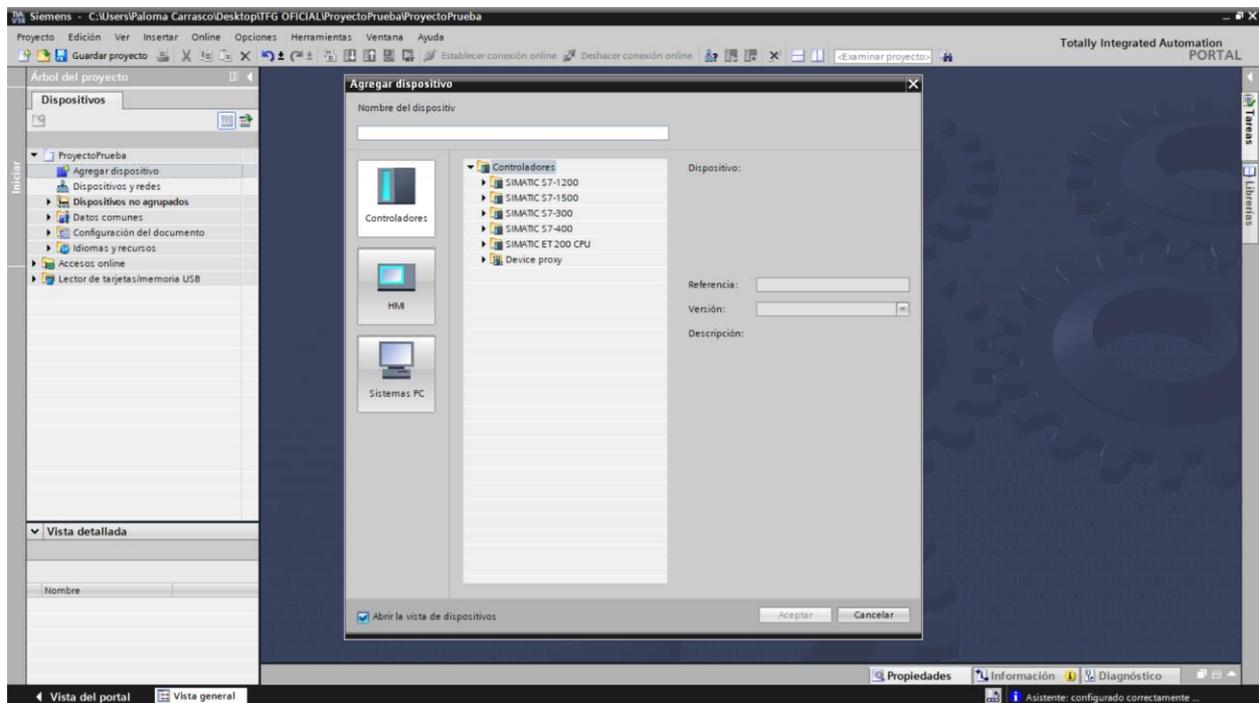


Figura 3.4 Agregar dispositivo.

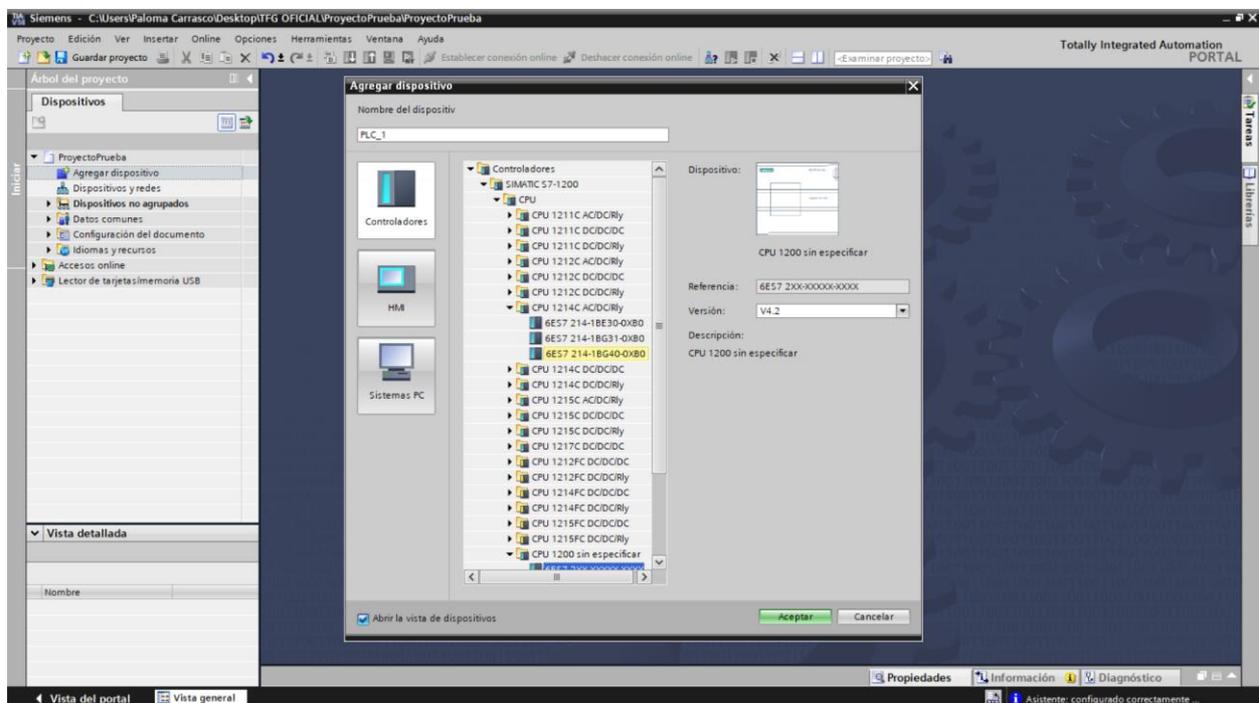


Figura 3.5 Agregar PLC.

Una vez agregado, la vista es la observada en la figura 3.6. En dicha vista se puede observar la configuración del dispositivo para administrarle la dirección correcta, los bloques de programa, variables, tipos de datos ...

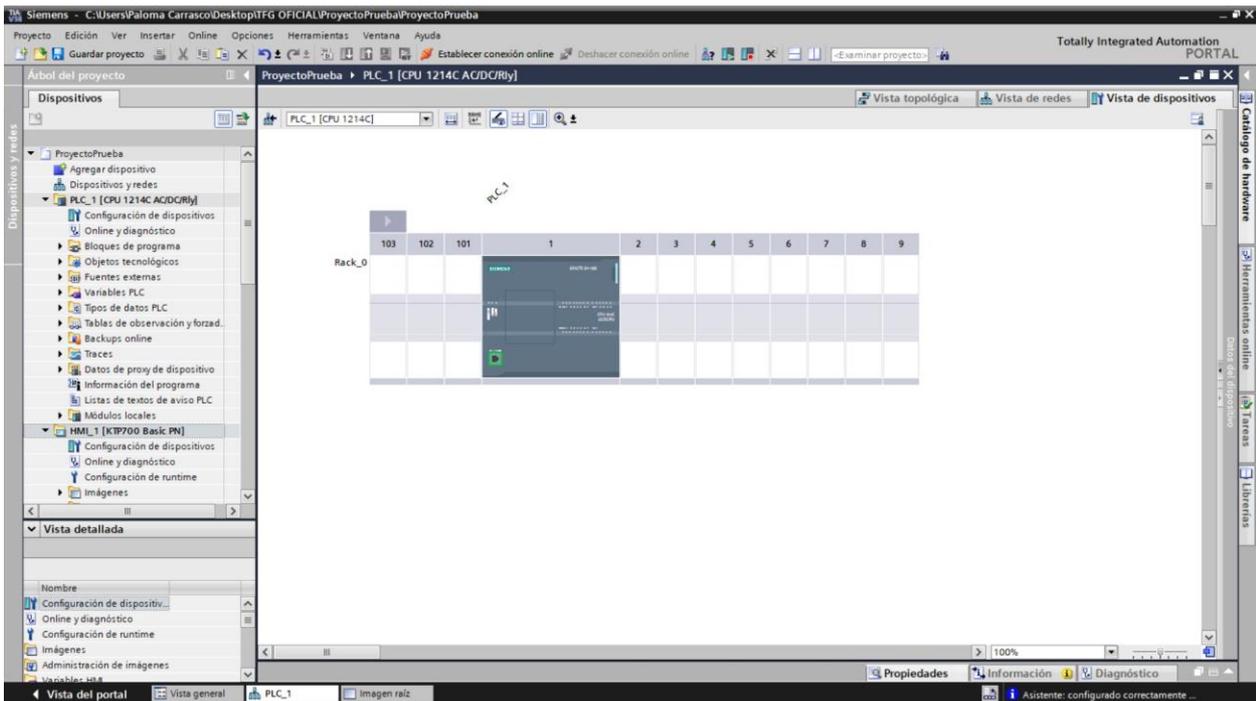


Figura 3.6 Vista previa del PLC.

De igual forma, se agrega la pantalla SIMATIC HMI y realizando los pasos indicados se puede conectar con el PLC tal y como se muestra en la figura 3.7.

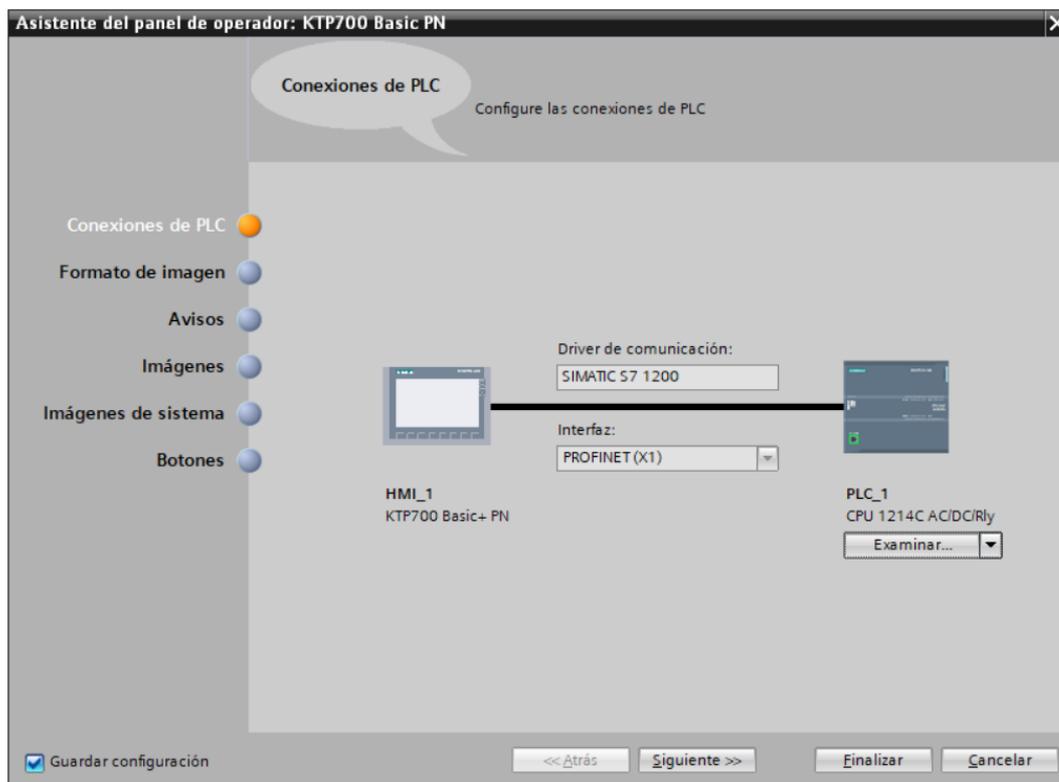


Figura 3.7 Conexión de la SIMATIC HMI con el PLC.

Tras finalizar todos los pasos, se tiene la vista observada en la figura 3.8, pudiendo modificar su configuración, imágenes, variables, etc.

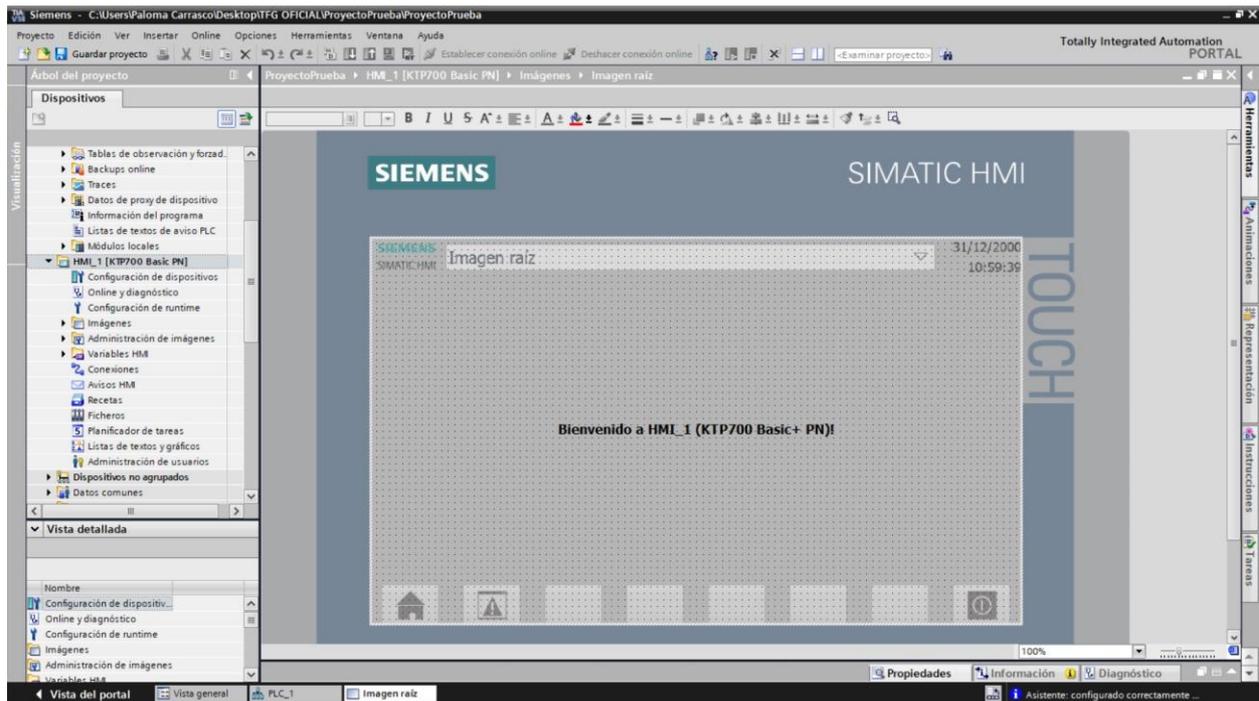


Figura 3.8 Vista previa de la HMI.

Tras finalizar lo anterior, en el apartado de dispositivos y redes se pueden consultar las distintas conexiones que hay. En la figura 3.9 se observa que se unen mediante una conexión PN/IE. Esto va a permitir que a la hora de programar la pantalla, se pueda establecer una conexión entre las variables de la pantalla y las del PLC.

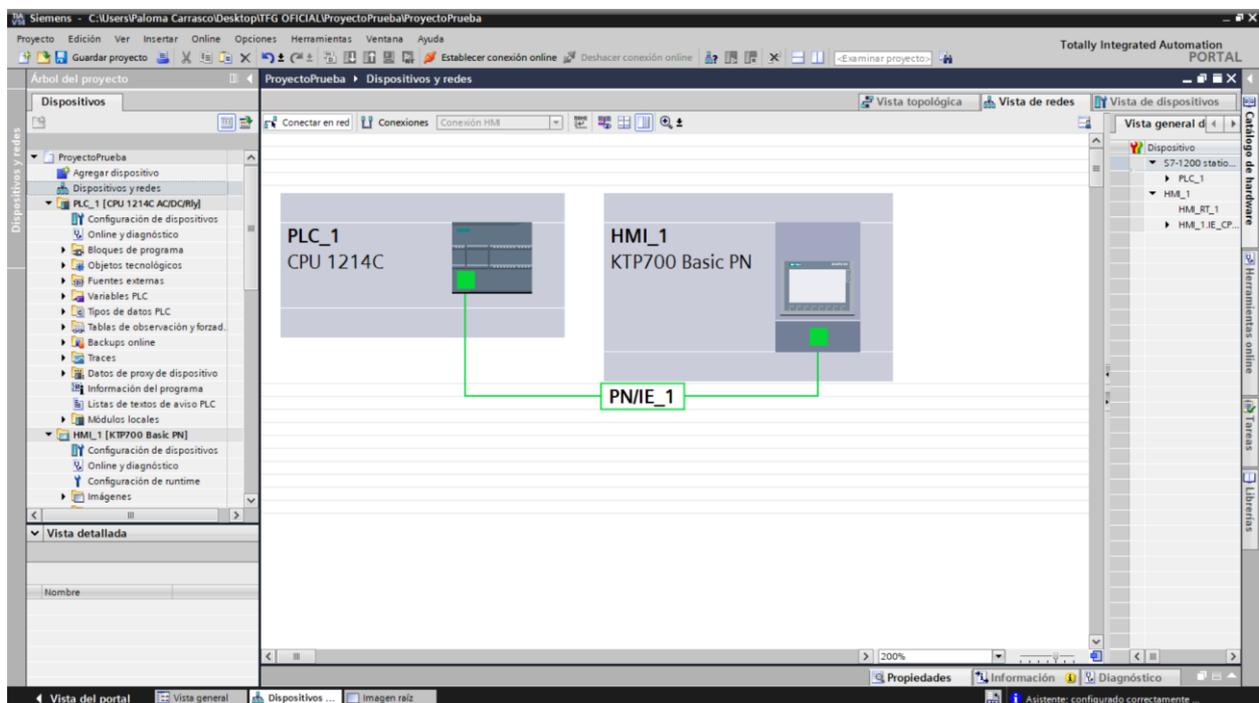


Figura 3.9 Dispositivos y redes.

Siempre se debe tener a vista el icono de compilar y transmitir al dispositivo. Si no disponemos de los dispositivos físicos, se puede utilizar el icono que permite iniciar la simulación. Se puede acceder a ellos desde cualquier pantalla ya que se encuentran en la barra de herramientas superior, véase la figura 3.10.



Figura 3.10 Iconos de compilar, transmitir e iniciar la simulación.

Cabe mencionar la utilidad de la conexión online a la hora de hacer un programa en un PLC, esta conexión que se establece con el dispositivo permite ver que ocurre a tiempo real en el programa gracias al icono mostrado en la figura 3.11. Además, esta opción también permite alertar sobre cualquier incidente con el dispositivo.

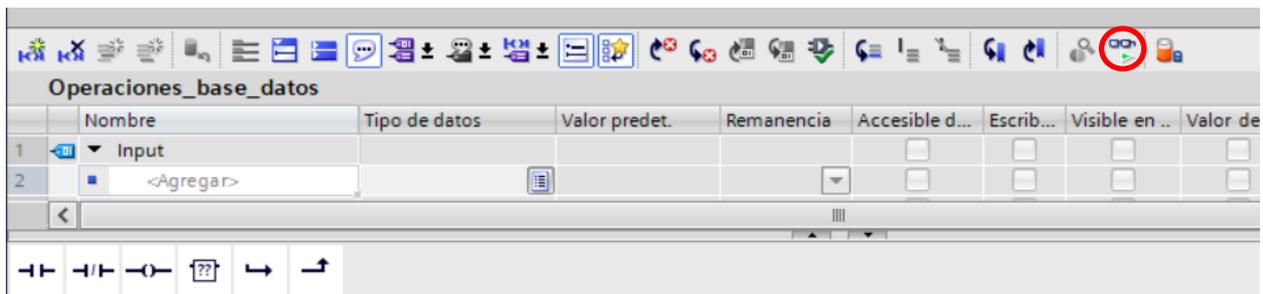


Figura 3.11 Icono de observación.

3.2. Programación PLCs

Para programar un PLC hay que tener en cuenta que existen distintos tipos de bloques. Existen bloques de organización (OB), como el main que se puede observar en la figura 3.12, que se ejecutan cíclicamente y dentro de ellos se pueden llamar a otros bloques; bloques de función (FB) que son bloques lógicos que depositan su valor de forma permanente en el bloque de instancia permitiendo así el uso de la función en otro bloque de instancia distinto; funciones (FC) que consisten en bloques lógicos sin memoria y bloques de datos (DB) para almacenar datos del programa.

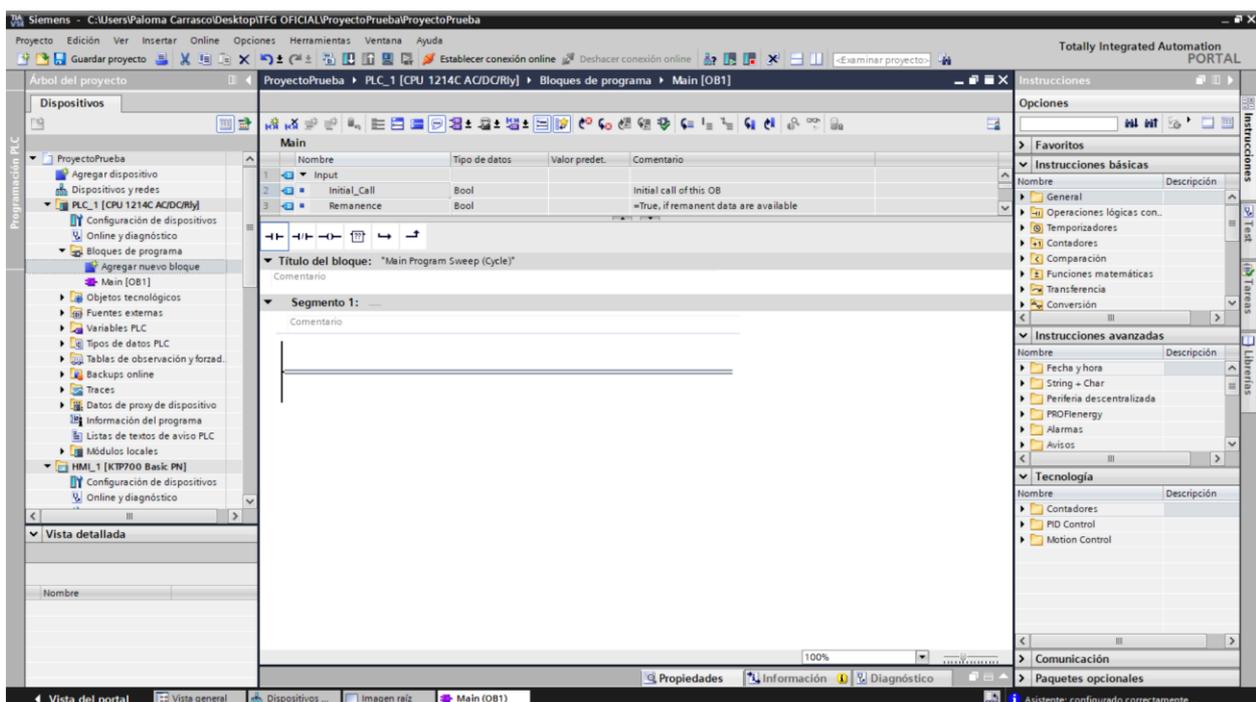


Figura 3.12 Vista del programa principal PLC (main).

3.2.1. Lenguaje PLCs

El programa principal se realiza en lenguaje KOP, lenguaje de contactos. Pero si se desea, simplemente se podría generar una función y llamarla en el main. De esta forma se puede usar FUP, basado en bloques lógicos; y SCL, lenguaje de alto nivel de SIEMENS.

Para evitar complicaciones, aunque en el proyecto se han usado funciones, se ha usado lenguaje KOP durante todo el proyecto.

3.2.2. Instrucciones empleadas PLCs

A la hora de programar, el Tia Portal tiene una ventana de favoritos que permite poner los elementos más usados, véase las figuras 3.11 y 3.12. Aunque también tiene una ventana de instrucciones, con diferentes carpetas ya sean instrucciones básicas, avanzadas, tecnológicas, de comunicación o de paquetes opcionales, como se puede observar en la figura 3.12.

A lo largo del proyecto se ha hecho uso estas instrucciones, por ello en las tablas que se muestran a continuación se han expuesto las empleadas con una breve descripción de la misma. Para más información, consultar la ayuda que ofrece Tia Portal.

	Permite introducir un nuevo segmento en el programa
	Permite introducir instrucciones, como p.e. temporizadores
	Permite crear ramas paralelas
	Permite cerrar las ramas creadas

Tabla 3.1 Instrucciones básicas - General

- -	Contacto normalmente abierto
- / -	Contacto normalmente cerrado
-()-	Asignar asignación
-(S)-	Activar salida
-(R)-	Desactivar salida

Tabla 3.2 Instrucciones básicas - Operaciones lógicas con bits

CTUD	Contador ascendente - descendente
	CTUD ???
• CU	• QU •
• CD	• QD •
• R	• CV •
• LD	
• PV	

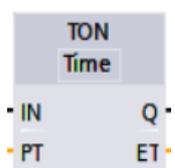
Tabla 3.3 Instrucciones básicas - Contadores.

CMP ==	Igual
CMP < >	Diferente
CMP > =	Mayor o igual
CMP < =	Menor o igual
CMP >	Mayor
CMP <	Menor

Tabla 3.4 Instrucciones básicas - Comparación.

TON	Retardo al conectar
-----	---------------------

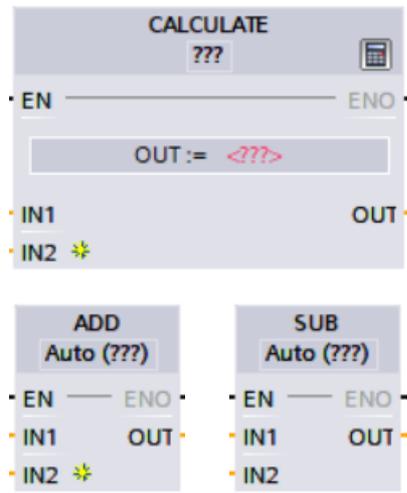
Tabla 3.5 Instrucciones básicas - Temporizadores.



The image shows the TON instruction symbol with the following details: a box labeled 'TON' with 'Time' below it. On the left side, there are two inputs: 'IN' and 'PT'. On the right side, there are two outputs: 'Q' and 'ET'.

CALCULATE	Calcular
ADD	Sumar
SUB	Restar

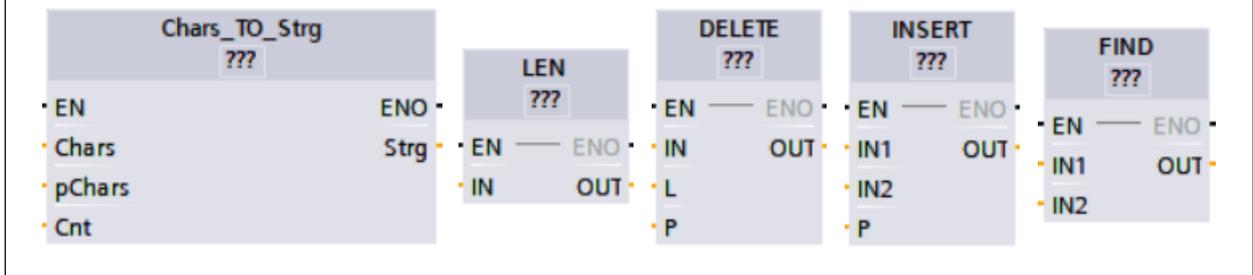
Tabla 3.6 Instrucciones básicas - Funciones matemáticas.



The image shows three instruction symbols. The top one is 'CALCULATE' with '???' below it and a calculator icon. It has an 'EN' input and an 'ENO' output, and a central box 'OUT := <???' with a red arrow. Below it are 'ADD' and 'SUB' instructions, both with 'Auto (???)' below them. 'ADD' has 'IN1' and 'IN2' inputs and 'OUT' output. 'SUB' has 'IN1' and 'IN2' inputs and 'OUT' output. Both 'ADD' and 'SUB' have 'EN' and 'ENO' inputs/outputs.

Chars_TO_Strg	Convierte un Array of Char/Byte en una cadena de caracteres a partir de un elemento de Array determinado
LEN	Determina la longitud de una cadena
DELETE	Borra caracteres de una cadena
INSERT	Introduce caracteres en una cadena
FIND	Busca caracteres en una cadena

Tabla 3.7 Instrucciones avanzadas - String + Char.



3.2.2.1. Instrucciones para la comunicación entre PLCs

En este proyecto se ha hecho uso de las indicadas en la Tabla 3-8. Este tipo de instrucciones permiten una comunicación continua sin necesidad de activación o desactivación. Concretamente, permiten el envío de mensajes con un tamaño de un byte.

TSEND_C	Enviar datos vía Ethernet
TRCV_C	Recibir datos vía Ethernet

Tabla 3.8 Comunicación – Open user

- TSEND_C.

Se trata del envío. Para poder configurarlo se debe pulsar el maletín de la figura 3.13, lo cual abre una pantalla como la de la figura 3.14.

Para poder unirlo al autómata deseado, se debe establecer una conexión entre ellos como se muestra en la figura 3.15. Cuando se tiene establecida la conexión se debe completar la configuración creando los datos de envío correspondientes y introduciendo los parámetros del bloque. Se puede observar un ejemplo en las figuras 3.16 y 3.17.

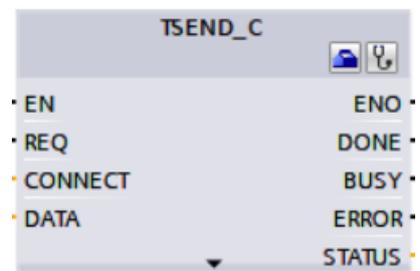


Figura 3.13 TSEND_C.

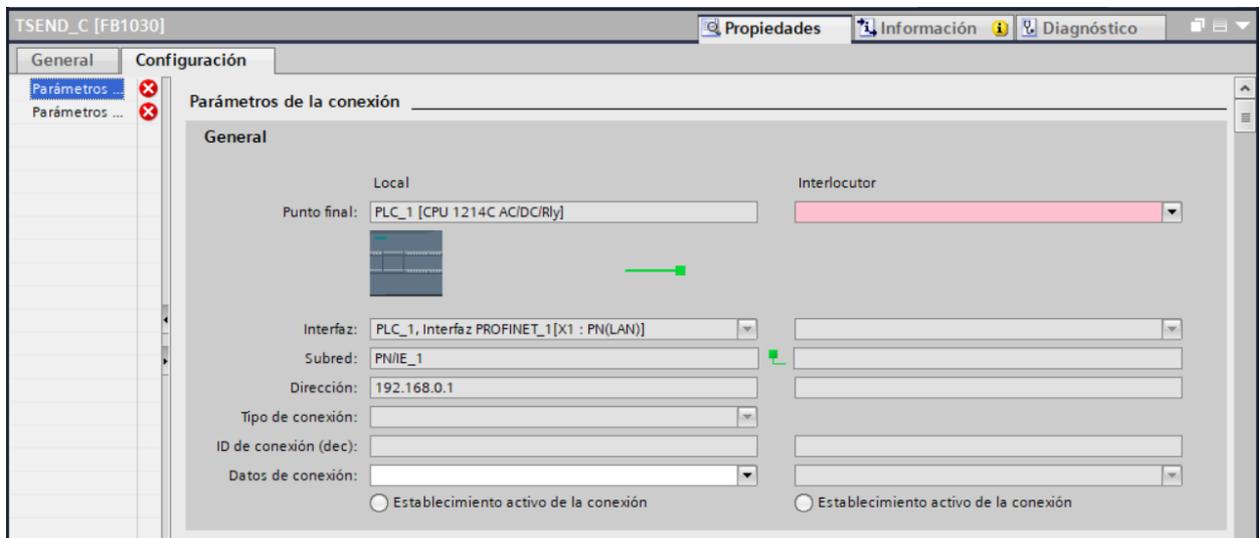


Figura 3.14 TSEND_C Configuración.

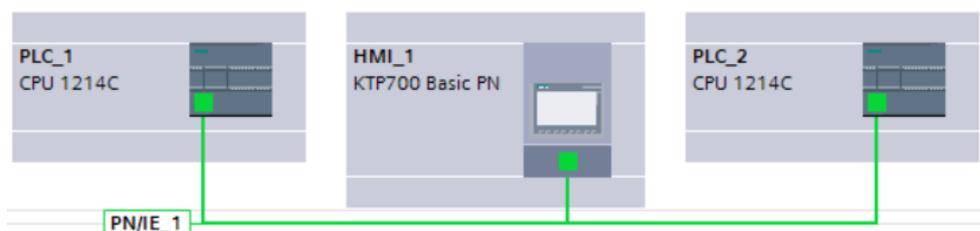


Figura 3.15 Conexión entre PLCs.

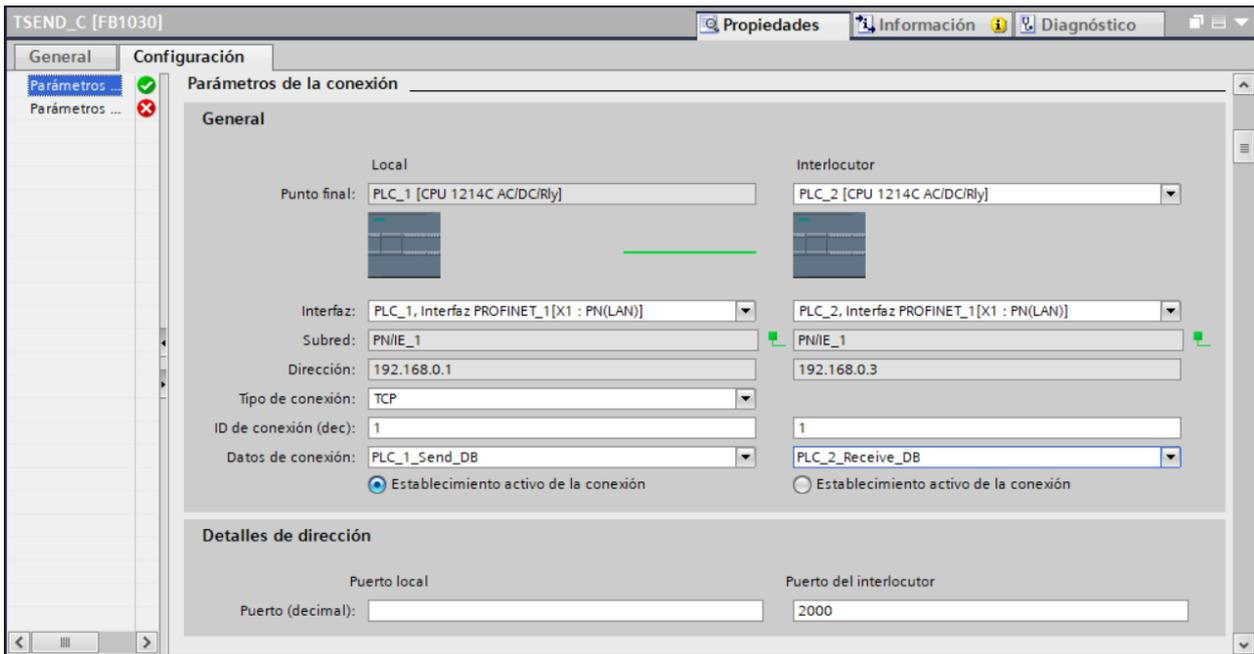


Figura 3.16 Ejemplo parámetros de conexión TSEND_C.



Figura 3.17 Ejemplo parámetros del bloque TSEND_C.

- TRCV_C.

Instrucción para la recepción de información, figura 3.18. Si ya se ha configurado anteriormente el envío, la conexión ya está hecha, en el caso contrario sería de la misma forma que la que se ha indicado anteriormente. Hecha la conexión se debe completar la configuración creando los datos de envío correspondientes y introduciendo los parámetros del bloque. Se puede observar un ejemplo en las figuras 3.19 y 3.20.

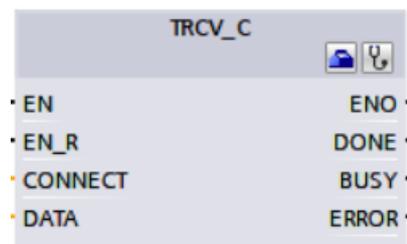


Figura 3.18 Ejemplo parámetros del bloque TSEND_C.

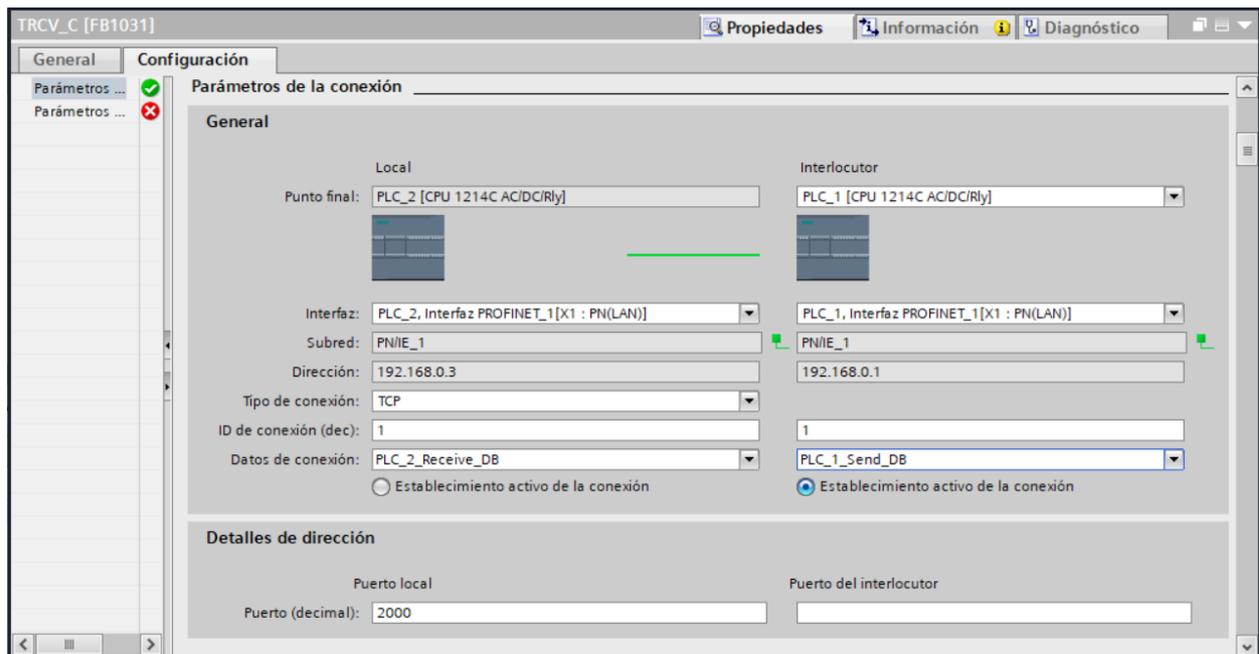


Figura 3.19 Ejemplo parámetros de conexión TRCV_C.



Figura 3.20 Ejemplo parámetros del bloque TRCV_C.

3.3. Programación HMIs

La programación de una pantalla SIMATIC HMI es totalmente distinta, dado que no se emplea un lenguaje como tal. Primeramente, es fundamental crear una plantilla al gusto, o bien usar la que ya viene creada. Esto permite que a medida que se vayan añadiendo pantallas, inicialmente todas sean iguales. Para acceder a las plantillas solamente hay que abrir la carpeta de administración de imágenes y en su interior se encuentran las plantillas.

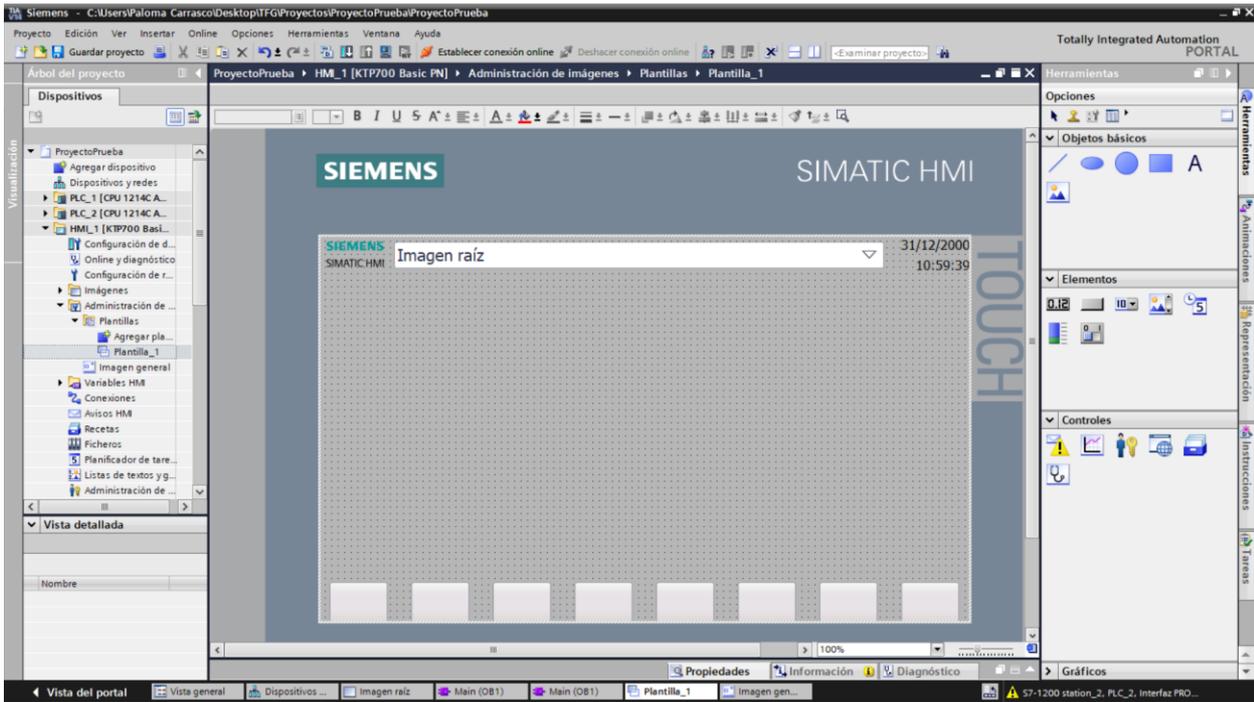


Figura 3.21 Plantilla HMI.

En la barra de herramientas situada en el lateral derecho, véase la figura 3.21, resultan interesantes: herramientas, que dispone de los objetos, elementos y controles que se pueden introducir en la pantalla; animaciones, que permiten variar los elementos u objetos; y representación, permite ver de forma esquemática lo que contiene la pantalla.

Para poder comprender la programación de las pantallas hay que tener claro la diferencia entre eventos y animaciones. Un evento es una función relacionada con un cambio, no de apariencia o visualización. En la Tabla 3.9 se muestran las posibles formas en las que puede suceder un evento de las herramientas empleadas en el proyecto. Dentro de las posibles funciones que se pueden realizar en un evento, las más usadas en este proyecto han sido: ActivarImagen, ActivarBit y DesactivarBit. Una animación se refiere a la forma física de la herramienta, es decir, su visibilidad o apariencia. Por ejemplo: usando como herramienta un botón, un evento puede ser que al pulsarlo aparezca otra pantalla mientras que una animación puede ser que el botón no aparezca hasta que una cierta variable no esté activa.

	Texto	No permite eventos
	Circulo	No permite eventos
	Rectángulo	No permite eventos
	Campo entrada-salida	Activar / Desactivar
	Botón	Hacer clic / Pulsar / Soltar / Activar / Desactivar / Cambio
	Interruptor	Cambio / Conmutar ON / Conmutar OFF / Activar / Desactivar

Tabla 3.9 Herramientas empleadas y eventos disponibles.

3.3.1. Comunicación entre HMI y PLC

Para establecer una conexión entre la pantalla y el PLC, debe existir una conexión entre los dispositivos. Si a la hora de agregar la pantalla no se estableció esta conexión, ésta debe hacerse de forma manual al igual que cuando se conectaron dos autómatas, véase la figura 3.15.

Si la conexión ya está establecida, cuando se cree una variable en la pantalla tiene que dar la opción de crear un acceso a una del autómata. Este acceso puede ser simbólico o absoluto. Si se trata de una simbólico, la pantalla no podrá modificar la variable, de forma absoluta sí puede.

Nombre	Tabla de variables	Tipo d...	Conexión	Nombre del PLC	Variable PLC	Direc...	Modo de acceso	Ciclo de adquisi..
Variable_prueba_HMI	Tabla de variables ..	Bool	HMI_Conexión_1	PLC_1	variable_prueba_PLC		<Acceso simbólico>	100 ms
Variable_prueba_HMI_1	Tabla de variables ..	Bool	HMI_Conexión_1	PLC_1	variable_prueba_PLC	%I0.0	<Acceso absoluto>	100 ms

Figura 3.22 Conexión de variables.

Como se puede observar en la figura 3.22, se elige el PLC al que se desea acceder. De forma absoluta se accede a la dirección ya que se va a modificar según lo que se haga en la pantalla con la variable, pero de forma simbólica no. Además es importante tener en cuenta el ciclo de adquisición, dado que al ser aparatos reales por muy pequeños que sean, lo mínimo es 100 ms.

Para evitar complicaciones, en el proyecto propuesto, todas las variables que resultan conectadas entre los autómatas y las pantallas están en acceso absoluto.

4 PROGRAMACIÓN DE LOS PLCs

Este capítulo trata sobre la propia programación realizada en el proyecto en los PLCs para la automatización del parking. Primeramente, se hablará de la configuración realizada en el proyecto así como de la conexión de los elementos del programa y posteriormente la programación cada uno.

4.1. Configuración y conexión de la arquitectura implementada

Como se ha mencionado en capítulos anteriores se dispone de dos PLCs, PLC_usuario y PLC_administrador, y dos pantallas HMI, HMI_usuario y HMI_administrador. Aunque como se verá en el apartado siguiente para que la comunicación sea correcta todos deben estar conectados, realmente en este proyecto están conectados dos a dos. Es decir por un lado se tiene la parte del administrador, operador del parking, y por otro lado la del usuario, cliente del parking. De forma que las pantallas únicamente transmiten información con su PLC correspondiente y son los PLCs entre ellos, los encargados de comunicar las dos partes mediante PROFINET.

Haciendo uso del capítulo 3 de esta memoria se pueden incluir las dos pantallas y los dos PLCs estableciendo una conexión entre ellos tal y como se muestra en la figura 4.1. Cabe destacar que las direcciones de cada elemento debe ser la propia del dispositivo del laboratorio, véase tabla 4.1.

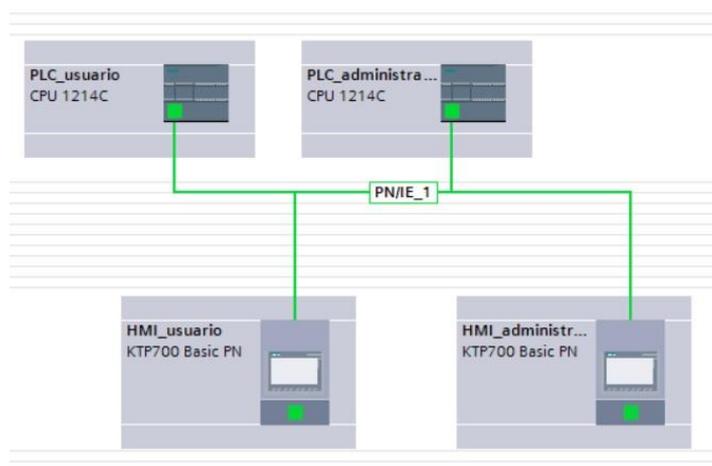


Figura 4.1 Conexión del Proyecto.

PLC_usuario	192.168.1.107
PLC_administrador	192.168.1.111
HMI_usuario	192.168.1.120
HMI_administrador	192.168.1.121

Tabla 4.1 Direcciones de los dispositivos del trabajo.

4.2. Proyecto_FINAL PLCs

4.2.1. PLC_usuario

A continuación, se va a realizar la descripción paso a paso de lo que realiza este autómata.

- Segmento 1. Elección de operación a realizar.

El usuario decide si aparcar el coche o sacar el coche (si ya lo tenía aparcado) y se inicializan los índices para la base de datos, así como se resetean las banderas de envío, de enviar el aparcar y de enviar el sacar. El programa propuesto es el que se muestra en la figura 4.2. En la Tabla 4.2 se muestran las variables usadas para este segmento.

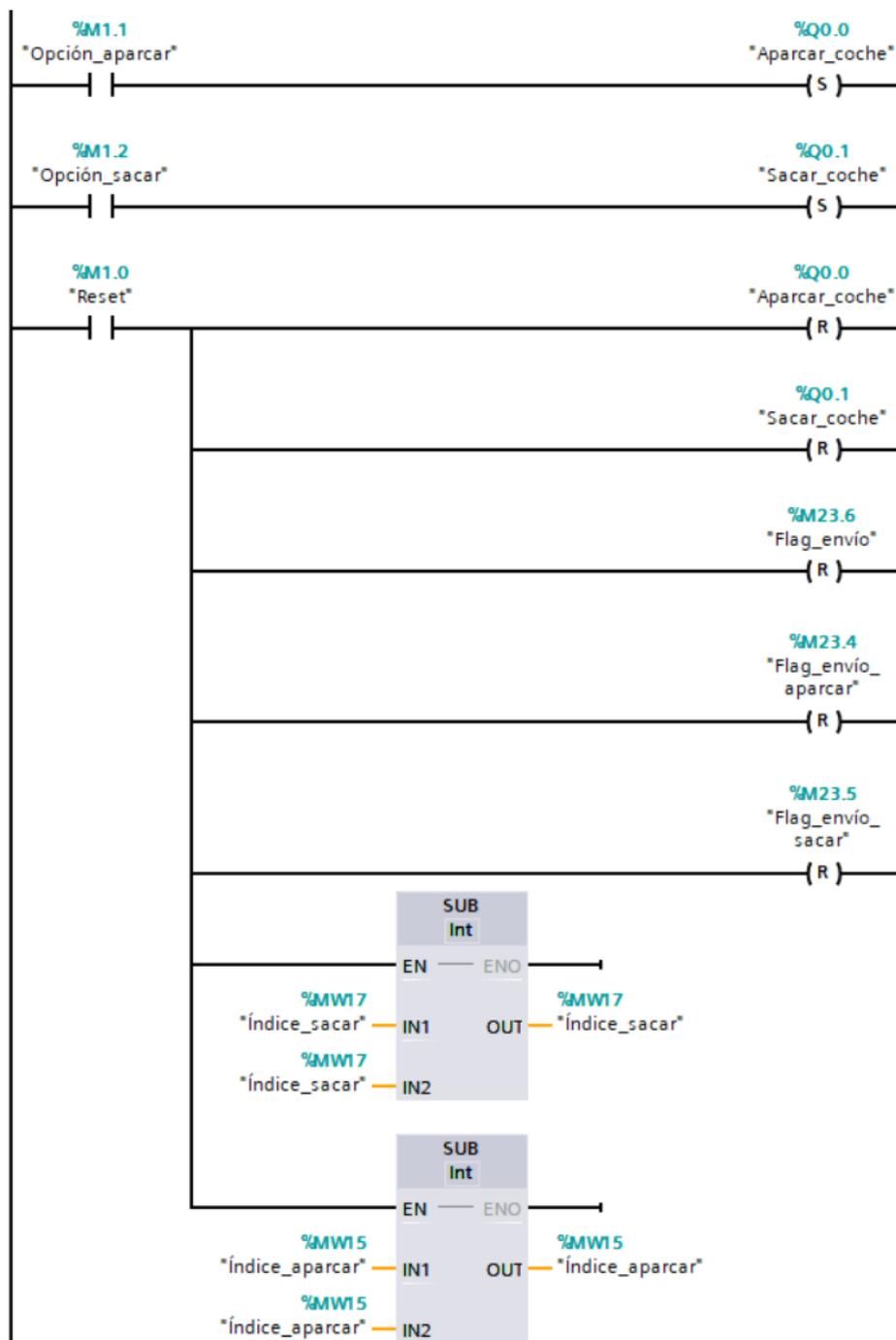


Figura 4.2 Programación Segmento-1 PLC_usuario.

Nombre	Tipo de datos	Dirección	Comentario
Aparcar_coche	Bool	%Q0.0	Elección de aparcar coche
Sacar_coche	Bool	%Q0.1	Elección de sacar coche
Reset	Bool	%M1.0	Reseteo de elección
Opción_aparcar	Bool	%M1.1	Flag elección aparcar
Opción_sacar	Bool	%M1.2	Flag elección sacar
Flag_envío_aparcar	Bool	%M23.4	Flag para activar el envío aparcar
Flag_envío_sacar	Bool	%M23.5	Flag para activar el envío sacando
Flag_envío	Bool	%M23.6	Flag para activar el envío
Índice_aparcar	Int	%MW15	Índice de base de datos para coches que se aparcan
Índice_sacar	Int	%MW17	Índice de base de datos para coches que se sacan

Tabla 4.2 Variables Segmento-1 PLC_usuario.

- Segmento 2. Introducir la matrícula.

El usuario introduce su matrícula y se almacena en un string para posteriormente mostrarlo en la pantalla, se guarda la longitud de la matrícula y la longitud de la matrícula más uno para posteriores operaciones. El programa propuesto es el que se muestra en la figura 4.3. En la Tabla 4.3 se muestran las variables usadas para este segmento.

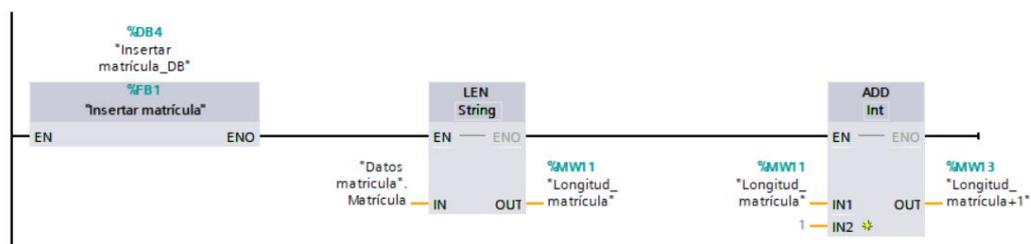


Figura 4.3 Programación Segmento-2 PLC_usuario.

Para poder introducir la matrícula y almacenarla se ha creado una función llamada “Insertar matrícula”. Se crea un bloque de datos que contiene las letras del abecedario y los números. Y se crean variables que simbolizan cada una de las letras y números, estableciendo una conexión entre el bloque de datos y las variables para que cuando se pulse los botones de la pantalla, se introduzca el char/int del bloque de datos en el string bandera que simboliza la matrícula. Para que no se solapen unos caracteres a otros existe un contador que aumenta cada vez que se pulsa una tecla, que disminuye cuando se pulsa borrar en la pantalla y se resetea cada vez que en el menú se marca una opción. Esto hace que cada vez que llegue un nuevo usuario, el contador de la posición de la matrícula esté en 0. Una vez que se pulsa una tecla, se inserta el string bandera en el string de la matrícula real en la posición indicada por el contador. Como la función Insert copia en todo el string, se tiene que borrar a partir de la posición que marque el contador. Por ejemplo: si el usuario pulsase una sola A, se introduciría A en el string bandera y AAA... en el string real de la matrícula. Posteriormente se borra a partir de la posición 2 dado que sólo se ha introducido una tecla A, resultando la matrícula real como A. Si ahora pulsa una B el contador marca un 2. El string bandera sería una B y el string real de la matrícula ABB... por lo que al borrar a partir de la posición 3 se obtiene AB. En la figura 4.4 se muestra el programa propuesto para la función.

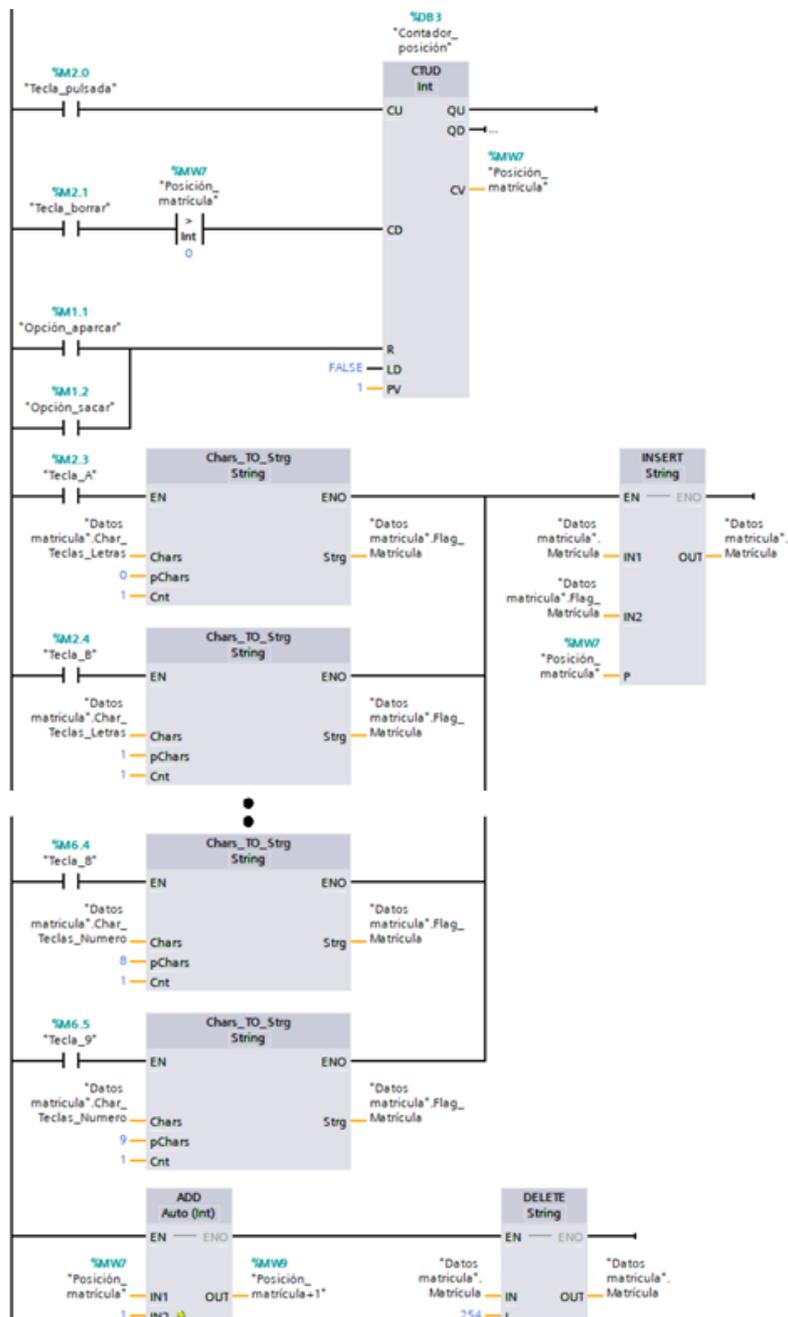


Figura 4.4 Programación Segmento-2 “Insertar matrícula” PLC_usuario.

Nombre	Tipo de datos	Dirección	Comentario
Opción_aparcar	Bool	%M1.1	Flag elección aparcar
Opción_sacar	Bool	%M1.2	Flag elección sacar
Tecla_pulsada	Bool	%M2.0	Pulsado un botón del teclado
Tecla_borrar	Bool	%M2.1	Pulsado el borrar del teclado
Tecla_A	Bool	%M2.3	Botón del teclado
Tecla_B	Bool	%M2.4	Botón del teclado
	...		
Tecla_9	Bool	%M6.5	Botón del teclado
Tecla_0	Bool	%M6.6	Botón del teclado
Posición_matrícula	Int	%MW7	Posición actual de la matrícula
Posición_matrícula+1	Int	%MW9	Posición para borrar el string restante
Longitud_matrícula	Int	%MW11	Longitud de la matrícula introducida por el usuario
Longitud_matrícula+1	Int	%MW13	Longitud de la matrícula +1 para borrar el string restante

Tabla 4.3 Variables Segmento-2 PLC_usuario.

Para poder tener una visión completa, se adjunta en la figura 4.5 una captura del bloque de datos relacionados con la matrícula. En ella se puede ver que existen los dos string, uno que hace de bandera guardando la última tecla pulsada y el real que guarda la matrícula completa. Además están todos los caracteres del abecedario y los números.

Datos matricula (instantánea generada: 20/07/2018 22:43:12)									
	Nombre	Tipo de datos	Valor de arranq...	Remanen...	Accesible d...	Escrib...	Visible en ..	Valor de a...	
1	Static								
2	Matricula	String	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
3	Flag_Matricula	String	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
4	Char_Teclas_Letras	Array[0..25] of Char		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
5	Char_Teclas_Letras..	Char	'A'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
6	Char_Teclas_Letras..	Char	'B'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
7	Char_Teclas_Letras..	Char	'C'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
8	Char_Teclas_Letras..	Char	'D'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
9	Char_Teclas_Letras..	Char	'E'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
10	Char_Teclas_Letras..	Char	'F'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
11	Char_Teclas_Letras..	Char	'G'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
12	Char_Teclas_Letras..	Char	'H'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
13	Char_Teclas_Letras..	Char	'I'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
14	Char_Teclas_Letras..	Char	'J'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
15	Char_Teclas_Letras..	Char	'K'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
16	Char_Teclas_Letras..	Char	'L'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
17	Char_Teclas_Letras..	Char	'M'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
18	Char_Teclas_Letras..	Char	'N'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
19	Char_Teclas_Letras..	Char	'O'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
20	Char_Teclas_Letras..	Char	'P'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
21	Char_Teclas_Letras..	Char	'Q'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
22	Char_Teclas_Letras..	Char	'R'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
23	Char_Teclas_Letras..	Char	'S'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
24	Char_Teclas_Letras..	Char	'T'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
25	Char_Teclas_Letras..	Char	'U'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
26	Char_Teclas_Letras..	Char	'V'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
27	Char_Teclas_Letras..	Char	'W'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
28	Char_Teclas_Letras..	Char	'X'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
29	Char_Teclas_Letras..	Char	'Y'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
30	Char_Teclas_Letras..	Char	'Z'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
31	Char_Teclas_Numero	Array[0..9] of Char		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
32	Char_Teclas_Num...	Char	'0'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
33	Char_Teclas_Num...	Char	'1'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
34	Char_Teclas_Num...	Char	'2'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
35	Char_Teclas_Num...	Char	'3'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
36	Char_Teclas_Num...	Char	'4'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
37	Char_Teclas_Num...	Char	'5'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
38	Char_Teclas_Num...	Char	'6'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
39	Char_Teclas_Num...	Char	'7'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
40	Char_Teclas_Num...	Char	'8'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>
41	Char_Teclas_Num...	Char	'9'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input type="checkbox"/>

Figura 4.5 Bloque de datos relacionados con la matrícula.

- Segmento 3. Base de datos.

Este segmento sirve para escribir y leer de la base de datos del parking, para ver las plazas libres o ocupadas y así asignar correctamente la plaza. En el caso de aparcar, se trata de aparcar en la plaza mas cercana posible. Teniendo en cuenta que el parking tiene en cada planta 6 aparcamientos, véase la figura 4.6, la asignación sería de 1-6. Siendo la más cercana el 1 y la más lejana el 6.

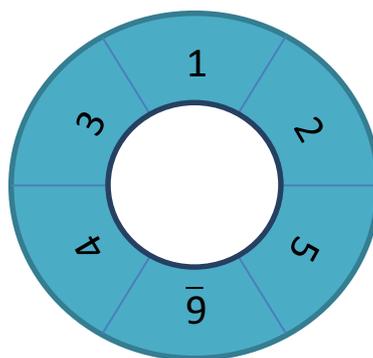


Figura 4.6 Aparcamientos por planta

Base de datos (instantánea generada: 20/07/2018 22:43:12)									
	Nombre	Tipo de datos	Valor de arranq...	Remanen...	Accesible d...	Escrib...	Visible en ..	Valor de a..	
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	Base	Array[0..11] of Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	Base[0]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	Planta_Plaza	Int	11	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	Base[1]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	Planta_Plaza	Int	12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
11	Base[2]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
12	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	Planta_Plaza	Int	13	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
15	Base[3]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
16	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
17	Planta_Plaza	Int	14	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
18	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
19	Base[4]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
20	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
21	Planta_Plaza	Int	15	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
22	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
23	Base[5]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
24	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
25	Planta_Plaza	Int	16	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
26	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
27	Base[6]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
28	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
29	Planta_Plaza	Int	21	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
30	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
31	Base[7]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
32	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
33	Planta_Plaza	Int	22	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
34	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
35	Base[8]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
36	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
37	Planta_Plaza	Int	23	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
38	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
39	Base[9]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
40	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
41	Planta_Plaza	Int	24	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
42	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
43	Base[10]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
44	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
45	Planta_Plaza	Int	25	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
46	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
47	Base[11]	Struct		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
48	Matrícula_usua...	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
49	Planta_Plaza	Int	26	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
50	Ocupación	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Figura 4.7 Bloque de datos de la base de datos del parking.

En la figura 4.7 se puede observar el bloque de datos que se ha realizado para crear la base de datos. Se crea una estructura que contiene la matrícula, la plaza por orden de cercanía y la ocupación. Si la ocupación está en false significa que está vacía y en el caso contrario ocupada. Como se puede observar, solamente hay disponibles dos plantas. En el siguiente segmento se explicará cuantas plazas se pueden tener como máximo. Para poder ampliar las plazas simplemente hay que ampliar el bloque de datos y modificar un poco el código en el PLC_administrador, lo cual será explicado cuando se explique su programa.

Se ha usado una función para ponerla en el programa principal y así simplificar la visualización global. El programa propuesto para manipular la base de datos es el que se muestra en la figura 4.8. En la Tabla 4.4 se muestran las variables usadas para este segmento.

Nombre	Tipo de datos	Dirección	Comentario
Aparcar_coche	Bool	%Q0.0	Elección de aparcar coche
Sacar_coche	Bool	%Q0.1	Elección de sacar coche
Matrícula_correcta	Bool	%M2.2	Afirmación de matrícula
Longitud_matrícula+1	Int	%MW13	Longitud de la matrícula +1 para borrar el string restante
Índice_aparcar	Int	%MW15	Índice de base de datos para coches que se aparcan
Índice_sacar	Int	%MW17	Índice de base de datos para coches que se sacan
Encontrar_matrícula	Int	%MW19	Índice para moverse por la base de datos y encontrar la matrícula introducida para sacar el coche
Contador_coches	Int	%MW21	Número de coches que se encuentran en el parking
Decrementar_contador_coches	Bool	%M23.0	Flag para decrementar el número de coches que se encuentran en el parking
Parking_completo	Bool	%M23.1	Indicador de que no hay plazas disponibles para aparcar
Matrícula_noencontrada	Bool	%M23.2	Indicador de que no se encuentra la matrícula en la base de datos
Flag_envío_aparcar	Bool	%M23.4	Flag para activar el envío aparcando
Flag_envío_sacar	Bool	%M23.5	Flag para activar el envío sacando
Flag_envío	Bool	%M23.6	Flag para activar el envío
Índice_aparcar_correcto	Int	%MW26	Índice de aparcar
Índice_sacar_correcto	Int	%MW28	Índice de sacar
Longitud_Base_Datos	Int	%MW31	Longitud de la matrícula de la base de datos

Tabla 4.4 Variables Segmento-2 PLC_usuario

- Segmento 4. Envío de datos al administrador.

Este segmento sirve para codificar la información y hacer en un único envío. De esta forma se indica la acción a realizar y la plaza en la que se tiene que hacer. El envío se encuentra limitado por un byte. Esto quiere decir que el máximo número que se puede enviar es un 255.

Dado que en la base de datos se numeró cada posición como posición y planta, es decir, la planta 1 y posición 1 es la 11. Y que la planta 1 tendría de 11-16, la planta 2 de 21-26 y así sucesivamente en el caso de tener más de dos plantas. La codificación llevada a cabo trata de dividir el byte en dos y usar cada parte para cada uno de los procesos.

Para poder indicar que lo que quiere el usuario es aparcar, se coge el número de la planta y posición y se le resta 10. De esta forma, si al administrador le llega un número del 1-6 sabrá que es la planta 1 y la posición que indique dicho número. Si le llega un 11-16 sabrá que es la planta 2 y así sucesivamente. Y para poder indicar que se saca el coche, se le suma 120. En la tabla 4.5 se pueden encontrar las posibles codificaciones de la planta según se aparque o saque el coche. Como se podrá observar el número máximo de plantas que se pueden tener de esta forma es 11 teniendo 66 aparcamientos, dado que no se puede enviar un 256 que haría que fueran posible 12 plantas.

Numero enviado	Elección	Posición real	Numero enviado	Elección	Posición real
1 - 6	APARCAR	Planta 1	131 - 136	SACAR	Planta 1
11 - 16	APARCAR	Planta 2	141 - 146	SACAR	Planta 2
21 - 26	APARCAR	Planta 3	151 - 156	SACAR	Planta 3
31 - 36	APARCAR	Planta 4	161 - 166	SACAR	Planta 4
41 - 46	APARCAR	Planta 5	171 - 176	SACAR	Planta 5
51 - 56	APARCAR	Planta 6	181 - 186	SACAR	Planta 6
61 - 66	APARCAR	Planta 7	191 - 196	SACAR	Planta 7
71 - 76	APARCAR	Planta 8	201 - 206	SACAR	Planta 8
81 - 86	APARCAR	Planta 9	211 - 216	SACAR	Planta 9
91 - 96	APARCAR	Planta 10	221 - 226	SACAR	Planta 10
101 - 106	APARCAR	Planta 11	231 - 236	SACAR	Planta 11
110 - 116	APARCAR	Planta 12	241 - 246	SACAR	Planta 12
121 - 126	APARCAR	Planta 13	251 - 256	SACAR	Planta 13

Tabla 4.5 Tabla de codificación del envío.

La programación de esta parte se puede encontrar en la figura 4.10 así como las variables usadas en a tabla 4.6. Lo único que cabe destacar es el uso de un word para la plaza y posición y así poder manipular en forma de int y luego transmitir en forma de byte. Se van a adjuntar las imágenes correspondientes con el TSEND_C en las figuras 4.11 y 4.12.

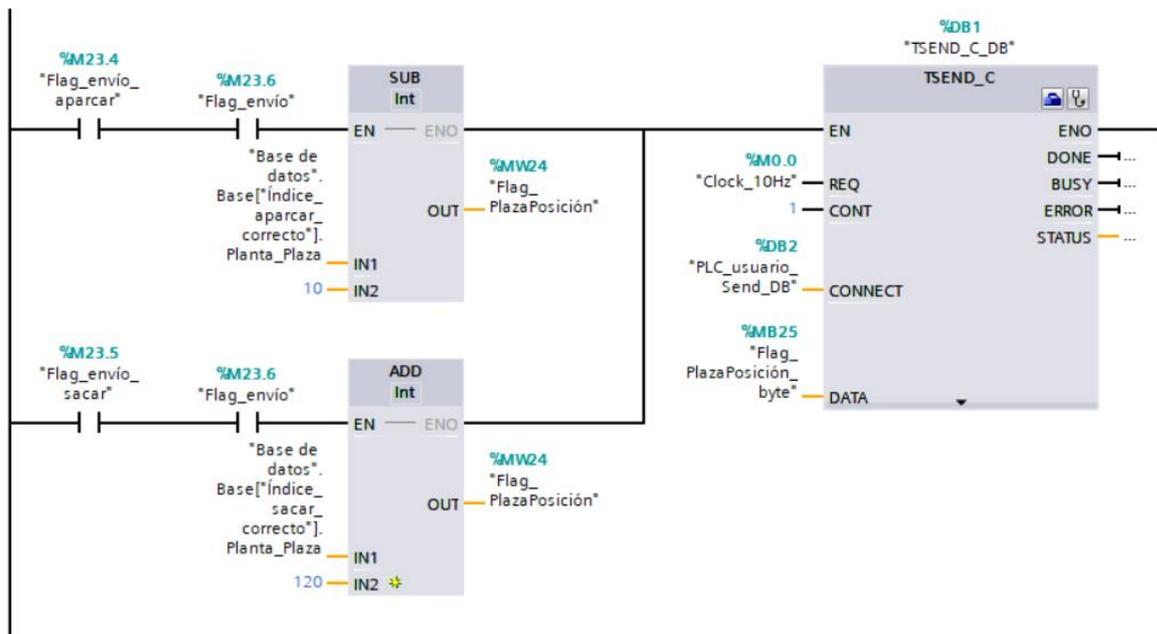


Figura 4.10 Programación Segmento-4 PLC_usuario.

Nombre	Tipo de datos	Dirección	Comentario
Flag_envío_aparcar	Bool	%M23.4	Flag para activar el envío aparcando
Flag_envío_sacar	Bool	%M23.5	Flag para activar el envío sacando
Flag_envío	Bool	%M23.6	Flag para activar el envío
Flag_PlazaPosición	Int	%MW24	Posición y plaza en word
Flag_PlazaPosición_byte	Byte	%MB25	Posición y plaza en byte
Clock_10Hz	Bool	%M0.0	Reloj

Tabla 4.6 Variables Segmento-4 PLC_usuario.

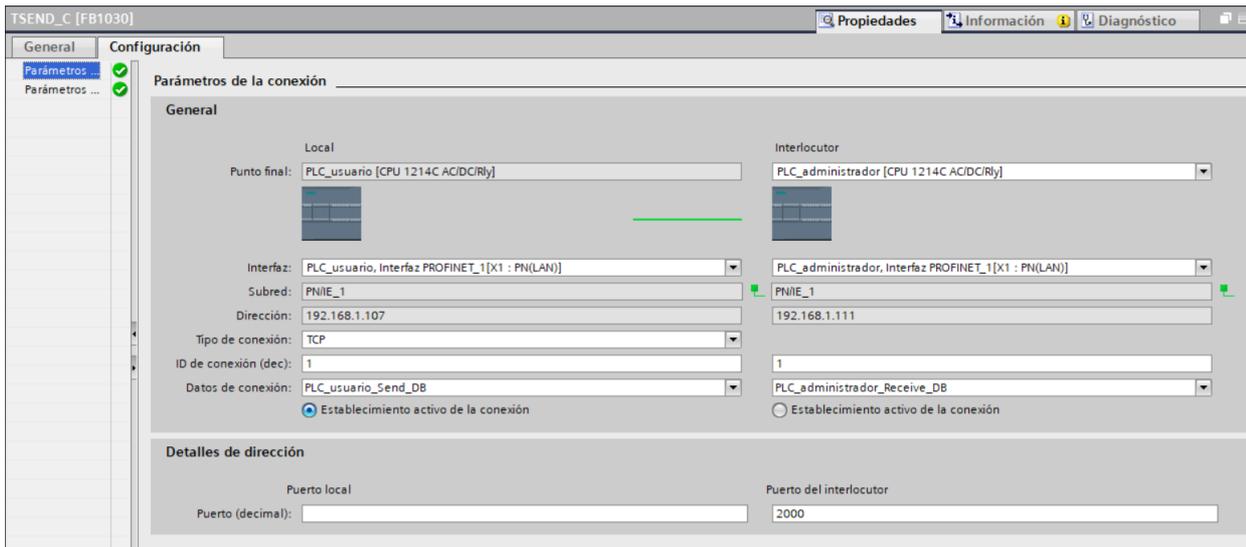


Figura 4.11 TSEND_C Parámetros de la conexión.

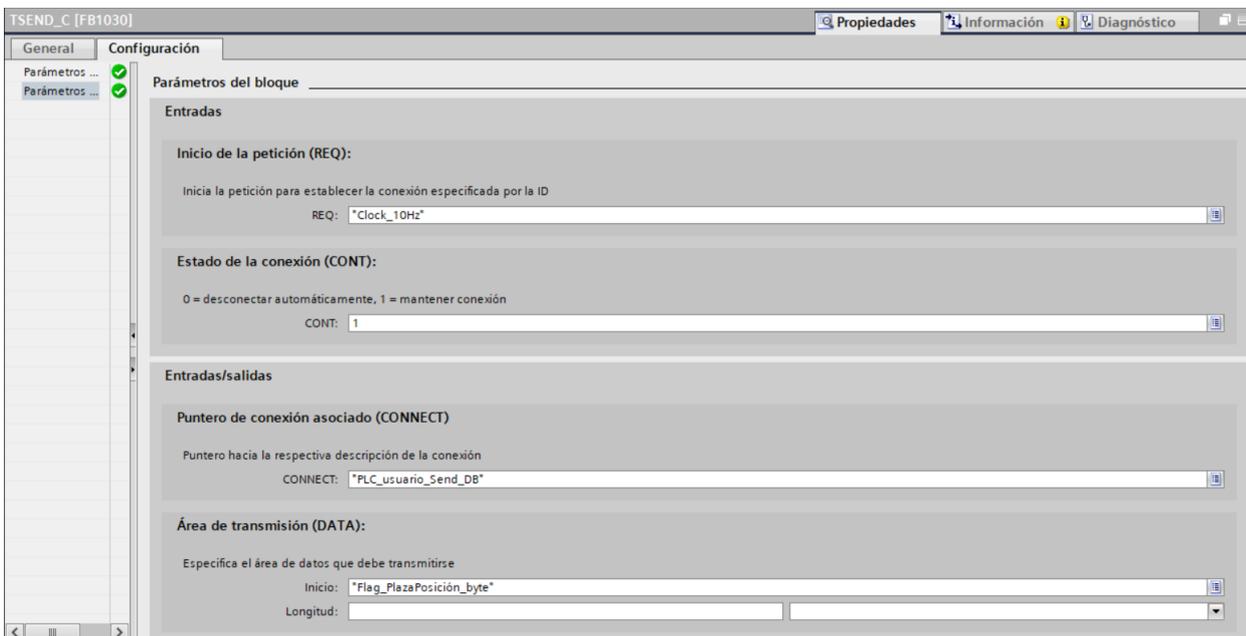


Figura 4.12 TSEND_C Parámetros del bloque.

- Segmento 5. Recibe información del administrador.

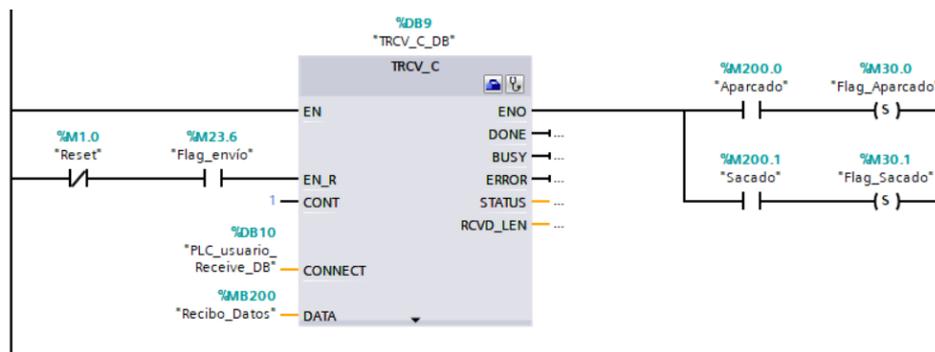


Figura 4.13 Programación Segmento-5 PLC_usuario.

Este segmento sirve para ver cuando se ha aparcado completamente el coche o cuando se ha sacado. Cabe destacar que solo se permite que se reciba información en el momento en el que el usuario ha enviado su información. Dado que en el caso de no tenerlo en cuenta se producen activaciones en bobinas que no son deseadas. Se van a adjuntar las imágenes correspondientes con el TRCV_C en las figuras 4.14 y 4.15. La programación de esta parte se puede encontrar en la figura 4.14 así como las variables usadas en a Tabla 4.7.

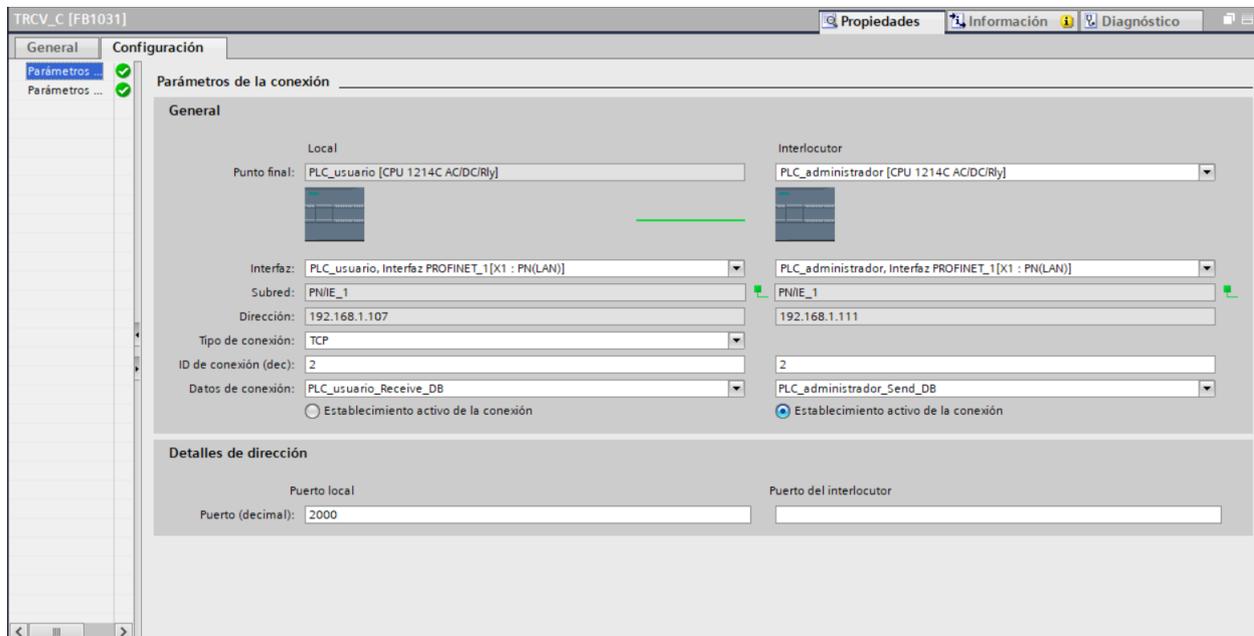


Figura 4.14 TRCV_C Parámetros de la conexión.

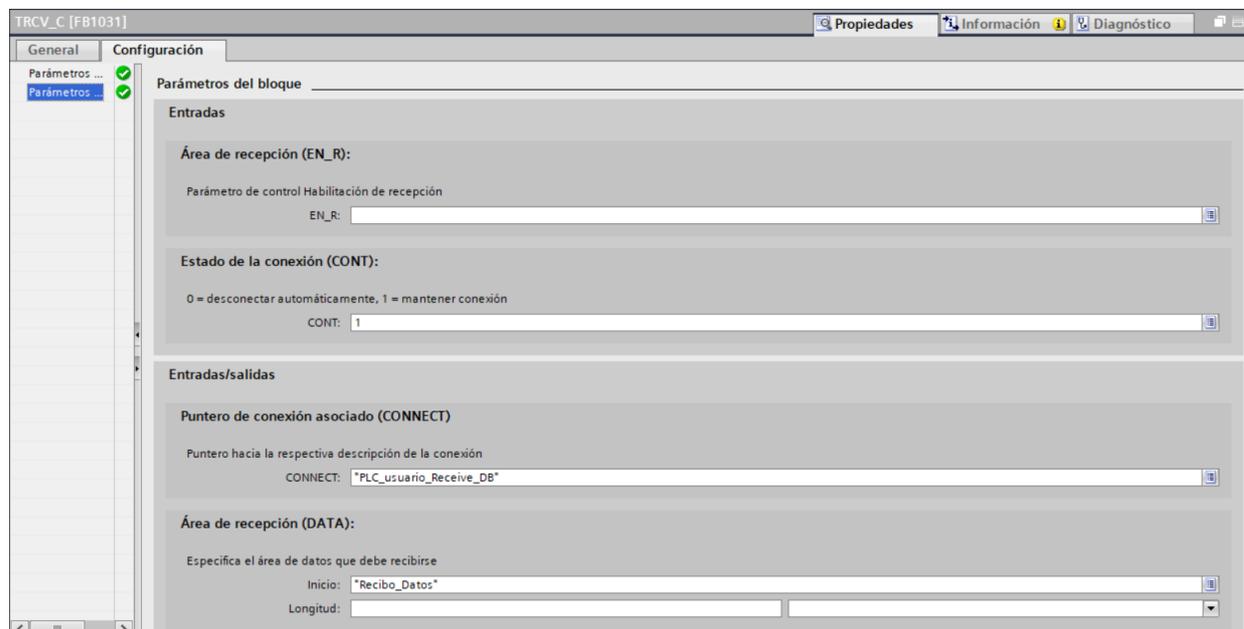


Figura 4.15 TRCV_C Parámetros del bloque.

Nombre	Tipo de datos	Dirección	Comentario
Flag_envío	Bool	%M23.6	Flag para activar el envío
Reset	Bool	%M1.0	Reseteo de la elección
Flag_Aparcado	Bool	%M30.0	Bandera que indica que se ha aparcado el vehículo
Flag_Sacado	Bool	%M30.1	Bandera que indica que se ha sacado el vehículo
Recibo_Datos	Byte	%MB200	El administrador indica lo que ocurre
Aparcado	Bool	%M200.0	El administrador indica que ya se ha aparcado
Sacado	Bool	%M200.1	El administrador indica que ya se ha sacado

Tabla 4.7 Variables Segmento-5 PLC_usuario.

4.2.2. PLC_administrador

A continuación, se va a realizar la descripción paso a paso de lo que realiza este autómata.

- Segmento 1. Recibo de información y descodificación.

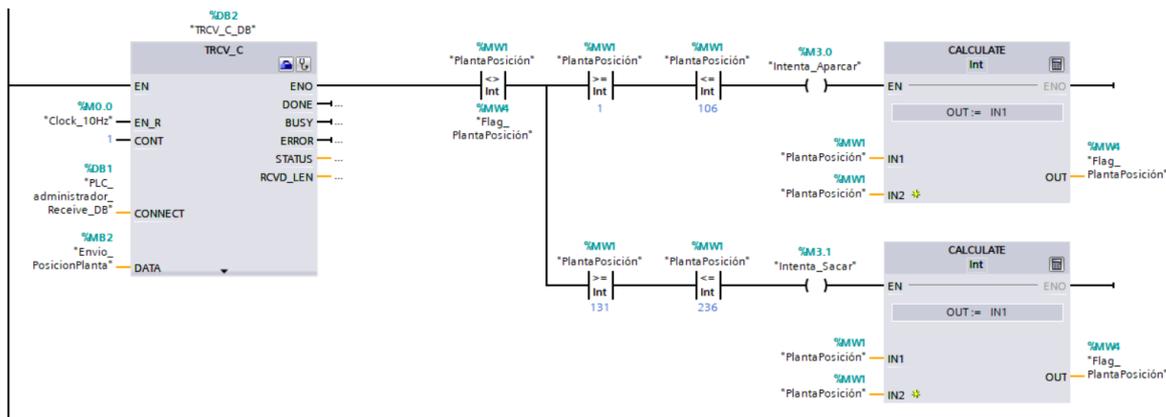


Figura 4.16 Programación Segmento-1 PLC_administrador.

Este segmento hace la descodificación de la planta y posición del coche, así como la acción a realizar. Se tiene en cuenta no realizar dos veces el mismo proceso en la misma planta, por eso se usan banderas.

Se van a adjuntar las imágenes correspondientes con el TRCV_C en las figuras 4.17 y 4.18. En lo referente a la programación, véase figura 4.16, cabe destacar el volver a usar word e int en lo mismo por manejar operaciones con números. Las variables se encuentran en la Tabla 4.8.

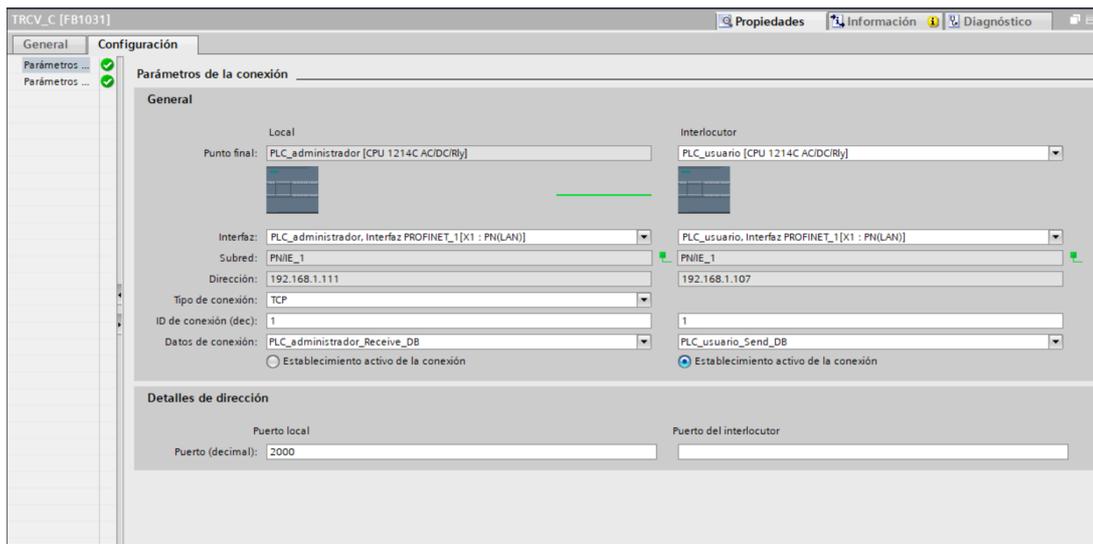


Figura 4.17 TRCV_C Parámetros de la conexión, administrador.

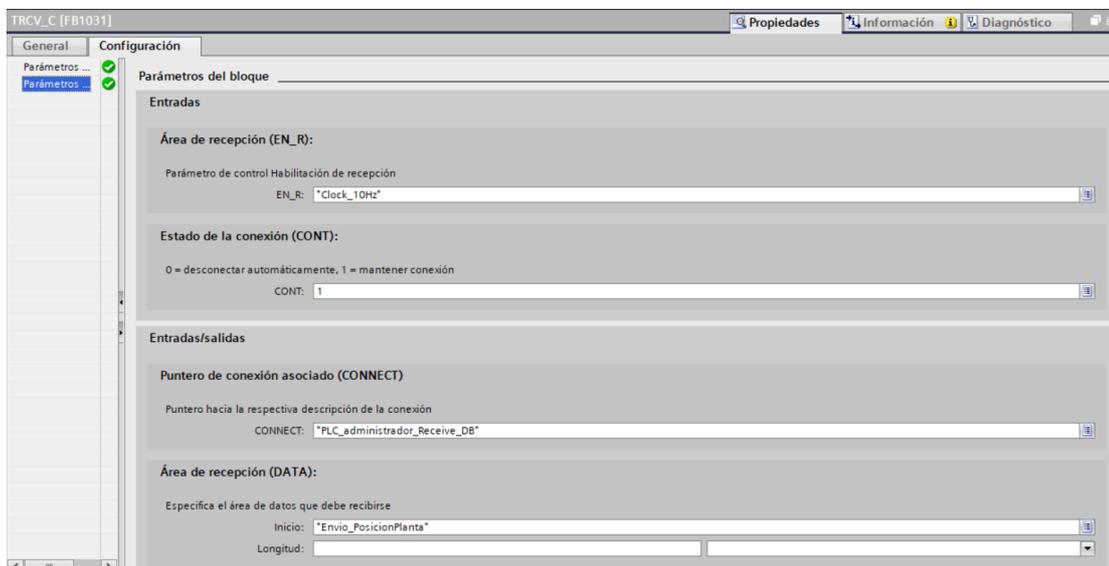


Figura 4.18 TRCV_C Parámetros del bloque, administrador.

Nombre	Tipo de datos	Dirección	Comentario
Clock_10Hz	Bool	%M0.0	Reloj
PlantaPosición	Int	%MW1	Número de planta y posición
Envio_PosicionPlanta	Byte	%MB2	Byte que recibe de planta y posición
Intenta_Aparcar	Bool	%M3.0	Acción aparcar
Intenta_Sacar	Bool	%M3.1	Acción sacar
Flag_PlantaPosición	Int	%MW4	Bandera de número de planta y posición

Tabla 4.8 Variables Segmento-1 PLC_administrador.

- Segmento 2 y 3. Proceso de aparcar y sacar.

Aunque son procesos distintos, tienen la misma forma dado que se basan en “sensores”. Se ha simulado el uso de sensores de forma que existen 3, el de seguridad, el de posición correcta y el de inicio. El de seguridad fijaría el coche para que no se moviera. El de posición correcta indica que la grúa ya ha colocado el coche en el sitio correcto, y el de inicio, que la plataforma se encuentra en la posición inicial. Se ha desarrollado de forma que si estos sensores funcionan correctamente se encontraran activos y en falso en el caso de que den fallo. De forma que se ha hecho uso de temporizadores para poder ver correctamente el proceso. En la figura 4.19 se puede encontrar el proceso de aparcar y en la 4.20 el de sacar. En la Tabla 4.9 y 4.10 las variables respectivamente.

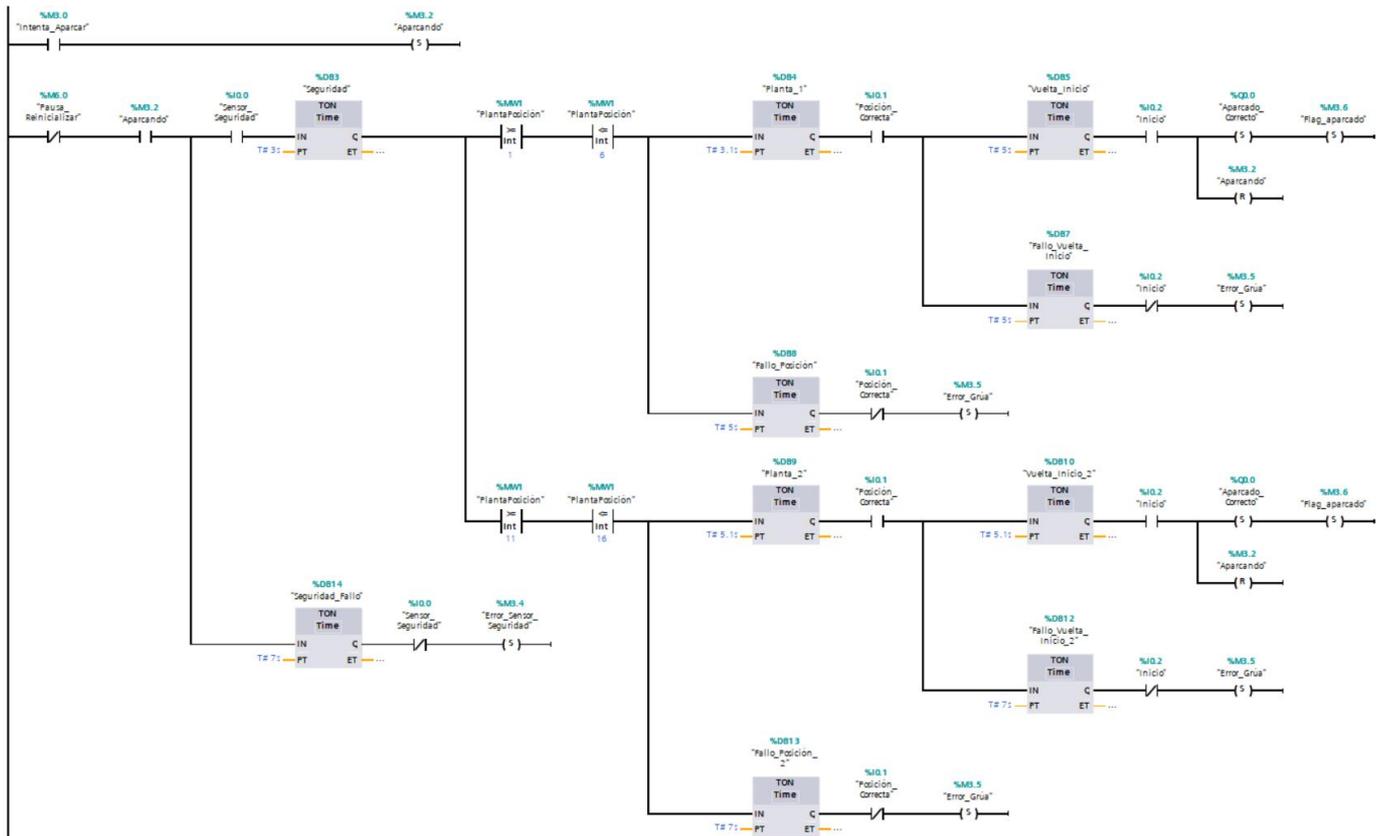


Figura 4.19 Programación Segmento-2 PLC_administrador.

Nombre	Tipo de datos	Dirección	Comentario
Sensor_Seguridad	Bool	%I0.0	Sensor
Posición_Correcta	Bool	%I0.1	Sensor
Inicio	Bool	%I0.2	Sensor
PlantaPosición	Int	%MW1	Número de planta y posición
Intenta_Aparcar	Bool	%M3.0	Bandera aparcando
Aparcando	Bool	%M3.2	Aparcando
Error_Sensor_Seguridad	Bool	%M3.4	Error sensor
Error_Grua	Bool	%M3.5	Error sensor

Flag_aparcado	Bool	%M3.6	Bandera aparcado
Aparcado_Correcto	Bool	%Q0.0	Aparcado

Tabla 4.9 Variables Segmento-2 PLC_administrador.

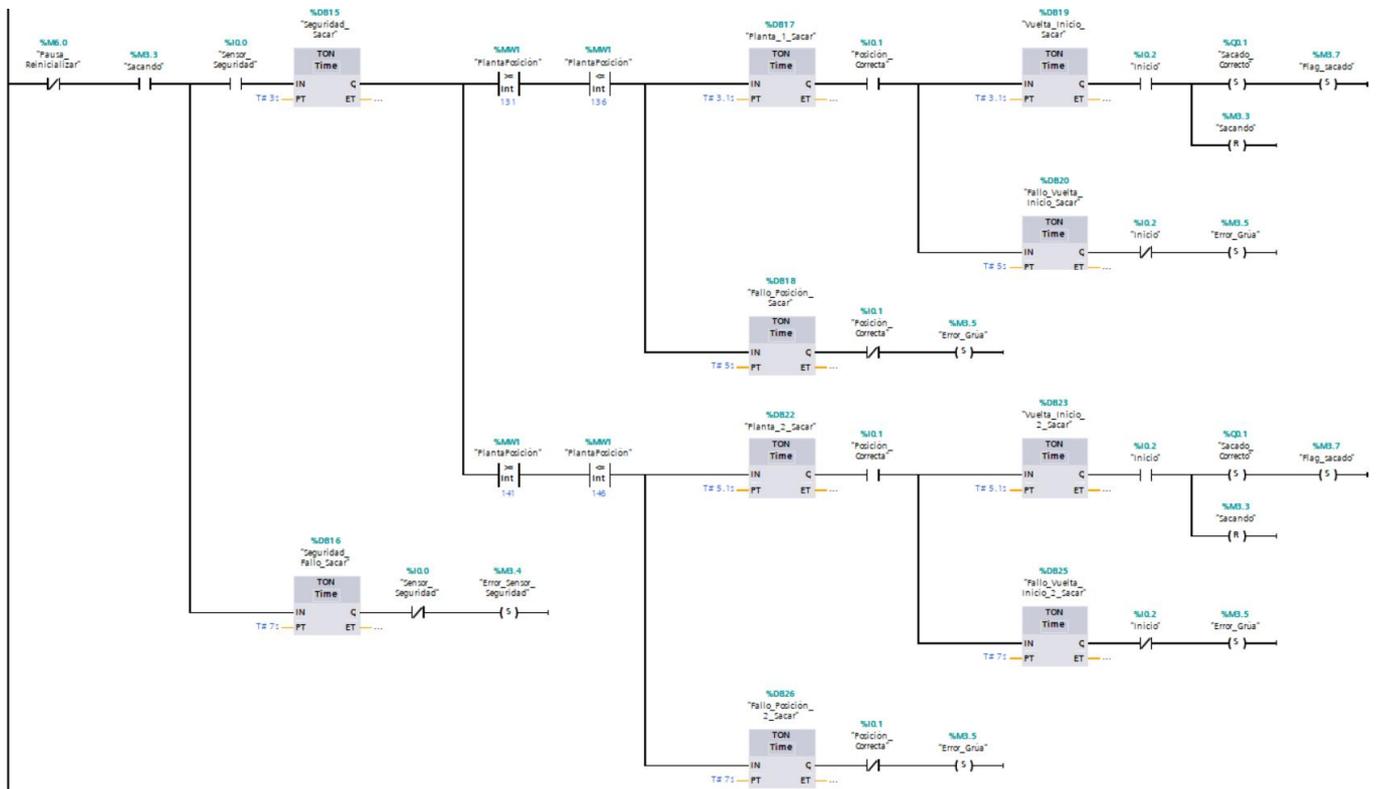


Figura 4.20 Programación Segmento-3 PLC_administrador.

Nombre	Tipo de datos	Dirección	Comentario
Sensor_Seguridad	Bool	%I0.0	Sensor
Posición_Correcta	Bool	%I0.1	Sensor
Inicio	Bool	%I0.2	Sensor
PlantaPosición	Int	%MW1	Número de planta y posición
Intenta_Sacar	Bool	%M3.1	Bandera sacando
Sacando	Bool	%M3.3	Sacando
Error_Sensor_Seguridad	Bool	%M3.4	Error sensor
Error_Grua	Bool	%M3.5	Error sensor
Flag_sacado	Bool	%M3.7	Bandera sacado
Sacado_Correcto	Bool	%Q0.1	Sacado

Tabla 4.10 Variables Segmento-3 PLC_administrador.

- Segmento 4. Corrección de errores.

Este segmento está hecho para que hasta que no vuelvan a funcionar correctamente los sensores, no deje de estar en set la bobina. Cuando se solucionan los errores se resetea y todo vuelve a la normalidad.

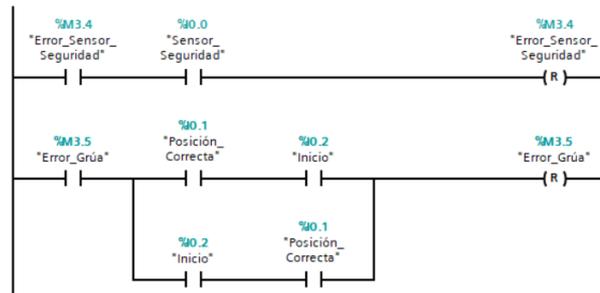


Figura 4.21 Programación Segmento-4 PLC_administrador.

Nombre	Tipo de datos	Dirección	Comentario
Sensor_Seguridad	Bool	%I0.0	Sensor
Posición_Correcta	Bool	%I0.1	Sensor
Inicio	Bool	%I0.2	Sensor
Error_Sensor_Seguridad	Bool	%M3.4	Error sensor
Error_Grúa	Bool	%M3.5	Error sensor

Tabla 4.11 Variables Segmento-4 PLC_administrador.

- Segmento 5. Envío de información al cliente.

Este segmento es el encargado de enviarle al usuario si está completo ya el proceso. Y resetear posteriormente los valores.

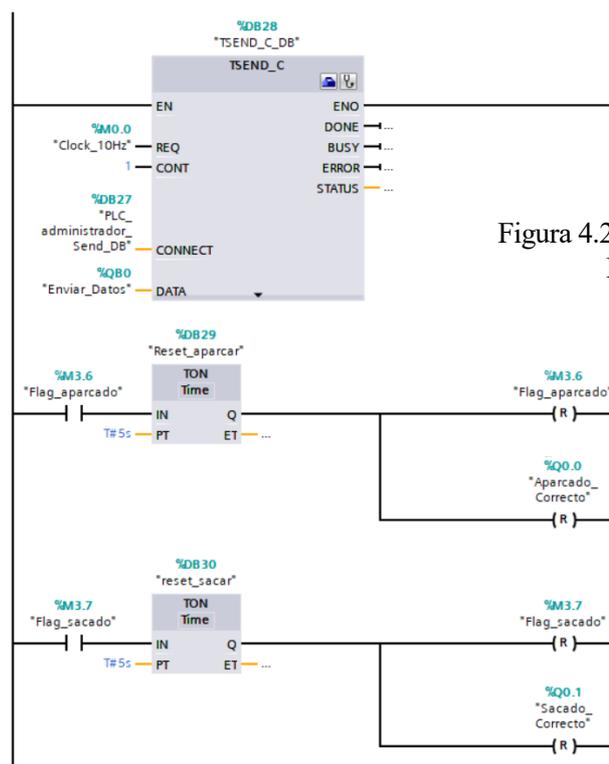


Figura 4.22 Programación Segmento-5 PLC_administrador

Nombre	Tipo de datos	Dirección	Comentario
Flag_aparcado	Bool	%M3.6	Bandera aparcado
Aparcado_Correcto	Bool	%Q0.0	Aparcado
Flag_aparcado	Bool	%M3.7	Bandera sacado
Sacado_Correcto	Bool	%Q0.1	Sacado

Tabla 4.12 Variables Segmento-5 PLC_administrador.

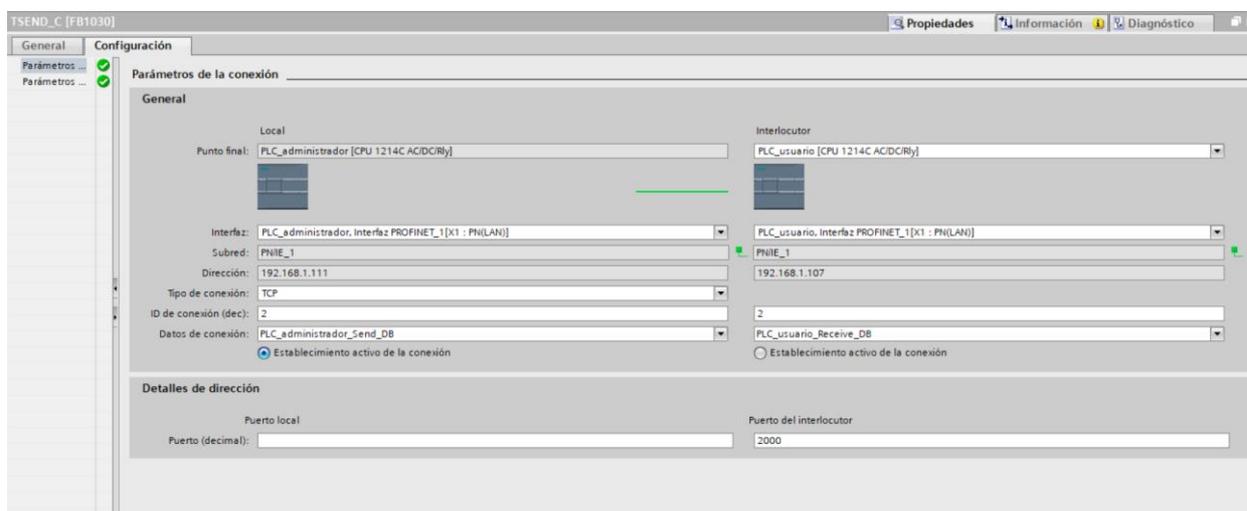


Figura 4.23 TSEND_C Parámetros de la conexión, administrador.



Figura 4.24 TSEND_C Parámetros del bloque, administrador.

5 SCADA

Los Sistemas de Supervisión de Control y Adquisición de Datos (SCADA, Supervisory Control And Data) permiten la gestión y control de cualquier sistema local o remoto gracias a una interfaz gráfica que comunica al Usuario con el sistema. Sin embargo hay dos niveles superiores al SCADA, véase la figura 5.1.

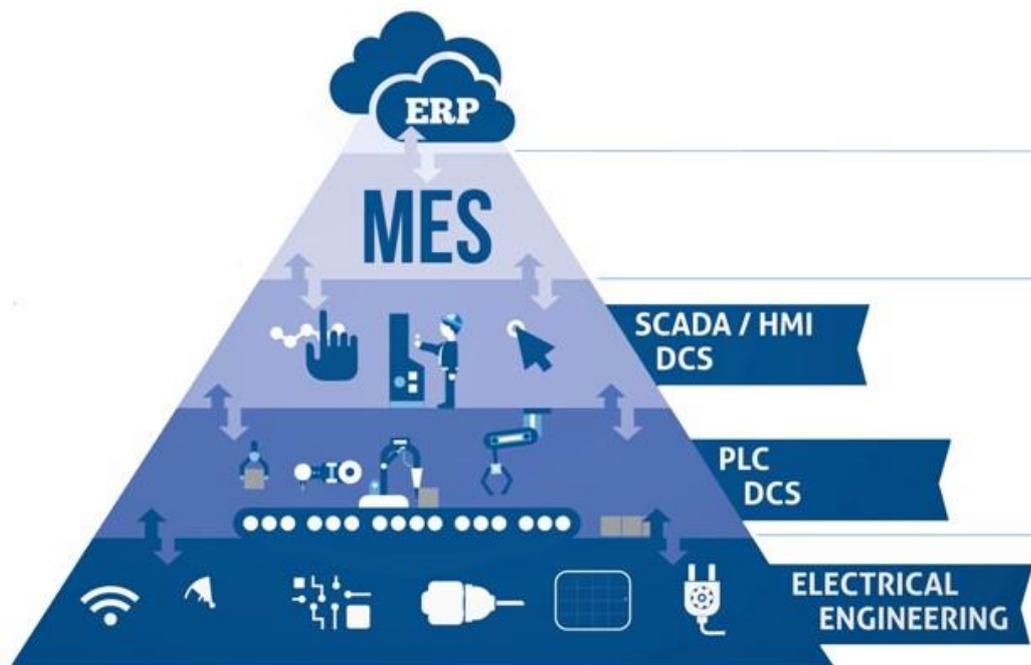


Figura 5.1 Pirámide de Automatización.

El término ERP se refiere a *Enterprise Resource Planning*, que significa “sistema de planificación de recursos empresariales”. Estos programas se hacen cargo de distintas operaciones internas de una empresa, desde producción a distribución o incluso recursos humanos. Y los sistemas de ejecución de fabricación (MES) son sistemas computarizados que se utilizan en la fabricación para rastrear y documentar la transformación de las materias primas en productos terminados. MES proporciona información que ayuda a los responsables de la toma de decisiones en la fabricación a entender cómo se pueden optimizar las condiciones actuales en el piso de la planta para mejorar la producción.

En este capítulo se va a realizar una descripción general de los principales objetivos de un SCADA y de ciertas normas a cumplir que resultan interesantes para el diseño que se ha llevado a cabo en este proyecto.

5.1. Concepto de Control

Cualquier sistema controlado puede serlo de dos maneras, según los caminos de la información dentro del mismo:

- Confiando absolutamente en que los parámetros de diseño son correctos y que las órdenes que se envíen serán cumplidas.
- Vigilando continuamente que las órdenes enviadas se cumplen y realizando las correcciones adecuadas siempre que sea necesario.

Todos los sistemas de control que se engloban dentro del primer modelo reciben el nombre de *Sistemas de Regulación en Lazo Abierto*. Se basan en el diseño adecuado de los parámetros y las condiciones de trabajo del

elemento control para que éste sea capaz de mantener el sistema controlado dentro de los límites deseados. A los que se engloban dentro de la segunda definición, se les denomina *Sistemas de Regulación en Lazo Cerrado*, o Realimentados, y trabajan vigilando continuamente las reacciones del sistema a controlar, efectuando las acciones de corrección necesarias para mantener el control dentro de los límites deseados.

5.2. Sistemas de supervisión y mando

Damos el nombre de Scada (*Supervisory Control And Data Acquisition* o Control con Supervisión y Adquisición de Datos) a cualquier software que permita el acceso a datos remotos de un proceso y permita, utilizando las herramientas de comunicación necesarias en cada caso, el control del mismo. Atendiendo a la definición vemos que no se trata de un sistema de control, sino de una utilidad software de monitorización o supervisión, que realiza la tarea de interfase entre los niveles de control (PLC) y los de gestión a un nivel superior.

Los sistemas SCADA se conciben principalmente como una herramienta de supervisión y mando. Entre sus objetivos podemos destacar:

- **Economía:** es más fácil ver qué ocurre en la instalación desde la oficina que enviar a un operario a realizar la tarea. Ciertas revisiones se convertirán en innecesarias.
- **Accesibilidad:** un parque eólico al completo (velocidad de cada rotor, producción de electricidad), lo tenemos en un *clic* de ratón encima de la mesa de trabajo. Será posible modificar los parámetros de funcionamiento de cada aerogenerador, poniendo fuera de servicio los que den indicios de anomalías; consultar el estado de las estaciones transformadoras del parque, detener los molinos que no sean necesarios, etc.
- **Mantenimiento:** la adquisición de datos materializa la posibilidad de obtener datos de un proceso, almacenarlos y presentarlos de manera inteligible para un usuario no especializado. La misma aplicación se puede programar de manera que avise cuando se aproximen las fechas de revisión o cuando una máquina tenga más fallos de los considerados normales.
- **Ergonomía:** es la ciencia que procura hacer que la relación entre el usuario y el proceso sea lo más confortable posible.
- **Gestión:** todos los datos recopilados pueden ser valorados de múltiples maneras mediante herramientas estadísticas, gráficas, valores tabulados, etc., que permitan explotar el sistema con el mejor rendimiento posible.
- **Flexibilidad:** cualquier modificación de alguna de las características del sistema de visualización no significa un gasto en tiempo y medios, pues no hay modificaciones físicas que requieran la instalación de un cableado o del contador.
- **Conectividad:** se buscan sistemas abiertos. La documentación de los protocolos de comunicación actuales permite la interconexión de sistemas de diferentes proveedores y evita la existencia de lagunas informativas que puedan causar fallos en el funcionamiento o en la seguridad.

Entre las prestaciones de una herramienta de este tipo destacan:

- **Monitorización**
Representación de datos en tiempo real a los operadores de planta. Se leen los datos de los autómatas (temperaturas, velocidades, detectores...).
- **Supervisión**
Supervisión, mando y adquisición de datos de un proceso y herramientas de gestión para la toma de decisiones (mantenimiento predictivo, por ejemplo). Tienen además la capacidad de ejecutar programas que puedan supervisar y modificar el control establecido y, bajo ciertas condiciones, anular o modificar tareas asociadas a los autómatas. Evita una continua supervisión humana.

- **Adquisición de datos**

Un sistema de captación solar se puede observar mediante herramientas registradoras y obtener así un valor medio de la irradiación en la zona, guardando los valores obtenidos y evaluándolos a posterioridad.

- **Visualización de alarmas y eventos**

Reconocimiento de eventos excepcionales acaecidos en la planta y su inmediata puesta en conocimiento de los operarios para efectuar las acciones correctoras pertinentes.

- **Mando**

Posibilidad de que los operadores puedan cambiar consignas u otros datos claves del proceso directamente desde el ordenador (marcha, paro, modificación de parámetros...). Se escriben datos sobre los elementos de control.

- **Grabación de acciones o recetas**

Por ejemplo, en las líneas de vulcanizado en continuo (donde fabrican los perfiles de goma de las ventanas, por ejemplo) se componen de varias máquinas encadenadas con múltiples parámetros (velocidad y temperatura principalmente) que dependen del tipo de perfil a elaborar. Con una sola pulsación se pueden poner en marcha todas las máquinas y programar las diferentes zonas de temperatura o velocidad de toda la línea del ejemplo.

- **Seguridad de los datos**

Tanto el envío como la recepción de datos deben de estar suficientemente protegidos de influencias no deseadas, intencionadas o no (fallos en la programación, intrusos, situaciones inesperadas, etc.).

- **Seguridad en los accesos**

Restringiendo zonas de programa comprometidas a usuarios no autorizados, registrando todos los accesos y acciones llevadas a cabo por cualquier operador.

- **Programación numérica**

Permite realizar cálculos aritméticos de elevada resolución sobre la CPU del ordenador (lenguajes de alto nivel, C y Visual Basic, generalmente).

5.2.1. Principios generales de diseño

Se pueden definir varios principios generales que son aplicables a cualquier tipo de interacción hombre-máquina:

5.2.1.1. Consistencia y coherencia

Si la aplicación es consistente, las secuencias de trabajo son similares en situaciones similares:

- Los tipos de letra son siempre los mismos en toda la aplicación.
- Los deslizadores, las posiciones de los pulsadores, los colores de las ventanas son de tipo único.
- Las pantallas tienen rasgos característicos que las diferencian entre sí. Por ejemplo, las pantallas de alarma con un color de fondo diferente al resto.

Gracias al principio de coherencia, se consigue una interfase amigable y predecible, agilizando las respuestas del usuario.

5.2.1.2. Correspondencia

Hace que la interfaz sea familiar y predecible al adecuarse al sistema real (representación de pulsadores o selectores de una máquina, representación en pantalla de un documento a rellenar, tal como el impreso de la declaración de la Renta).

5.2.1.3. Adecuación a la tarea

El programa proporciona las herramientas de interactividad necesarias para lograr la máxima eficiencia y eficacia en el desarrollo del diálogo con el operador (ayudas contextuales).

5.2.1.4. Autodescriptividad

Cada etapa de la relación entre operador y máquina es capaz de indicar cómo se debe desarrollar ésta (ayudas contextuales, pantallas de ayuda, gráficos animados del proceso).

5.2.1.5. Controlabilidad

El usuario debe poder controlar en todo momento el desarrollo de la relación desde su inicio hasta su finalización. Pasos consecutivos con posibilidad de ir hacia atrás en cualquier momento, guardar el trabajo ya hecho para ser retomado posteriormente, cancelar una acción. Esto ayuda a que el usuario pierda el miedo a tocar cosas que pueden convertirse en serios problemas y le anima a convertirse en explorador de la aplicación.

5.2.1.6. Predictibilidad

El usuario ve una respuesta del sistema acorde con sus conocimientos, formación y experiencia.

5.2.1.7. Tolerancia a fallos y control de errores

Se deben contemplar los posibles errores del usuario y limitarlos, impedirlos o corregirlos de forma automática si ello es posible. Por ejemplo:

- Utilizar listas en la entrada de datos de tipo formulario.
- Limitación de valores numéricos en la entrada de datos.
- Auto corrección de textos.
- Posibilitar recuperación de errores (opción *deshacer*).

Ante un error de cualquier tipo en una aplicación, deberíamos saber cuatro cosas básicas: ¿Qué ha ocurrido? ¿Por qué? ¿Es muy serio el problema? ¿Tiene arreglo?

5.2.1.8. Flexibilidad y eficiencia de uso

El sistema debe ser práctico de utilizar y poder adaptarse a las necesidades y conocimientos del usuario mediante modificaciones simples. Por ejemplo, sistemas de diseño gráfico ampliables según tipo de licencia o privilegios de acceso, posibilidad de uso de atajos para usuarios expertos. Suponga un SCADA que permita supervisar una planta química, y que por su complejidad se deben tener varias pantallas principales. Cuando se produce un error, un usuario que sepa que el error se produce en un cierto sensor, debe tener la posibilidad de acceder directamente a la pantalla que controla ese sensor sin tener que pasar por el resto.

5.2.1.9. Facilidad de aprendizaje

Debe facilitar el aprendizaje mediante herramientas de ayuda, documentación adecuada y estímulos al usuario.

5.2.2. Colores

Cuando se utilizan colores primarios con sus opuestos secundarios se obtiene el llamado *Contraste Complementario*, el cual es útil para destacar y crear impacto pero oblige a forzar el enfoque del ojo creando estrés visual. Si se utilizan colores adyacentes, se obtiene el Contraste de Tonos y se puede dar contrastes mas fuertes con los colores secundarios que los primarios.

Usando colores como Fuente de información, se acepta la siguiente convención de colores (Real Decreto 485/97):

Estados: marcha - abierto	VERDE
Estados: parado - cerrado	ROJO
Estados: atención - preparado	AMARILLO
Alarmas: atención - prealarma	AMARILLO
Alarmas: alarma	ROJO
Alarma: sin alarma	GRIS, INVISIBLE
Elementos: metal	GRIS
Fondos	GRIS, VERDE, AZUL

Tabla 5.1 Colores según el estado.

5.3. Principios generales de diseño

5.3.1. Norma ISO 9241

La norma UNE-EN ISO 9241, en su parte 10, *Principios de diálogo*, trata el diseño ergonómico de programas para equipos con Pantallas de Visualización de Datos. Enumera una serie de ideas que se pretende sirvan de guía a la hora de realizar el planteamiento y desarrollo de las interfaces gráficas:

- La aplicación debe estar adaptada a la tarea para la cual se ha diseñado; el diálogo con el usuario debe ser limpio, presentando y exigiendo solamente la información estrictamente necesaria.
- La aplicación debe informar del progreso al interlocutor de forma comprensible para éste (auto descriptividad).
- La aplicación debe poder adaptarse al nivel de capacitación del usuario.
- La aplicación debe ser controlable por el usuario, no al revés.
- Las respuestas de la interfase deben ser coherentes y adaptadas al nivel de capacitación del usuario.
- La aplicación debería ser tolerante a fallos y con herramientas de corrección automáticas.
- Debería ser clara y sencilla de utilizar.

5.3.2. Elementos gráficos

5.3.2.1. El color de las pantallas

Similar al apartado 5.2.2, el uso de alto contraste se puede convertir en algo incómodo o casi ilegible. Los colores extremos no deberían aparecer simultáneamente en la pantalla (por ejemplo, rojo y azul). Además, no es conveniente convertir las pantallas en ejercicios de diseño incluyendo más colores de los necesarios ya que resultan poco prácticas.

5.3.2.2. El fondo de pantalla

Generalmente se deben usar colores neutros para el fondo (gris, beige, arena, azul). No se debe usar blanco o negro dado que dan resplandor. Y debe contrastar con el resto de elementos. Se puede usar diferentes colores de fondo para diferenciar o agrupar procesos o áreas de la planta.

5.4. Proyecto_FINAL pantallas HMI

5.4.1. HMI_usuario

En este subapartado se van a indicar las distintas pantallas diseñadas para la interfaz de usuario con sus respectivos objetos. Se hará una indicación de los eventos y animaciones usados en cada uno. En la figura 5.2 se puede encontrar un esquema de las imágenes empleadas en este dispositivo.

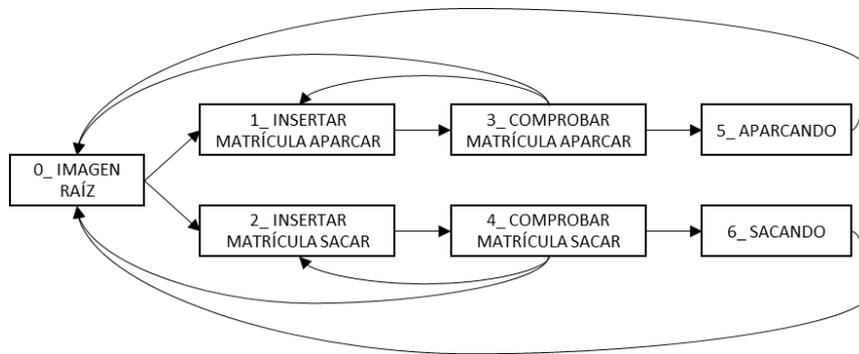


Figura 5.2 Esquema imágenes HMI_usuario.

Partiendo de la plantilla de la figura 5.3, se procede a indicar de cada una de las imágenes, sus capas, objetos usados en cada una de ellas y su respectiva función.

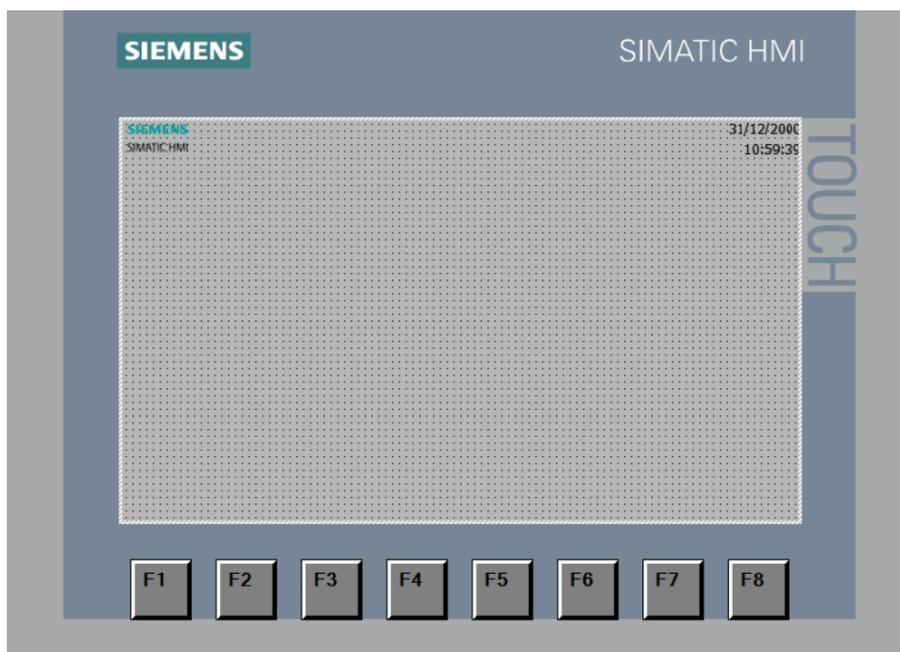


Figura 5.3 Plantilla HMI_usuario.

- 0_Imagen raíz.

Se trata del menú inicial donde primeramente da la bienvenida al usuario con un texto e indicándole que escoja una opción. Las dos opciones son botones que activan la variable correspondiente a aparcar o sacar y activan las pantallas correspondientes. Más concretamente, la opción aparcar lleva a la pantalla 1_Insertar matrícula aparcar y la opción sacar a la pantalla 2_Insertar matrícula sacar. Además, cabe destacar que aparcar contiene una animación dado que cuando el parking está completo, el botón debe desaparecer. Véase la figura 5.4.



Figura 5.4 0_Imagen raíz.

Cabe destacar que la propia imagen tiene eventos, relacionados con el reseteo de banderas. Resetea Matricula_noencontrada, Flag_aparcado y Flag_sacado.

- 1_Insertar matrícula aparcar y 2_Insertar matrícula sacar.
 - Capa 0.

Esta capa contiene únicamente un campo de texto, “INSERTE SU MATRÍCULA” y un campo de entrada-salida que permite indicar la matrícula que está introduciendo el usuario. La configuración del campo entrada-salida se muestra en la figura 5.5 y la apariencia de esta capa se puede ver en la figura 5.6.

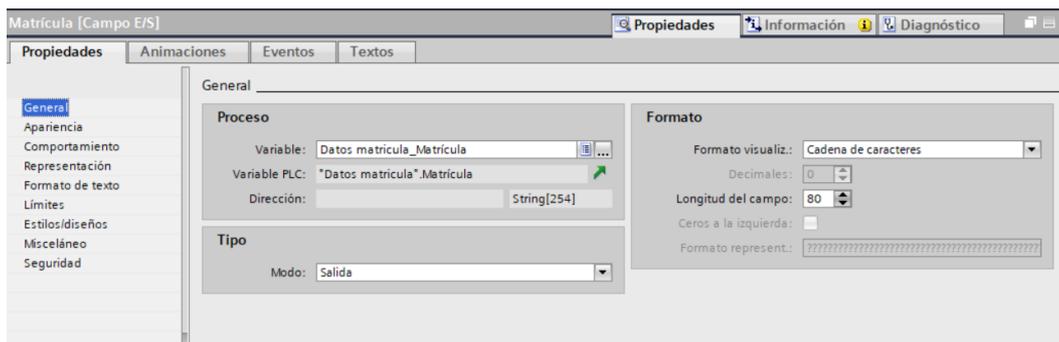


Figura 5.5 Capa 0, campo entrada-salida de Insertar matrícula.

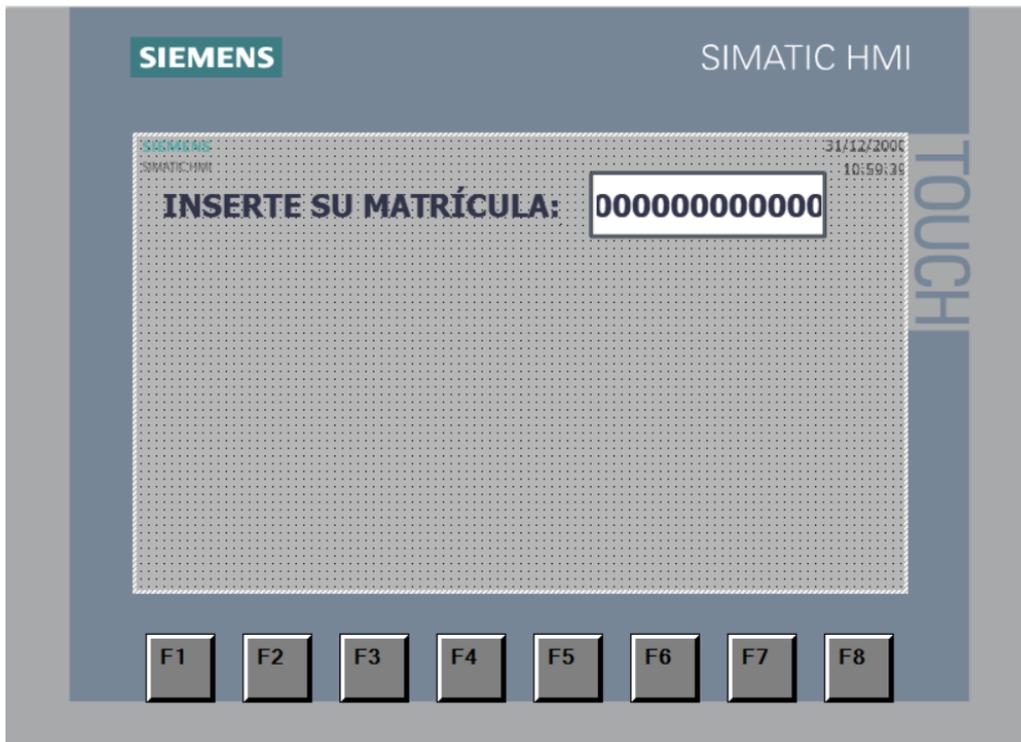


Figura 5.6 Capa 0, Insertar matrícula.

- Capa 1.

Es la que contiene todos los botones que permiten crear un teclado para introducir la matrícula. Se puede observar en la figura 5.7. Cabe destacar que todos estos botones contienen un evento, cada uno asignado a su función. Por ejemplo, la tecla 1 activa la variable que contiene el PLC_usuario y permite escribir un 1 en la matrícula. La tecla A, hace lo mismo pero asociado a dicha letra. La tecla borrar se indica mediante “←”, y el enter redirige a la siguiente pantalla, que según estemos en aparcar o sacar irá a su comprobación.

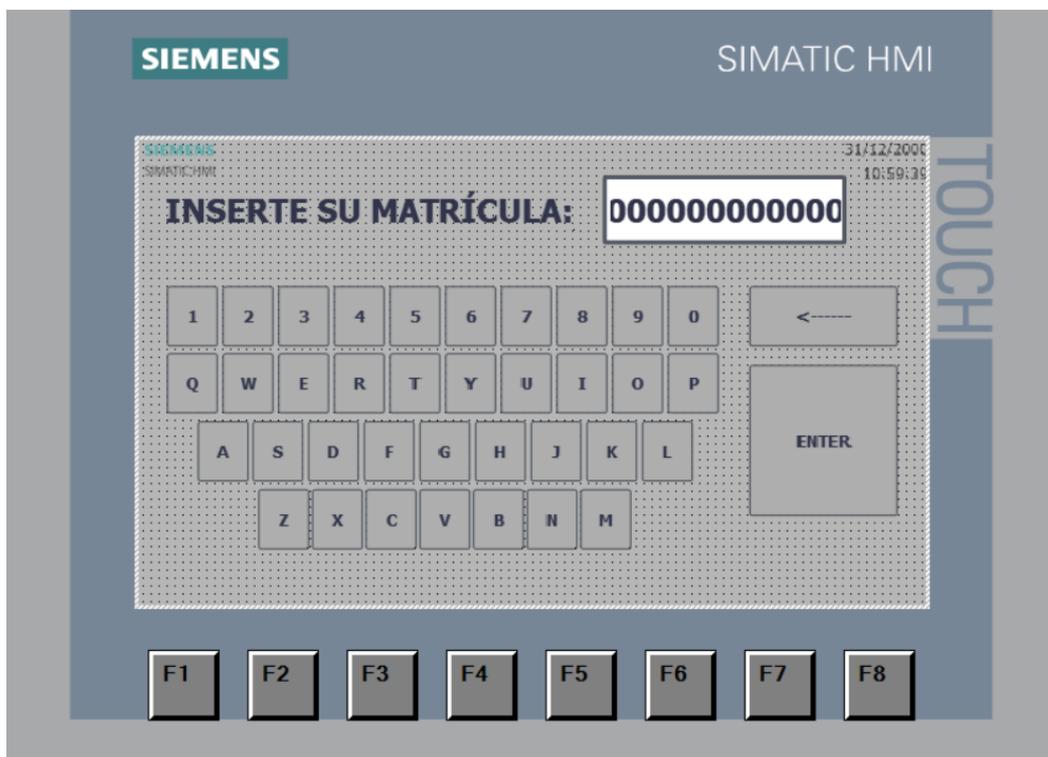


Figura 5.7 Capa 0 y 1, Insertar matrícula.

La diferencia entre que sea 1_Insertar matrícula aparcar o 2_Insertar matrícula sacar es que cuando se activa el botón ENTER, una dirige a Comprobar matrícula aparcar y otra a Comprobar matrícula sacar.

- 3_Comprobar matrícula aparcar y 4_Comprobar matrícula sacar.

En estas pantallas se da la opción de volver al inicio pulsando el botón que a su vez activa Reset, se puede dar a que la matrícula no es correcta y volver al teclado para introducirla correctamente o decir que sí es correcta lo cual activa Matrícula_correcta y pasa a la siguiente pantalla. Se usa de nuevo un campo de entrada-salida para indicarle la matrícula al usuario para la comprobación. La apariencia se puede observar en la figura 5.8.

Esto hace que el PLC_usuario pueda calcular donde aparcarse o de donde sacar el coche. Ocurre lo mismo que en el caso anterior, cuando se comprueba la matrícula en 3_Comprobar matrícula aparcarse y se asegura la matrícula por parte del usuario, se dirige a la pantalla de 5_Aparcando y en el caso de 4_Comprobar matrícula sacar ocurre lo mismo, pero para 6_Sacando. Al igual que en 0_Imagen raíz, el fondo vuelve a resetear Flag_aparcado y Flag_sacado para evitar problemas de que dé el proceso por realizado sin que éste haya ocurrido.

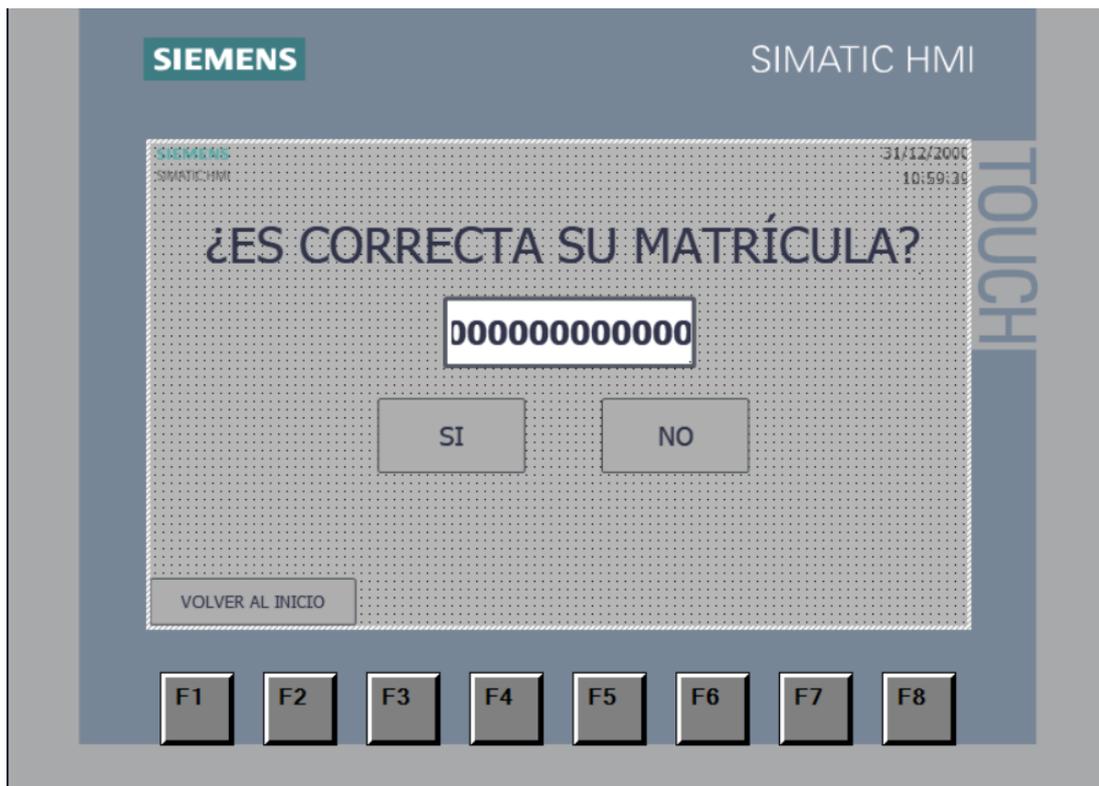


Figura 5.8 Comprobar matrícula.

- 5_Aparcando.

Para hacerle ver al usuario el proceso de aparcado, se ha optado por realizar varias capas. En la capa 0 se puede encontrar que se le indica al usuario que se está aparcando y que se espere. El botón de volver tiene una animación para que solo se pueda volver al inicio cuando se ha terminado el proceso. Para poder escribir en la pantalla de forma sencilla, se ha decidido crear en la capa 1 un rectángulo que tape el texto de la capa 0. Los rectángulos aparecen, al igual que ciertos textos, mediante las animaciones. Por ejemplo, aparecen cuando la variable Flag_Aparcado está activa. Y así en la capa 2 se puede indicar al usuario mediante un texto que ya se ha aparcado. El rectángulo y el texto que indica que el proceso ya se ha acabado aparecen cuando el llega al usuario la información de que se ha aparcado correctamente. En las figuras 5.9, 5.10 y 5.11 se pueden encontrar las capas.

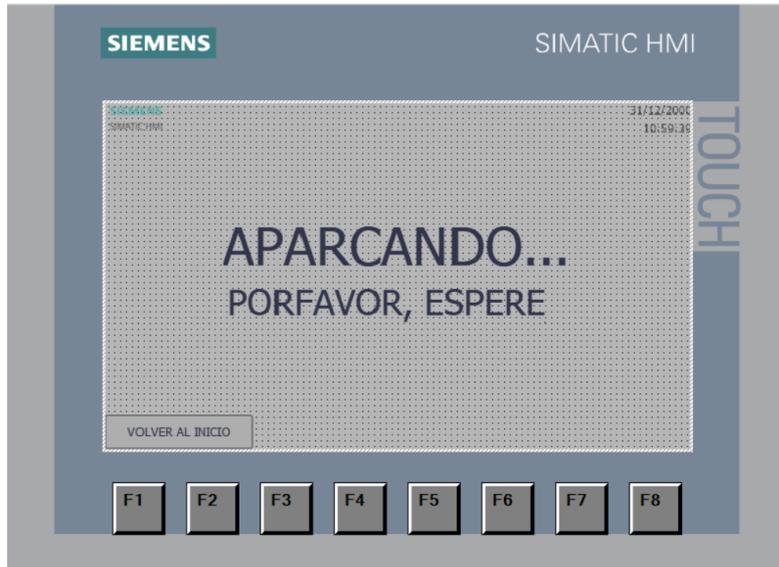


Figura 5.9 Capa 0, 5_Aparcando.

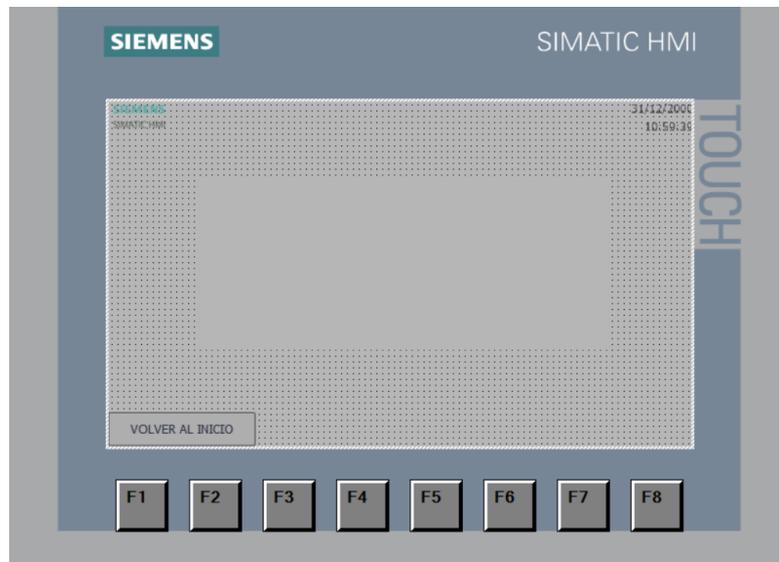


Figura 5.10 Capa 1, 5_Aparcando.

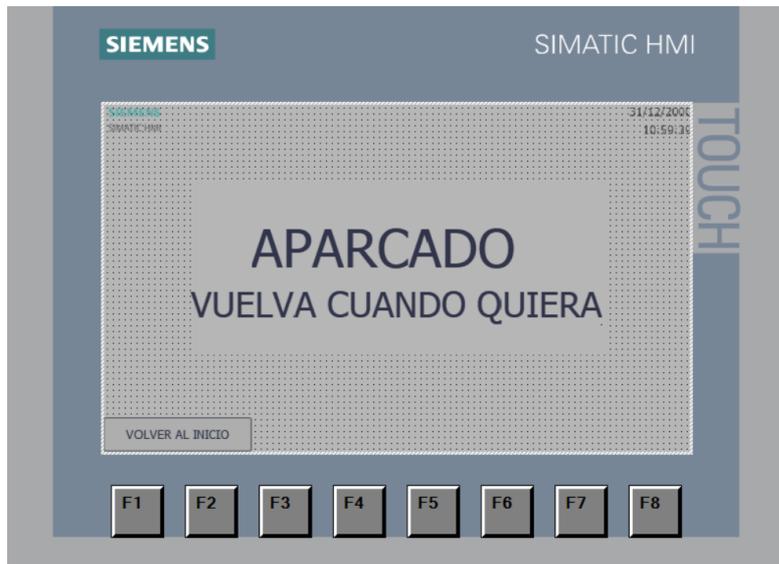


Figura 5.11 Capa 2, 5_Aparcando.

- 6_Sacando

La idea es exactamente la misma que en 5_Aparcando, pero en esta imagen aparte de indicarle que se esta sacando o que ya se ha sacado, también se le tiene que indicar si la matrícula introducida no se encuentra en el parking.

Se tienen igualmente 3 capas, la capa 0 tiene el proceso de sacando sin que aparezca la opción de volver al inicio reseteando. La capa 1 indica que ya se ha sacado, dando la opción de volver al inicio y usando un rectángulo para tapar el texto de la capa 0. Y por ultimo la capa 2 que indica el fallo en la matrícula, tapando el resto y dando la opción de volver al inicio. En las figuras 5.12, 5.13 y 5.14 se pueden encontrar las capas.

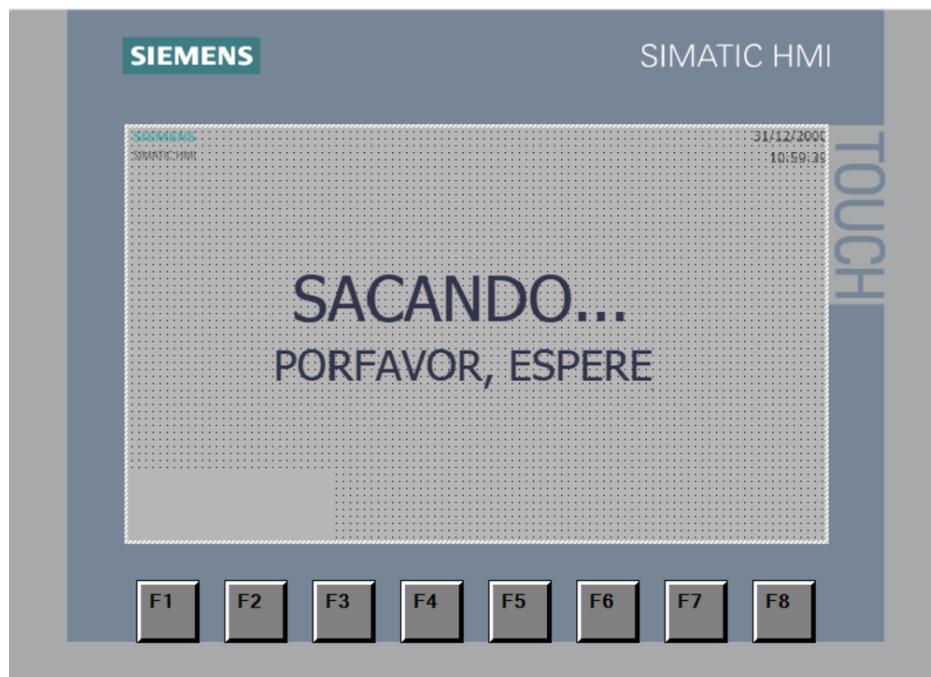


Figura 5.12 Capa 0, 5_Sacando.

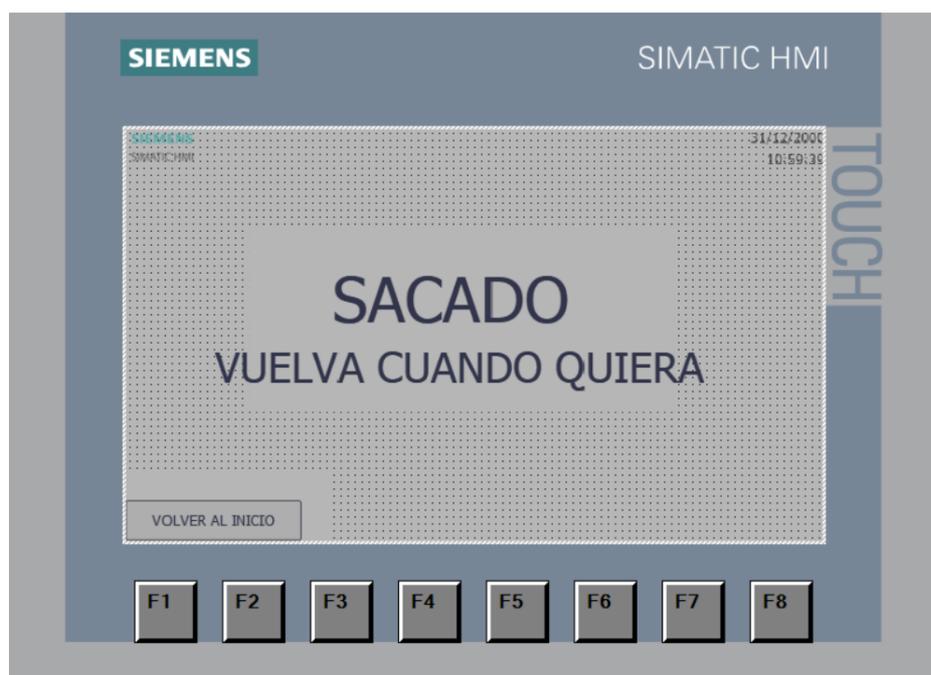


Figura 5.13 Capa 1, 5_Sacando.



Figura 5.14 Capa 2, 5_Sacando.

5.4.2. HMI_administrador

A diferencia del dispositivo anterior, este solamente tiene una pantalla. Esta pantalla es la que tiene que hacer el SCADA y por ello es conveniente que se pueda ver en una pantalla lo que ocurre.

La idea es que el operario pueda ver el proceso que se está haciendo y si surge algún error. Para eso se ha propuesto la pantalla que se puede ver en la figura 5.15, cada proceso se indica con un leds según las normas indicadas al inicio de este capítulo. Al tener la conexión con el PLC, el operario puede activar el reset/reinicializar del proceso del PLC y llevarlo a la posición inicial. Se supone que el botón reset incluye un motor aparte para que si los sensores relacionados con la grúa se estropean, el coche no quede suspendido. Hasta que el operario no considere oportuno quitar el reset/reinicializar, el parking se encuentra en pausa.

El error de seguridad está relacionado con el sensor de seguridad, y el de grúa con los sensores de posición e inicio.

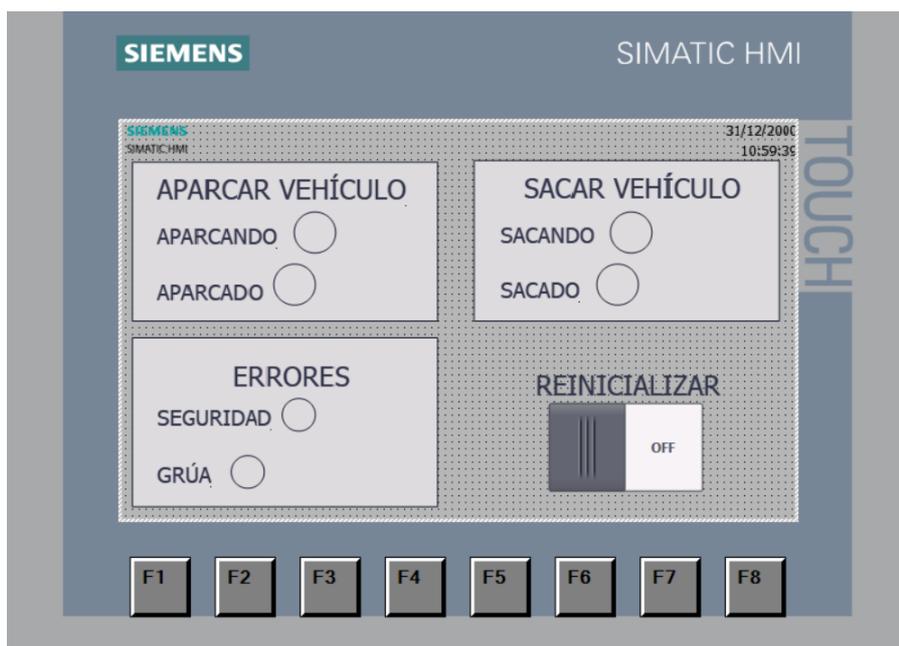


Figura 5.15 Única imagen HMI_administrador.

6 MEJORAS IMPLEMENTADAS

Partiendo de que el Proyecto_Final es el que se ha probado en físico en el laboratorio, cabe destacar que a la hora de hacer la memoria se ha visto una falta en la programación. Y es que cuando se produce un error, con la programación propuesta el usuario estaría minutos esperando sin saber que pasa o quizás días si la avería tarda en solucionarse, y su coche quedaría atrapado dentro del parking.

Para poder solucionar esta falta se ha modificado ligeramente el código del PLC_administrador, haciendo que informe también al PLC_usuario si se ha producido algún error y así a través de una variable notificar al usuario por la HMI_usuario. Además, también se ha modificado que si el botón de reinicializar que maneja el operario se encuentra activo durante más de un minuto se produzca un paro en el parking y se le permita al usuario retirar su coche.

Esto obligaría a volver a inicializar todos los dispositivos e introducir de forma manual en la base de datos los coches que se encuentran en el parking antes de volver a abrirlo al público. Resulta una tarea pesada, pero si se mira por parte del usuario, si se produjera un error que tarde en solucionarse, no querrá perder su coche durante días.

6.1. PLC_usuario

En el PLC_usuario solamente cambia el último segmento, figura 6.1. Para poder activar que le ha llegado un error por parte del administrador y así comunicarse con la HMI. También se ha modificado el cuándo recibe información dado que se encontró una forma más útil, ya que aquí no depende de que pulse o que no pulse el usuario. La variable Errores está relacionada con los errores de los sensores y la de Fallo con el paro del parking, por tardar mas de 1 minuto en arreglar los errores estando en pausa.

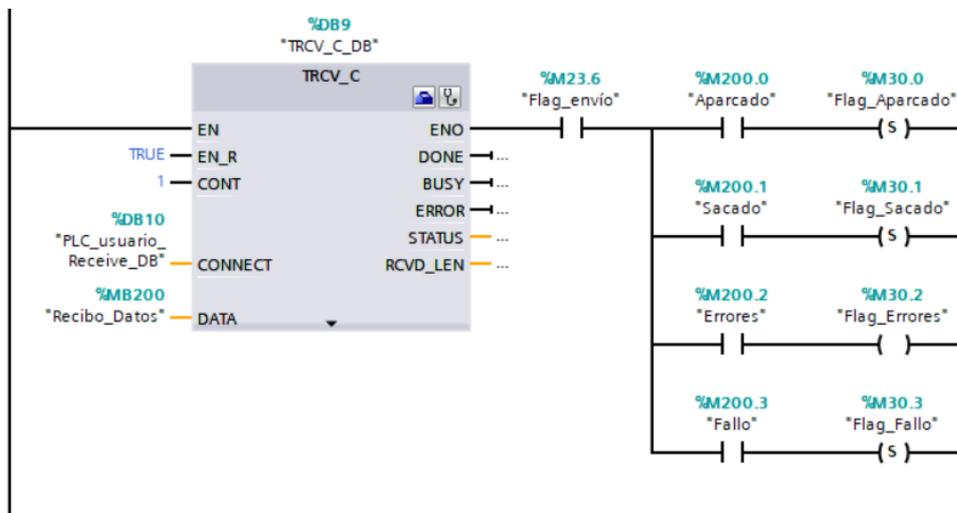


Figura 6.1 Modificación Segmento5 PLC_usuario.

6.2. PLC_administrador

En el PLC_administrador se activan las banderas que permiten mandar la información de los errores al cliente y se tiene en cuenta cuanto tiempo se tiene la pausa activa para parar el parking, ya que cuando pasen 60 segundos debe pararlo por completo. Imposibilitando la entrada de más coches o la salida de alguno que ya esté en su

interior. Al usar un set para activar la bobina de Fallo, es decir stop del parking, y no resetearla en ningún sitio está obligando a que para poder volver a arrancar el parking se tenga que volver a arrancar todos los dispositivos y modificando la base de datos a mano para introducir los coches que aún quedan dentro, como se ha mencionado anteriormente. Solamente se modifica el último segmento al igual que con el PLC_usuario. En la figura 6.2 se puede encontrar cómo resulta finalmente la programación del segmento.

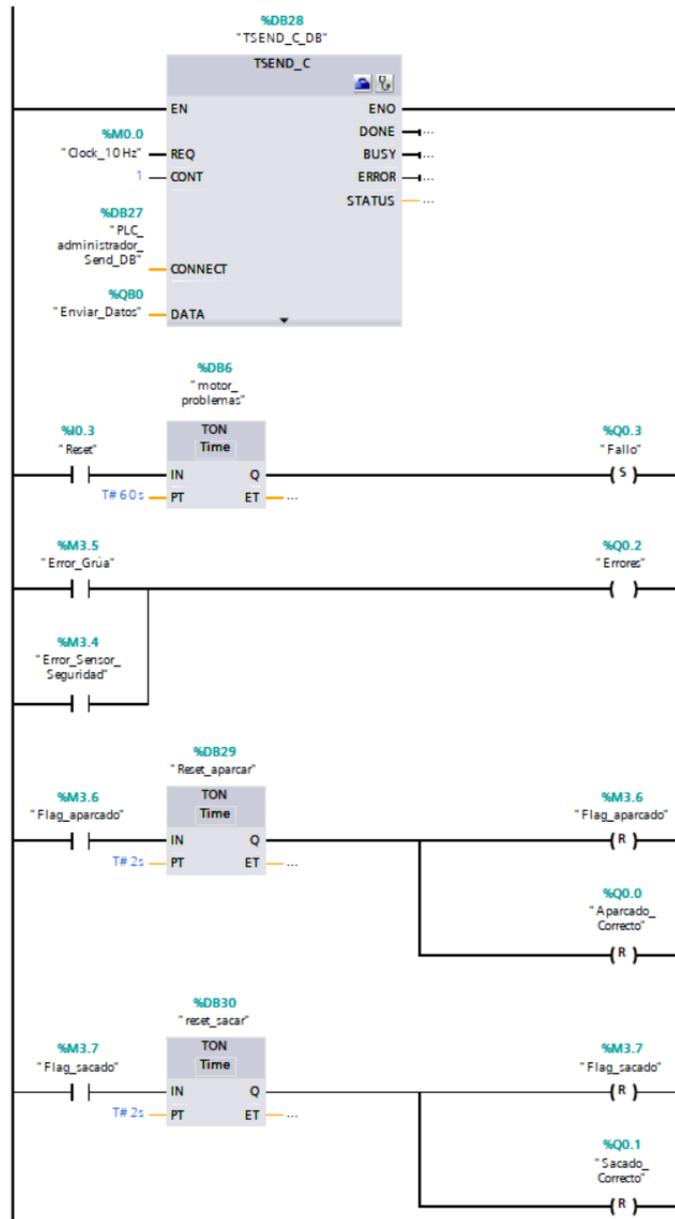


Figura 6.2 Modificación Segmento5 PLC_administrador.

6.3. HMI_usuario

En la HMI_usuario, solamente resultan modificadas las pantallas 5_Aparcando y 6_Sacando. Además, resultan modificadas de la misma forma. Partiendo de la idea de crear rectángulos que tapen el texto que se encontraba antes, ahora aparece una nueva capa para indicarle al usuario que se ha producido un error. Si se produce el parón en el parking hay otra pantalla para indicar que retire su coche y no deja irse a ninguna otra pantalla para que no se puedan sacar ni aparcar más coches.

En las figuras 6.3 y 6.4 se pueden encontrar las dos nuevas capas.

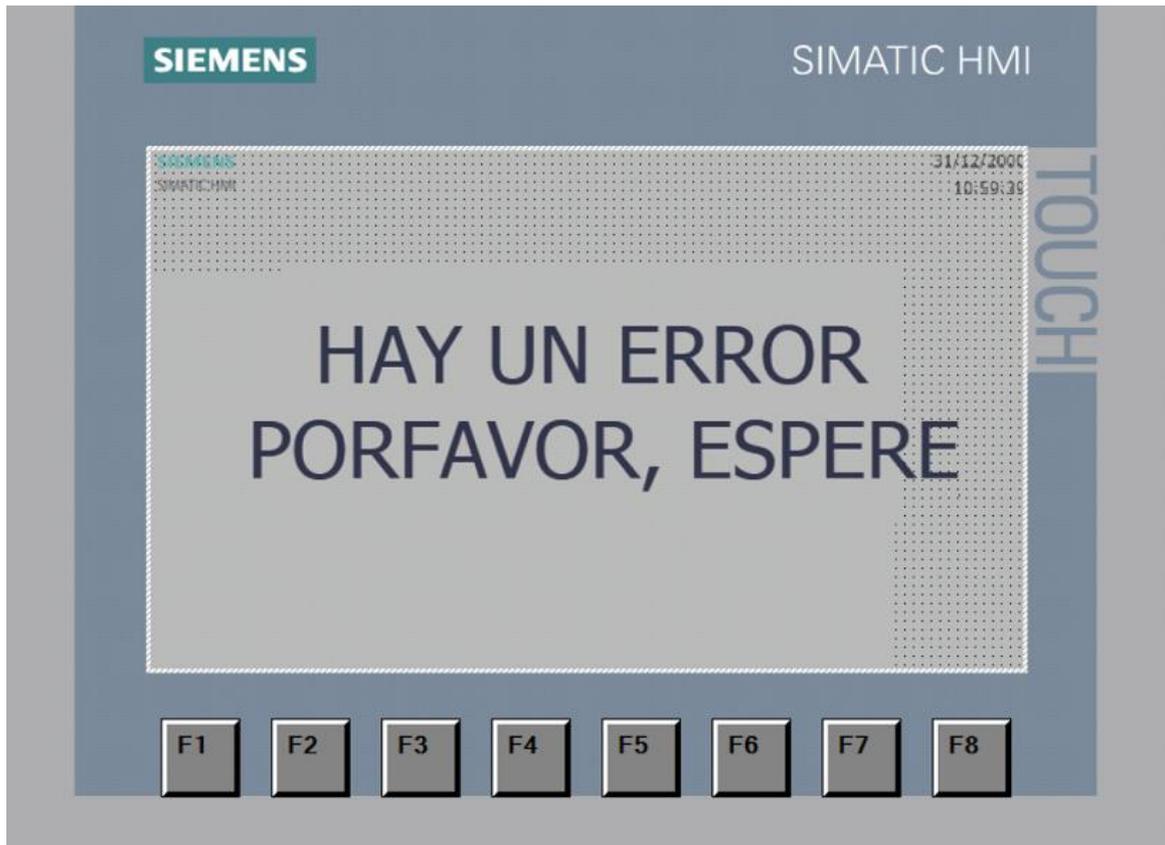


Figura 6.3 Nueva capa para errores, HMI_usuario.

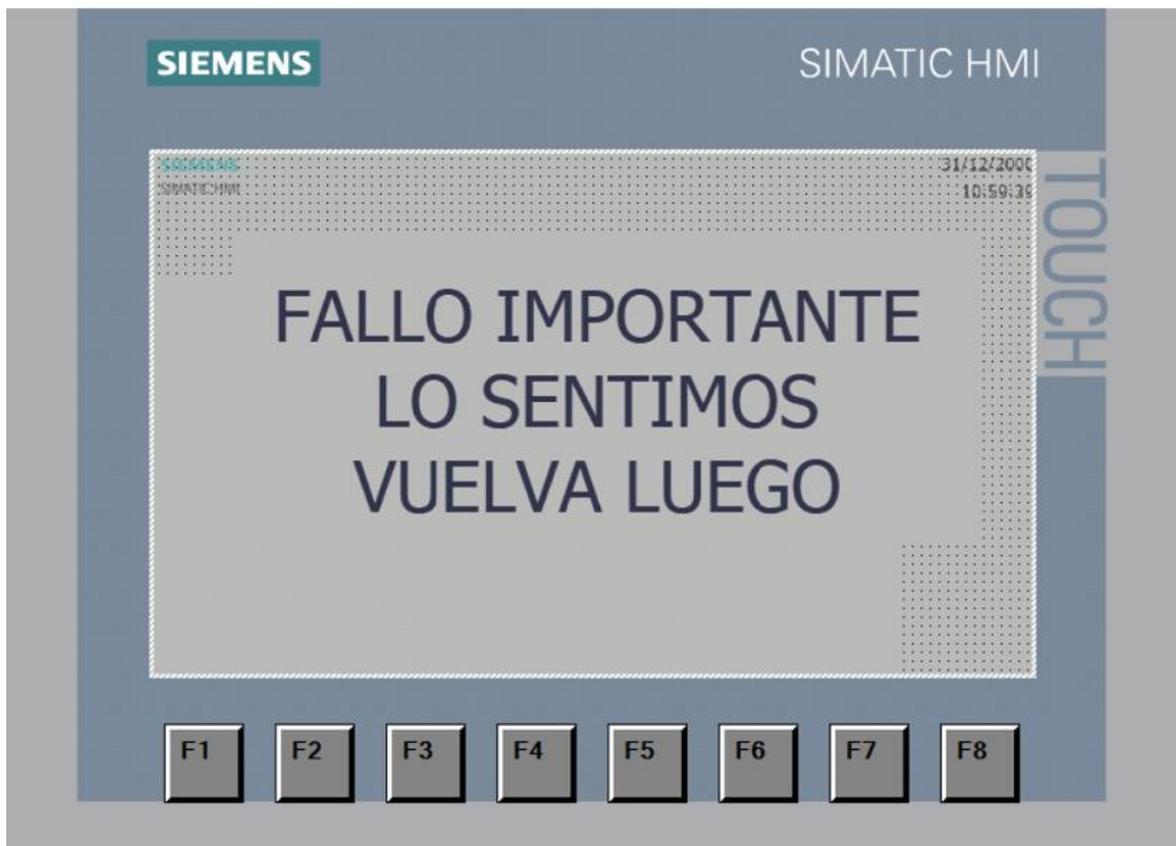


Figura 6.4 Nueva capa para fallo, HMI_usuario.

6.4. HMI_administrador

En la HMI_administrador únicamente se le ha incluido un LED para indicar cuando el parking se ha parado. El resto es igual, se puede observar en la figura 6.5.



Figura 6.5 Nueva imagen HMI_administrador

A pesar de no tener la posibilidad de implementar esta modificación en los dispositivos físicos, se puede realizar la simulación propia del programa para ver que funciona correctamente.

7 CONCLUSIONES Y FUTUROS TRABAJOS

El objetivo de este TFG consistía en el diseño e implementación de la automatización para un parking, así como de un sistema SCADA para la supervisión y control del mismo. Aunque ya existen numerosos parking automatizados en la realidad, como se ha mencionado al principio de esta memoria, el objetivo era ser capaz de hacerlo partiendo de cero con cuatro dispositivos principales, PLCs y pantallas HMI.

Resulta evidente pensar que existen numerosas formas de programar este trabajo, pero la programación propuesta se ha elaborado de la forma más simple posible haciendo uso del conocimiento adquirido durante las prácticas de la carrera y con la propia librería del programa, por lo que es fácil de comprender incluso para alguien que nunca haya usado el programa.

Al separar el trabajo en dos áreas, la del cliente y la del administrador del parking, se pueden programar por separado gran parte del trabajo hasta que sea necesario comunicarlos. Es decir, la pantalla HMI y el PLC encargado de hacer la interfaz para el usuario puede programarse casi completamente sin necesidad de comunicarse con el administrador, exceptuando el tramo final en el que se le indica al usuario que se ha realizado el proceso o que ha ocurrido un error. Y de igual forma, tomando alguna entrada de la botonera como la opción elegida por el cliente, se puede realizar el proceso de aparcar o sacar el vehículo sin necesidad de comunicación con el cliente.

Partiendo de que lo más complicado a primera vista es establecer correctamente las comunicaciones entre los dispositivos, este tipo de configuración permite avanzar la programación sin necesidad de un correcto funcionamiento de la comunicación a priori. Aunque, lógicamente, para un correcto funcionamiento de la automatización propuesta es necesario que la comunicación funcione.

Resumiendo, el Trabajo Fin de Grado propuesto plantea una posible alternativa a los parking automatizados existentes hoy en día, partiendo de una programación con funciones relativamente simples.

Como posibles futuros trabajos relacionados con este mismo Trabajo de Fin de Carrera, está la posibilidad de aprender a utilizar las funciones relacionadas con el tiempo para así poder calcular el tiempo que cada coche pasa dentro del parking y proponer un coste por hora. También relacionado con este tema se podría diseñar otro tipo de plataforma que permita insertar dos grúas a la vez, estableciendo una automatización para que éstas no choquen en ningún momento y se reduzcan los tiempos de espera por parte del cliente. Y por último, pero posiblemente la más interesante, crear una maqueta que permita introducir sensores reales y así sustituir los simulados en la programación propuesta.

REFERENCIAS

E. Pérez-López (2015) Los sistemas SCADA en la automatización industrial. Tecnología en Marcha, 28 (4), pp. 3-14.

Comité de Automatización. Importancia de la Automatización en tiempo de crisis
http://www.aie.cl/files/file/comites/ca/articulos/automat_tiempos_de_crisis10-9.pdf Último acceso: 30 de Agosto, 2018.

Noticia del periódico El País 2/3/2016 https://elpais.com/ccaa/2016/03/02/madrid/1456947278_361561.html
Último acceso: 30 de Agosto, 2018.

Siemens. S7 Controlador programable S7-1200. Manual de Sistema, 378pp.

A. Rodríguez (2007) Sistemas SCADA Guía Práctica. Ed. Marcombo, 256pp.

ANEXOS

Se encuentran como Anexos los siguientes documentos:

- Datasheet HMI KTP700 Basic PN.
- Datasheet S71200 CPU 1214C AC-DC-Rly.

SIMATIC HMI, KTP700 BASIC, BASIC PANEL, KEY AND TOUCH OPERATION, 7" TFT DISPLAY, 65536 COLORS, PROFINET INTERFACE, CONFIGURATION FROM WINCC BASIC V13/ STEP7 BASIC V13, CONTAINS OPEN SOURCE SW WHICH IS PROVIDED FREE OF CHARGE FOR DETAILS SEE CD



Product type designation

Display

Design of display	TFT widescreen display, LED backlighting
Screen diagonal	7 in
Display width	154.1 mm
Display height	85.9 mm
Number of colors	65 536

Resolution (pixels)

• Horizontal image resolution	800
• Vertical image resolution	480

Backlighting

• MTBF backlighting (at 25 °C)	20 000 h
• Dimmable backlight	Yes

Control elements

Keyboard

• Function keys	
— Number of function keys	8
• Keys with LED	No
• System keys	No
• Numeric/alphabetical input	
— Numeric keyboard	Yes; Onscreen keyboard
— Alphanumeric keyboard	Yes; Onscreen keyboard

Touch operation

• Design as touch screen	Yes
--------------------------	-----

Installation type/mounting	
Mounting in portrait format possible	Yes
Mounting in landscape format possible	Yes
maximum permissible angle of inclination without external ventilation	35°
Supply voltage	
Type of supply voltage	DC
Rated value (DC)	24 V
permissible range, lower limit (DC)	19.2 V
permissible range, upper limit (DC)	28.8 V
Input current	
Current consumption (rated value)	230 mA
Starting current inrush I ² t	0.2 A ² ·s
Power	
Power consumption, typ.	5.5 W
Processor	
Processor type	ARM
Memory	
Flash	Yes
RAM	Yes
Usable memory for user data	10 Mbyte
Type of output	
Acoustics	
• Buzzer	Yes
• Speaker	No
Time of day	
Clock	
• Hardware clock (real-time clock)	Yes
• Software clock	Yes
• Battery-backed	Yes
• Synchronizable	Yes
Interfaces	
Number of RS 485 interfaces	0
Number of USB interfaces	1; Up to 16 GB
Number of 20 mA interfaces (TTY)	0
Number of RS 232 interfaces	0
Number of RS 422 interfaces	0
Number of parallel interfaces	0
Number of other interfaces	0
Number of SD card slots	0

With software interfaces	No
Industrial Ethernet	
• Number of industrial Ethernet interfaces	1
• Industrial Ethernet status LED	2
Protocols	
PROFINET	Yes
PROFINET IO	No
IRT	No
PROFIBUS	No
MPI	No
Protocols (Ethernet)	
• TCP/IP	Yes
• DHCP	Yes
• SNMP	Yes
• DCP	Yes
• LLDP	Yes
WEB characteristics	
• HTTP	No
• HTML	No
Further protocols	
• CAN	No
• MODBUS	Yes; Modicon (MODBUS TCP/IP)
• Supports protocol for EtherNet/IP	Yes
Interrupts/diagnostics/status information	
Diagnostic messages	
• Diagnostic information readable	No
EMC	
Emission of radio interference acc. to EN 55 011	
• Limit class A, for use in industrial areas	Yes
• Limit class B, for use in residential areas	No
Degree and class of protection	
IP (at the front)	65
Enclosure Type 4 at the front	Yes
Enclosure type 4x at the front	Yes
IP (rear)	20
Standards, approvals, certificates	
CE mark	Yes
KC approval	Yes
cULus	Yes
RCM (formerly C-TICK)	Yes

Use in hazardous areas	
• ATEX Zone 2	No
• ATEX Zone 22	No
• cULus Class I Zone 1	No
• cULus Class I Zone 2, Division 2	No
• FM Class I Division 2	No

Ambient conditions

Ambient temperature in operation

• Operation (vertical installation)	
— For vertical installation, min.	0 °C
— For vertical installation, max.	50 °C
• Operation (max. tilt angle)	
— At maximum tilt angle, min.	0 °C
— At maximum tilt angle, min.	40 °C
• Operation (vertical installation, portrait format)	
— For vertical installation, min.	0 °C
— For vertical installation, max.	40 °C
• Operation (max. tilt angle, portrait format)	
— At maximum tilt angle, min.	0 °C
— At maximum tilt angle, min.	35 °C

Storage/transport temperature

• Min.	-20 °C
• max.	60 °C

Relative humidity

• Operation, max.	90 %
-------------------	------

Operating systems

Pre-installed operating system	
• Windows CE	No
proprietary	Yes

Configuration

Message indicator	Yes
Alarm logging system (incl. buffer and acknowledgment)	Yes
Process value display (output)	Yes
Process value default (input) possible	Yes
Recipe administration	Yes

Configuration software

• STEP 7 Basic (TIA Portal)	Yes; via integrated WinCC Basic (TIA Portal)
• STEP 7 Professional (TIA Portal)	Yes; via integrated WinCC Basic (TIA Portal)
• WinCC flexible Compact	No
• WinCC flexible Standard	No

• WinCC flexible Advanced	No
• WinCC Basic (TIA Portal)	Yes
• WinCC Comfort (TIA Portal)	Yes
• WinCC Advanced (TIA Portal)	Yes
• WinCC Professional (TIA Portal)	Yes

Languages

Online languages	
• Number of online/runtime languages	10
Languages	
• Languages per project	32
• Languages	
— D	Yes
— GB	Yes
— F	Yes
— I	Yes
— E	Yes
— Chinese traditional	Yes
— Chinese simplified	Yes
— DK	Yes
— FIN	Yes
— GR	Yes
— J	Yes
— KP / ROK	Yes
— NL	Yes
— N	Yes
— PL	Yes
— P	Yes
— RUS	Yes
— S	Yes
— CZ	Yes
— TR	Yes
— H	Yes

Functionality under WinCC (TIA Portal)

Libraries	Yes
Number of Visual Basic Scripts	No
Task planner	Yes
• time-controlled	No
• task-controlled	Yes
Help system	Yes
• Number of characters per info text	500
Message system	

• Number of alarm classes	32
• S7 alarm number procedure	No
• System messages HMI	Yes
• System messages, other (SIMATIC S7, Sinumerik, Simotion, etc.)	Yes; System message buffer of the SIMATIC S7-1200 and S7-1500
• Number of characters per message	80
• Number of process values per message	8
• Acknowledgment groups	Yes
• Message indicator	Yes
• Message buffer	Yes
— Number of entries	256
— Circulating buffer	Yes
— retentive	Yes
— maintenance-free	Yes
Recipe administration	
• Number of recipes	50
• Data records per recipe	100
• Entries per data record	100
• Size of internal recipe memory	256 kbyte
• Recipe memory expandable	No
Variables	
• Number of variables per device	800
• Number of variables per screen	100
• Limit values	Yes
• Multiplexing	Yes
• Structs	No
• Arrays	Yes
Images	
• Number of configurable images	250
• Permanent window/default	Yes
• Global image	Yes
• Start screen configurable	Yes
• Image selection by PLC	Yes
• Image number in the PLC	Yes
Image objects	
• Number of objects per image	100
• Text fields	Yes
• I/O fields	Yes
• Graphic I/O fields (graphics list)	Yes
• Symbolic I/O fields (text list)	Yes
• Date/time fields	Yes

• Switches	Yes
• Buttons	Yes
• Graphic view	Yes
• Icons	Yes
• Geometric objects	Yes
Complex image objects	
• Number of complex objects per screen	10
• Alarm view	Yes
• Trend view	Yes
• User view	Yes
• Status/control	No
• Sm@rtClient view	No
• Recipe view	Yes
• f(x) trend view	No
• System diagnostics view	Yes; System message buffer of the SIMATIC S7-1200 and S7-1500
• Media Player	No
• Bar graphs	Yes
• Sliders	No
• Pointer instruments	No
• Analog/digital clock	No
Lists	
• Number of text lists per project	150
• Number of entries per text list	100
• Number of graphics lists per project	100
• Number of entries per graphics list	100
Archiving	
• Number of archives per device	2
• Number of entries per archive	10 000
• Message archive	Yes
• Process value archive	Yes
• Archiving methods	
— Sequential archive	Yes
— Short-term archive	Yes
• Memory location	
— Memory card	No
— USB memory	Yes
— Ethernet	No
• Data storage format	
— CSV	No
— TXT	Yes

— RDB	No
Security	
• Number of user groups	50
• Number of user rights	32
• Number of users	50
• Password export/import	Yes
• SIMATIC Logon	No
Character sets	
• Keyboard	
— US American (English)	Yes
• Character sets	
— Tahoma	Yes
— Arial	No
— Courier New	No
— WinCC flexible Standard	Yes
— Ideographic languages	Yes
• Font size freely scalable	Yes
Transfer (upload/download)	
• MPI/PROFIBUS DP	No
• USB	No
• Ethernet	Yes
• using external storage medium	No
Process coupling	
• S7-1200	Yes
• S7-1500	Yes
• S7-200	Yes
• S7-300/400	Yes
• LOGO!	Yes
• Win AC	Yes
• SINUMERIK	No
• SIMOTION	Yes
• Allen Bradley (EtherNet/IP)	Yes
• Allen Bradley (DF1)	No
• Mitsubishi (MC TCP/IP)	Yes
• Mitsubishi (FX)	No
• OMRON (FINS TCP)	No
• OMRON (LINK/Multilink)	No
• Modicon (Modbus TCP/IP)	Yes
• Modicon (Modbus)	No
Service tools/configuration aids	
• Clean screen	Yes

• Touch calibration	Yes
• Backup/restore	Yes
• Backup/Restore automatically	No
• Simulation	Yes
• Device switchover	Yes
• Delta transfer	No

I/O / Options

I/O devices	
• Printer	No
• MultiMediaCard	No
• SD card	No
• USB memory	Yes

Mechanics/material

Type of housing (front)	
• plastic	Yes
• aluminum	No
• Stainless steel	No

Dimensions

Width of the housing front	214 mm
Height of housing front	158 mm
Mounting cutout, width	197 mm
Mounting cutout, height	141 mm
Device depth	39 mm

Weights

Without packaging	780 g
With packaging	990 g

last modified: 12.03.2015

SIMATIC S7-1200, CPU 1214C, compact CPU, AC/DC/relay, onboard I/O: 14 DI 24 V DC; 10 DO relay 2 A; 2 AI 0-10 V DC, Power supply: AC 85-264 V AC at 47-63 Hz, Program/data memory 100 KB



General information	
Product type designation	CPU 1214C AC/DC/relay
Firmware version	V4.2
Engineering with	
<ul style="list-style-type: none"> Programming package 	STEP 7 V14 or higher
Supply voltage	
Rated value (AC)	
<ul style="list-style-type: none"> 120 V AC 	Yes
<ul style="list-style-type: none"> 230 V AC 	Yes
permissible range, lower limit (AC)	85 V
permissible range, upper limit (AC)	264 V
Line frequency	
<ul style="list-style-type: none"> permissible range, lower limit 	47 Hz
<ul style="list-style-type: none"> permissible range, upper limit 	63 Hz
Input current	
Current consumption (rated value)	100 mA at 120 V AC; 50 mA at 240 V AC
Current consumption, max.	300 mA at 120 V AC; 150 mA at 240 V AC
Inrush current, max.	20 A; at 264 V

I ² t	0.8 A ² ·s
Output current	
for backplane bus (5 V DC), max.	1 600 mA; Max. 5 V DC for SM and CM
Encoder supply	
24 V encoder supply	
• 24 V	20.4 to 28.8V
Power loss	
Power loss, typ.	14 W
Memory	
Work memory	
• integrated	100 kbyte
• expandable	No
Load memory	
• integrated	4 Mbyte
• Plug-in (SIMATIC Memory Card), max.	with SIMATIC memory card
Backup	
• present	Yes
• maintenance-free	Yes
• without battery	Yes
CPU processing times	
for bit operations, typ.	0.08 µs; / instruction
for word operations, typ.	1.7 µs; / instruction
for floating point arithmetic, typ.	2.3 µs; / instruction
CPU-blocks	
Number of blocks (total)	DBs, FCs, FBs, counters and timers. The maximum number of addressable blocks ranges from 1 to 65535. There is no restriction, the entire working memory can be used
OB	
• Number, max.	Limited only by RAM for code
Data areas and their retentivity	
Retentive data area (incl. timers, counters, flags), max.	10 kbyte
Flag	
• Number, max.	8 kbyte; Size of bit memory address area
Local data	
• per priority class, max.	16 kbyte; Priority class 1 (program cycle): 16 KB, priority class 2 to 26: 6 KB
Address area	
Process image	
• Inputs, adjustable	1 kbyte

- Outputs, adjustable

1 kbyte

Hardware configuration

Number of modules per system, max. 3 comm. modules, 1 signal board, 8 signal modules

Time of day

Clock

- Hardware clock (real-time) Yes
- Backup time 480 h; Typical
- Deviation per day, max. ± 60 s/month at 25 °C

Digital inputs

Number of digital inputs 14; Integrated
 • of which inputs usable for technological functions 6; HSC (High Speed Counting)

Source/sink input Yes

Number of simultaneously controllable inputs

all mounting positions

— up to 40 °C, max. 14

Input voltage

- Rated value (DC) 24 V
- for signal "0" 5 V DC at 1 mA
- for signal "1" 15 V DC at 2.5 mA

Input delay (for rated value of input voltage)

for standard inputs

- parameterizable 0.2 ms, 0.4 ms, 0.8 ms, 1.6 ms, 3.2 ms, 6.4 ms and 12.8 ms, selectable in groups of four
- at "0" to "1", min. 0.2 ms
- at "0" to "1", max. 12.8 ms

for interrupt inputs

- parameterizable Yes

for technological functions

- parameterizable Single phase: 3 @ 100 kHz & 3 @ 30 kHz, differential: 3 @ 80 kHz & 3 @ 30 kHz

Cable length

- shielded, max. 500 m; 50 m for technological functions
- unshielded, max. 300 m; For technological functions: No

Digital outputs

Number of digital outputs 10; Relays

Switching capacity of the outputs

- with resistive load, max. 2 A
- on lamp load, max. 30 W with DC, 200 W with AC

Output delay with resistive load

- "0" to "1", max. 10 ms; max.

• "1" to "0", max.	10 ms; max.
Relay outputs	
• Number of operating cycles, max.	mechanically 10 million, at rated load voltage 100 000
Cable length	
• shielded, max.	500 m
• unshielded, max.	150 m
Analog inputs	
Number of analog inputs	2
Input ranges	
• Voltage	Yes
Input ranges (rated values), voltages	
• 0 to +10 V	Yes
• Input resistance (0 to 10 V)	≥100k ohms
Cable length	
• shielded, max.	100 m; twisted and shielded
Analog outputs	
Number of analog outputs	0
Analog value generation for the inputs	
Integration and conversion time/resolution per channel	
• Resolution with overrange (bit including sign), max.	10 bit
• Integration time, parameterizable	Yes
• Conversion time (per channel)	625 μs
Encoder	
Connectable encoders	
• 2-wire sensor	Yes
1. Interface	
Interface type	PROFINET
Physics	Ethernet
Isolated	Yes
automatic detection of transmission rate	Yes
Autonegotiation	Yes
Autocrossing	Yes
Interface types	
• Number of ports	1
• integrated switch	No
Protocols	
• PROFINET IO Controller	Yes
• PROFINET IO Device	Yes
• SIMATIC communication	Yes

• Open IE communication	Yes
• Web server	Yes
• Media redundancy	No
PROFINET IO Controller	
• Transmission rate, max.	100 Mbit/s
Services	
— PG/OP communication	Yes
— S7 routing	Yes
— Isochronous mode	No
— Open IE communication	Yes
— IRT	No
— MRP	No
— MRPD	No
— PROFINergy	No
— Prioritized startup	Yes
— Number of IO devices with prioritized startup, max.	16
— Number of connectable IO Devices, max.	16
— Number of connectable IO Devices for RT, max.	16
— of which in line, max.	16
— Activation/deactivation of IO Devices	Yes
— Number of IO Devices that can be simultaneously activated/deactivated, max.	8
— Updating time	The minimum value of the update time also depends on the communication component set for PROFINET IO, on the number of IO devices and the quantity of configured user data.
PROFINET IO Device	
Services	
— PG/OP communication	Yes
— S7 routing	Yes
— Isochronous mode	No
— Open IE communication	Yes
— IRT	No
— MRP	No
— MRPD	No
— PROFINergy	Yes
— Shared device	Yes
— Number of IO Controllers with shared device, max.	2
Protocols	
Supports protocol for PROFINET IO	Yes

PROFIBUS	Yes; CM 1243-5 required
AS-Interface	Yes; CM 1243-2 required
Protocols (Ethernet)	
• TCP/IP	Yes
• DHCP	No
• SNMP	Yes
• DCP	Yes
• LLDP	Yes
Open IE communication	
• TCP/IP	Yes
— Data length, max.	8 kbyte
• ISO-on-TCP (RFC1006)	Yes
— Data length, max.	8 kbyte
• UDP	Yes
— Data length, max.	1 472 byte
Web server	
• supported	Yes
• User-defined websites	Yes
Further protocols	
• MODBUS	Yes
Communication functions	
S7 communication	
• supported	Yes
• as server	Yes
• as client	Yes
• User data per job, max.	See online help (S7 communication, user data size)
Number of connections	
• overall	16; dynamically
Test commissioning functions	
Status/control	
• Status/control variable	Yes
• Variables	Inputs/outputs, memory bits, DBs, distributed I/Os, timers, counters
Forcing	
• Forcing	Yes
Diagnostic buffer	
• present	Yes
Traces	
• Number of configurable Traces	2
• Memory size per trace, max.	512 kbyte
Interrupts/diagnostics/status information	

Diagnostics indication LED	
• RUN/STOP LED	Yes
• ERROR LED	Yes
• MAINT LED	Yes

Integrated Functions	
Number of counters	6
Counting frequency (counter) max.	100 kHz
Frequency measurement	Yes
controlled positioning	Yes
Number of position-controlled positioning axes, max.	8
Number of positioning axes via pulse-direction interface	Up to 4 with SB 1222
PID controller	Yes
Number of alarm inputs	4

Potential separation	
Potential separation digital inputs	
• Potential separation digital inputs	500V AC for 1 minute
• between the channels, in groups of	1
Potential separation digital outputs	
• Potential separation digital outputs	Relays
• between the channels	No
• between the channels, in groups of	2

EMC	
Interference immunity against discharge of static electricity	
• Interference immunity against discharge of static electricity acc. to IEC 61000-4-2	Yes
— Test voltage at air discharge	8 kV
— Test voltage at contact discharge	6 kV
Interference immunity to cable-borne interference	
• Interference immunity on supply lines acc. to IEC 61000-4-4	Yes
• Interference immunity on signal cables acc. to IEC 61000-4-4	Yes
Interference immunity against voltage surge	
• on the supply lines acc. to IEC 61000-4-5	Yes
Interference immunity against conducted variable disturbance induced by high-frequency fields	
• Interference immunity against high-frequency radiation acc. to IEC 61000-4-6	Yes
Emission of radio interference acc. to EN 55 011	
• Limit class A, for use in industrial areas	Yes; Group 1
• Limit class B, for use in residential areas	Yes; When appropriate measures are used to ensure compliance with the limits for Class B according to EN 55011

Degree and class of protection	
Degree of protection acc. to EN 60529	
• IP20	Yes
Standards, approvals, certificates	
CE mark	Yes
UL approval	Yes
cULus	Yes
FM approval	Yes
RCM (formerly C-TICK)	Yes
KC approval	Yes
Marine approval	Yes
Ambient conditions	
Free fall	
• Fall height, max.	0.3 m; five times, in product package
Ambient temperature during operation	
• min.	-20 °C
• max.	60 °C; Number of simultaneously activated inputs or outputs 7 or 5 (no adjacent points) at 60 °C horizontal or 50 °C vertical, 14 or 10 at 55 °C horizontal or 45 °C vertical
• horizontal installation, min.	-20 °C
• horizontal installation, max.	60 °C
• vertical installation, min.	-20 °C
• vertical installation, max.	50 °C
Ambient temperature during storage/transportation	
• min.	-40 °C
• max.	70 °C
Air pressure acc. to IEC 60068-2-13	
• Operation, min.	795 hPa
• Operation, max.	1 080 hPa
• Storage/transport, min.	660 hPa
• Storage/transport, max.	1 080 hPa
Altitude during operation relating to sea level	
• Installation altitude, min.	-1 000 m
• Installation altitude, max.	2 000 m
Relative humidity	
• Operation, max.	95 %; no condensation
Vibrations	
• Vibration resistance during operation acc. to IEC 60068-2-6	2 g (m/s ²) wall mounting, 1 g (m/s ²) DIN rail
• Operation, tested according to IEC 60068-2-6	Yes
Shock testing	

<ul style="list-style-type: none"> • tested according to IEC 60068-2-27 	Yes; IEC 68, Part 2-27 half-sine: strength of the shock 15 g (peak value), duration 11 ms
Pollutant concentrations	
<ul style="list-style-type: none"> • SO2 at RH < 60% without condensation 	SO2: < 0.5 ppm; H2S: < 0.1 ppm; RH < 60% condensation-free
Configuration	
Programming	
Programming language	
— LAD	Yes
— FBD	Yes
— SCL	Yes
Know-how protection	
<ul style="list-style-type: none"> • User program protection/password protection 	Yes
<ul style="list-style-type: none"> • Copy protection 	Yes
<ul style="list-style-type: none"> • Block protection 	Yes
Access protection	
<ul style="list-style-type: none"> • Protection level: Write protection 	Yes
<ul style="list-style-type: none"> • Protection level: Read/write protection 	Yes
<ul style="list-style-type: none"> • Protection level: Complete protection 	Yes
Cycle time monitoring	
<ul style="list-style-type: none"> • adjustable 	Yes
Dimensions	
Width	110 mm
Height	100 mm
Depth	75 mm
Weights	
Weight, approx.	455 g
last modified:	08/04/2018

