

GRADO EN INGENIERIA EN TECNOLOGÍA
INDUSTRIAL

TRABAJO FIN DE GRADO

*ADQUISICIÓN DE DATOS DESDE
PLATAFORMA IOT2040. PROTOCOLOS
MODBUS/TCP Y OPC UA*

Alumno: Calvo, Luis, Borja

Director (1): Orive, Revillas, Darío

Director (2): Casquero, Oyarzabal, Oscar

Curso: 2018-2019

Fecha: 24 de Junio de 2019

DATOS BÁSICOS DEL TRABAJO DE FIN DE GRADO

- *Alumno:* Borja Calvo Luis
- *Director:* Darío Orive Revillas
- *Departamento:* Departamento de Ingeniería de Sistemas y Automática
- *Título:* “Adquisición de datos desde plataforma IOT2040. Protocolos Modbus/TCP y OPC UA”
- *Resumen:* Este trabajo se basa en la adquisición de datos de un proceso automatizados desde dispositivo SIMATIC IOT2040 de SIEMENS, con diferentes protocolos industriales, mediante una plataforma de programación (Node-RED). Así, se posibilita la lectura, envío y procesamiento de datos de un proceso en un servidor remoto, facilitando la implementación de la industria 4.0 a través del Internet de las Cosas (IoT) a cualquier proceso automatizado futuro.
- *Palabras clave:* Automatización, IOT, Industria 4.0, Node-RED, Modbus/TCP, OPC UA
- *Izenburua:* Datuen eskuratzea IOT2040 plataformatik protokolo industrialen bidez.
- *Laburpena:* Lan hau automatizazio prozesu datuen eskuratzean oinarritzen da SIMATIC IOT2040 gailuaren bidez, protokolo industrial desberdinekin, programazio plataforma baten (Node-RED) bitartez. Horrela, prozesuko datuen irakurketa, bidalketa eta prozesaketa zerbitzari urrun batean posible egiten da, 4.0 industriaren inplementazioa erraztuz Gauzen Internetaren (IoT) bidez etorkizuneko edozein automatizazio prozesutarako.
- *Hitz-gakoak:* Automatizazioa, IOT, Industria 4.0, Node-RED, Modbus/TCP, OPC UA
- *Title:* Acquisition data from the IOT2040 platform with industrial protocols.
- *Abstract:* This work is based on the acquisition of data from an automated process from SIEMENS SIMATIC IoT2040 device with different industrial protocols through a programming platform (Node-RED). Thus, it is possible to read, send and process data from a process in a remote server, facilitating the implementation of industry 4.0 through the Internet of Things (IoT) to any future automated process.
- *Keywords:* Automation, IOT, Industry 4.0, Node-RED, Modbus/TCP, OPC UA

Índice de Contenido:

Índice de Contenido:	5
Índice de ilustraciones:.....	7
Índice de tablas:	9
1. Introducción	11
2. Contexto.....	12
3. Objetivos y alcance.....	14
4. Beneficios que aporta el trabajo	15
5. Análisis de alternativas.....	17
6. Selección de la solución propuesta	20
7. Metodología. Diseño	25
7.1. Introducción.....	25
7.1.1 Programa de control en el PLC.....	25
7.1.2 Metodología de arranque	30
7.1.2.1 Grabación de imagen del SIMATIC IOT2040	30
7.1.2.2 Descarga de PuTTY, Configuración IP del dispositivo e inicio de sesión en SIMATIC IOT2040.....	31
7.2. Protocolo de comunicación Modbus/TCP	34
7.2.1 Demostrador	36
7.2.2 Analizador de redes eléctricas	37
7.2.3 Desarrollo de la aplicación en Node-RED para Modbus/TCP.....	39
7.2.3.1 Servicio de lectura de datos	40
7.2.3.2 Servicio de escritura de datos	42
7.3. Protocolo OPC UA.....	45
7.3.1 Demostrador	46
7.3.2 Proyecto TIA Portal para el PLC.....	47
7.3.3 Desarrollo de la aplicación en Node-RED para IOT2040	50
7.3.3.1 Servicio de lectura de datos	51
7.3.3.2 Servicio de escritura de datos	53
7.3.3.3 Servicio de suscripción de datos	56
8. Descripción de tareas. Diagrama de Gantt	59
9. Aspectos económicos. Descripción del presupuesto	64
10. Conclusiones.....	66
11. Bibliografía	67

Índice de definiciones y abreviaturas:

- *Siemens SIMATIC IOT2040*: Plataforma accesible y abierta fabricada por SIEMENS que permite el almacenamiento, procesamiento y transmisión de datos desde ámbitos industriales. Esta plataforma permite la integración de la producción con los sistemas IT (Information Technology) de los entornos industriales, facilitando la actualización de las plantas.
- *Node-RED*: Entorno de desarrollo basada en flujos para la programación visual desarrollada originalmente por IBM para conectar dispositivos de hardware, API y servicios en línea como parte del IOT (Internet de las cosas).
- *IOT (Internet Of Things)*: Sistema de dispositivos de automatización interrelacionados, máquinas mecánicas y digitales u objetos con identificadores únicos que cuentan con la capacidad de transferir datos a través de la red de forma autónoma.
- *Industria 4.0*: Nombre dado a la tendencia actual de automatización e intercambio de datos en tecnologías de fabricación. Incluye sistemas cibernéticos, Internet de las cosas, computación en la nube y computación cognitiva. La industria 4.0 se conoce comúnmente como la cuarta revolución industrial.
- *JavaScript*: Lenguaje de programación ligero, orientado al diseño de páginas web, pero también empleado en entornos sin navegador. Se trata de un lenguaje dinámico, basado en protocolos, que soporta estilos de programación funcional.
- *Modelo OSI*: Modelo de referencia sobre como las aplicaciones se comunican a través de una red. El propósito del modelo de referencia OSI es guiar a los proveedores y desarrolladores para que los productos de comunicación digital y los programas de software que se crean puedan interactuar, y para facilitar un marco claro que describa las funciones de un sistema de redes.
- *Yocto Linux*: Proyecto de código abierto colaborativo que tiene como objetivo producir herramientas y procesos que permitan la creación de distribuciones de Linux mediante plantillas y distintas herramientas.

Índice de ilustraciones:

<i>Ilustración 1. Siemens SIMATIC IOT2040.....</i>	<i>11</i>
<i>Ilustración 2. Esquema de revoluciones industriales.....</i>	<i>12</i>
<i>Ilustración 3. Tecnologías digitales en la cuarta revolución industrial.</i>	<i>12</i>
<i>Ilustración 4. Raspberri Pi3 (Modelo B).....</i>	<i>17</i>
<i>Ilustración 5. Arduino UNO.....</i>	<i>18</i>
<i>Ilustración 6. SIMATIC IOT2040.....</i>	<i>19</i>
<i>Ilustración 7.PLC SIMATIC S7-1516.....</i>	<i>21</i>
<i>Ilustración 8. TIA Portal V15.1.....</i>	<i>22</i>
<i>Ilustración 9. Aplicación Node-RED.....</i>	<i>24</i>
<i>Ilustración 10. Símbolo PuTTY.....</i>	<i>24</i>
<i>Ilustración 11. Escalera mecánica.....</i>	<i>25</i>
<i>Ilustración 12. Estructura Programa.....</i>	<i>26</i>
<i>Ilustración 13. Bloque de organización del programa.</i>	<i>29</i>
<i>Ilustración 14. Configuración DB2 para servicio de escritura en OPC UA.....</i>	<i>29</i>
<i>Ilustración 15. Grabación imagen de arranque.....</i>	<i>30</i>
<i>Ilustración 16. Inserción tarjeta de memoria Micro SD en IOT2040.....</i>	<i>30</i>
<i>Ilustración 17. Demostrador para inicio de sesión en SIMATIC IOT2040.....</i>	<i>31</i>
<i>Ilustración 18. Configuración por SSH al dispositivo.</i>	<i>31</i>
<i>Ilustración 19. Cambio de dirección IP por defecto mediante PuTTY.....</i>	<i>32</i>
<i>Ilustración 20. Conexión a IOT2040 con IP modificada.....</i>	<i>32</i>
<i>Ilustración 21. Paleta de nodos cargada a través de PuTTY.....</i>	<i>33</i>
<i>Ilustración 22. Acceso a Node-RED a través del navegador.....</i>	<i>33</i>
<i>Ilustración 23. Comunicación cliente/servidor en Modbus/TCP.....</i>	<i>34</i>
<i>Ilustración 24. Encapsulamiento de la trama Modbus/TCP.....</i>	<i>35</i>
<i>Ilustración 25. Demostrador para Modbus/TCP.....</i>	<i>36</i>
<i>Ilustración 26. Maqueta analizador de redes.....</i>	<i>37</i>
<i>Ilustración 27. Menú de configuración de Node-red.....</i>	<i>39</i>
<i>Ilustración 28. Instalación de paquete de nodos modbustcp.....</i>	<i>39</i>
<i>Ilustración 29. Nodos instalados para protocolo modbustcp.....</i>	<i>39</i>
<i>Ilustración 30. Nodos utilizados para el servicio de lectura.....</i>	<i>40</i>
<i>Ilustración 31. Configuración nodo input modbustcp.....</i>	<i>40</i>
<i>Ilustración 32. Configuración Modbustcp-server.....</i>	<i>41</i>
<i>Ilustración 33. Disposición de nodos para lectura en Modbustcp-server.....</i>	<i>42</i>
<i>Ilustración 34. Valor de las tensiones leídas desde Node-RED.....</i>	<i>42</i>
<i>Ilustración 35. Configuración nodo output modbustcp.....</i>	<i>43</i>
<i>Ilustración 36. Disposición de nodos para escritura en Modbustcp-server.....</i>	<i>43</i>
<i>Ilustración 37. Disposición de nodos para lectura y escritura en Modbus/TCP.....</i>	<i>44</i>
<i>Ilustración 38. Demostrador para OPC UA.....</i>	<i>46</i>
<i>Ilustración 39. Configuración Hardware.</i>	<i>47</i>
<i>Ilustración 40. IP del PLC.....</i>	<i>47</i>
<i>Ilustración 41. Licencia servidor OPC UA.....</i>	<i>48</i>
<i>Ilustración 42.Activación servidor OPC UA.....</i>	<i>48</i>
<i>Ilustración 43. Activación de la dirección del OPC UA.....</i>	<i>48</i>

<i>Ilustración 44. Exportación archivo XML OPC UA</i>	<i>49</i>
<i>Ilustración 45. Archivo XML OPC UA en escritorio</i>	<i>49</i>
<i>Ilustración 46. Variable Personas_E1. Archivo XML OPC UA.</i>	<i>49</i>
<i>Ilustración 47. Menú de configuración de Node-RED</i>	<i>50</i>
<i>Ilustración 48. Instalación de nodo</i>	<i>50</i>
<i>Ilustración 49. Nodos instalados para protocolo OPC UA</i>	<i>51</i>
<i>Ilustración 50. Configuración nodo [Inject]</i>	<i>51</i>
<i>Ilustración 51. Configuración nodo OpcUa-Client para lectura.....</i>	<i>52</i>
<i>Ilustración 52. Configuración nodo inject. Tiempo de funcionamiento.....</i>	<i>52</i>
<i>Ilustración 53. Disposición nodos lectura en OPC UA.....</i>	<i>52</i>
<i>Ilustración 54. Número de personas y tiempo de funcionamiento en Node-RED</i>	<i>52</i>
<i>Ilustración 55. Disposición nodos para servicio de escritura en OPC UA.....</i>	<i>53</i>
<i>Ilustración 56. Configuración nodo [Inject] para servicio de escritura en OPC UA</i>	<i>53</i>
<i>Ilustración 57. Configuración nodo OpcUa-Item.....</i>	<i>53</i>
<i>Ilustración 58. Configuración nodo OpcUa-Client para escritura.....</i>	<i>54</i>
<i>Ilustración 59. Escritura 10 en DB2</i>	<i>54</i>
<i>Ilustración 60. Configuración [Inject] para escritura OPC UA (Bool).....</i>	<i>54</i>
<i>Ilustración 61. Configuración [OpcUa-Item] para escritura OPC UA (Bool)</i>	<i>55</i>
<i>Ilustración 62. Escritura TRUE en DB2.....</i>	<i>55</i>
<i>Ilustración 63. Disposición nodos para servicio de escritura en OPC UA</i>	<i>56</i>
<i>Ilustración 64. Edición nodo OpcUa-Client para suscripción</i>	<i>56</i>
<i>Ilustración 65. Edición nodo OpcUa-Client para suscripción</i>	<i>57</i>
<i>Ilustración 66. Edición nodo OpcUa-Client para suscripción</i>	<i>57</i>
<i>Ilustración 67. Graficas suscripción.</i>	<i>58</i>
<i>Ilustración 68. Diagrama de Gantt (1 de 3).....</i>	<i>61</i>
<i>Ilustración 69. Diagrama de Gantt (2 de 3).....</i>	<i>62</i>
<i>Ilustración 70. Diagrama de Gantt (3 de 3).....</i>	<i>63</i>
<i>Ilustración 71. Partidas Presupuestarias.</i>	<i>65</i>

Índice de tablas:

<i>Tabla 1. Variables programa PLC</i>	26
<i>Tabla 2. Función 10.</i>	27
<i>Tabla 3 Interfaz de bloque</i>	29
<i>Tabla 4. Variables en el DB2</i>	29
<i>Tabla 5. Funciones en Modbus/TCP</i>	35
<i>Tabla 6. Mapa de memoria CVM-MINI</i>	38
<i>Tabla 7. Horas internas</i>	64
<i>Tabla 8. Presupuesto. Amortizaciones</i>	64
<i>Tabla 9. Presupuesto. Gastos</i>	64
<i>Tabla 10. Presupuesto. Resumen</i>	64

1. Introducción

Este documento de Trabajo de Fin de Grado, titulado “Adquisición de datos desde plataforma IOT2040. Protocolos Modbus/TCP y OPC UA”, se basa en la comunicación para la transferencia de datos entre un PLC (Programmable Logic Controller) y un dispositivo SIMATIC IOT2040 de SIEMENS (Ver Ilustración 1). En la primera parte del proyecto, se presenta el contexto, los objetivos y el alcance del mismo. Además, se detallarán los beneficios que aporta la realización del trabajo.



Ilustración 1. Siemens SIMATIC IOT2040

Una vez conocido el funcionamiento del dispositivo, se procederá a establecer la comunicación para la transferencia de datos con los diferentes protocolos industriales (OPC UA y Modbus/TCP), mediante la herramienta de programación visual Node-RED.

Asimismo, se realizará una planificación del proyecto, incorporando, la duración de las tareas, y las fechas de inicio y finalización de cada una de ellas. Posteriormente, se contará con un presupuesto pormenorizando todas sus partidas.

El proyecto terminará exponiendo las conclusiones que se pueden extraer del mismo y su potencial para posibles aplicaciones futuras. Por último, se incluirá la bibliografía utilizada.

2. Contexto

En el Foro Económico mundial del año 2016, su fundador, Klaus Schwab introdujo el concepto de cuarta revolución industrial (ver Ilustración 2), que se basaría en la digitalización de entornos industriales. Este hecho unido a una creciente globalización del mercado, requerirá la necesidad de tecnologías interconectadas entre sí, y de personal cualificando para llevar estas tareas a cabo. La cuarta revolución industrial es también conocida con “Industria 4.0”

La Industria 4.0 (cuarta revolución industrial) es capaz de adecuar sus sistemas de producción a las necesidades del mercado, siendo requerida para ello la interconexión in situ de sus máquinas y sistemas y el intercambio de información con el exterior. La industria 4.0 enfatiza y acentúa la idea de una creciente y adecuada digitalización y coordinación cooperativa en todas las unidades productivas. Uno de los pilares en los que se basa, es la captura de datos de los procesos de fabricación para poder analizar en todo momento el estado del sistema y tomar las acciones más adecuadas.

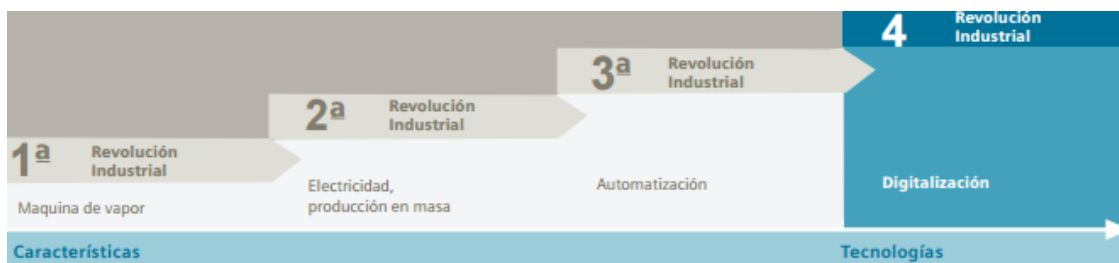


Ilustración 2. Esquema de revoluciones industriales.

Por lo tanto, el aprovechar las ventajas de un mercado único digital, con el progreso de las tecnologías digitales, en combinación con otras tecnologías habilitadoras clave, está cambiando la forma de diseñar, producir, comercializar y generar valor a partir de productos, sean éstos bienes o servicios. Entre estas tecnologías disruptivas digitales, se encuentran la computación en la nube, la movilidad, el big data, la analítica de datos, la impresión 3D, la robótica colaborativa y el Internet de las cosas (IoT) (Ver Ilustración 3).



Ilustración 3. Tecnologías digitales en la cuarta revolución industrial.

Es precisamente esta última, la que proporciona la información del estado de los dispositivos que intervienen en el control del proceso, pero quizás uno de los dispositivos más importante es la CPU del PLC que controlan los procesos. Por lo tanto, teniendo en cuenta las oportunidades que se presentan en este sector, el objetivo del presente trabajo será el desarrollo una aplicación de captura de datos desde la plataforma IOT2040 de Siemens, de la CPU del PLC que controla la plataforma, utilizando diferentes protocolos de comunicación (TCP/IP, S7, Modbus/TCP, OPC UA...). Ya que son los protocolos de comunicación más utilizados actualmente en el entorno industrial.

3. Objetivos y alcance

Debido a la naturaleza del proyecto, los objetivos del presente Trabajo de Fin de Grado son los siguientes:

- Conocimiento de la herramienta de programación visual Node-RED, que permite conectar dispositivos hardware, API y servicios en línea como parte del Internet de las cosas (IoT).
- Comprensión más profunda de los diferentes protocolos de comunicación utilizados en el ámbito industrial. Esto es posible ya que el funcionamiento del SIMATIC IOT 2040 basado en Linux permite utilizar diversos protocolos de comunicación existentes.
- Desarrollo de una aplicación de captura de datos desde la plataforma IOT2040 de SIEMENS, utilizando diferentes protocolos de comunicación (Modbus/TCP, OPC UA). Se utilizará la plataforma IOT2040 debido a que está específicamente diseñado para el ámbito industrial, además de ser capaz de soportar con muchos protocolos y sistemas operativos sin perder la flexibilidad.

En relación al alcance, existen diferentes protocolos de comunicación con los que se puede establecer la conexión entre el PLC y el dispositivo SIMATIC IOT2040. Sin embargo, SIMATIC IOT2040 cuenta con limitaciones que restringen su ámbito de actuación. El dispositivo tiene con un número limitados de entradas/salidas, presenta una capacidad computacional limitada y las tensiones máximas de los pines de entrada/salida que restringen la compatibilidad con dispositivos que trabajen con tensiones diferentes, para evitar así daños en el dispositivo.

Por otro lado, Node-RED es una aplicación basada en node.js, una plataforma JavaScript, que usa un modelo de entradas/salidas para la creación de aplicaciones intensivas en datos y tiempo real, pudiendo funcionar con total fiabilidad a lo largo de multitud de dispositivos de arquitecturas distribuidas. Por tanto, la única posible limitación para trabajar con Node-RED es que el dispositivo a emplear no sea compatible con la librería node.js.

4. Beneficios que aporta el trabajo

En este apartado se detallarán los beneficios que se pueden aportar un proyecto de estas características:

4.1 Mejora de la eficiencia

En un proceso de automatización, los diferentes dispositivos son conectados para el control y seguimiento de procesos industriales, y resolviendo cualquier problema que se pudiese generar durante su ejecución. Por lo tanto, cada vez más sectores, como el manufacturero, incorporan estas tecnologías aumentando la productividad, sustituyendo el trabajo manual y repetitivo por sistemas que trabajan de forma autónoma.

Para seguir aumentando la productividad de las empresas, y por lo tanto mejorar su eficiencia, progresivamente se están incorporando la aplicación de la conectividad (Internet de las cosas) a toda la industria. Todo ello permite recoger datos de gran interés, para una actuación más rápida cuando aparezcan problemas durante el proceso productivo o simplemente para tener un control histórico de todas las operaciones que se han llevado a cabo.

Por lo tanto, una mejora de la eficiencia de determinadas empresas, mediante la incorporación de este tipo de tecnologías permite reducir costes, y aumentar así el tiempo de vida útil de los bienes de capital de los que disponen ciertas industrias.

4.2 Reducción de gastos innecesarios

La conectividad de procesos industriales permite la mejora de los mismos, contribuyendo a su eficiencia y su precisión. La captura de datos de un proceso de automatización también permite identificar gastos innecesarios, para así poder eliminarlos o en su defecto limitarlos. Para ello es necesario que toda la información digital pueda ser recopilada, procesada y analizada de forma que permita reducir ciertos gastos en la toma de decisiones.

4.3 Mejora de la atención al cliente

Los dispositivos IOT posibilitan que los propios consumidores puedan compartir datos en tiempo real y que las empresas puedan reaccionar a tiempo de una mejor manera, para así poder satisfacer los deseos de sus clientes. La capacidad de monitorear los procesos permite detectar y corregir problemas, antes de que los clientes puedan percibirlos, esto es debido a la velocidad de análisis de datos que el internet de las cosas permite. Además, los ordenadores permiten realizar un seguimiento continuo tanto de la calidad del producto, como de su cantidad. Las funciones de identificación y direccionamiento facilitan los procesos de intercambio de datos a nivel empresarial.

4.4 Mantenimiento preventivo

El mantenimiento preventivo, se basa en la detección temprana de signos inminentes de desgaste en base a la evaluación de indicadores relevantes. Los costosos tiempos de inactividad de las plantas de producción se pueden minimizar efectivamente de esta manera. SIMATIC IOT2040 recopila y comunica datos relevantes a herramientas de análisis basados en la nube y, por lo tanto, permite realizar un mantenimiento preventivo.

El carácter sistemático de las operaciones, permite que el mantenimiento preventivo pueda realizarse con cierta frecuencia, ya que este tipo de actuación puede realizarse tanto por horas de funcionamiento de la instalación o por horas de utilización de la misma.

4.5 Big data-IoT

A partir de la extracción de datos que pueden proveer ciertos equipos, se puede lograr aplicar algoritmos de inteligencia artificial en procesos industriales, con el fin de optimizar la producción y prever cualquier tipo de incidente para reducir el coste y aumentar la calidad. La inclusión de geolocalización en los productos, la integración de todos los sistemas y la aplicación de los datos obtenidos en un entorno no seguro durante todo el proceso productivo, sientan las bases para las denominada “fábrica inteligente”. Así, se aumenta la productividad de piezas/año y se obtiene un ahorro de costes significativo en inventario.

4.6 Gestión de archivos

Gestionar archivos que están a miles de kilómetros sin utilizar una red móvil ni wifi, sino un sistema de comunicación que envía datos a la nube es ya una realidad gracias a una tecnología denominada Internet de la Cosas (IoT). Así, se puede conectar cualquier objeto independientemente de su estado, ubicación o forma para que aporte información. De este modo, es posible saber que sucede en lugares remotos a los que no se tiene acceso.

Por un lado, se trabaja esa gestión de activos en tiempo real, normalmente de gran valor y movilidad, y por otro lado, la administración de equipamiento destinado a hacer mantenimientos en sitios remotos, empresas con herramientas y técnicos que están continuamente viajando y que necesitan datos en tiempo real para planificarse.

4.7 Logística

Los dispositivos IoT aplicados a la logística permiten mejorar los procesos productivos y la automatización industrial, con lo que se reducen tiempos y ahorra costes. Por ejemplo, se puede enviar información de en qué condiciones se está realizando un transporte o como se está llevando un almacenamiento de una mercancía concreta. Además, sirve para optimizar los calendarios de los transportistas, lo que disminuye los tiempos de envíos.

5 Análisis de alternativas

En este apartado se presentarán diferentes alternativas que sean competitivas con este proyecto. Por lo tanto, se tratará de analizar diferentes dispositivos, con similares características, con una fácil implementación, a bajo coste y que soporten distintos protocolos de comunicación industriales. Es preferible que las características a analizar tengan interfaces WiFi o Ethernet.

En el mercado contamos con una amplia oferta de dispositivos, pero debido a la documentación que es ofrecida por los fabricantes y la comunidad Open Source, se elegirán los siguientes dispositivos para el análisis: Raspberry Pi, Arduino e IOT2040.

5.1 Raspberri Pi

Esta plataforma consta de una placa base sobre la que se monta un procesador, un chip gráfico y memoria RAM. El procesador funciona a 700 MHz y puede acelerar gráficos 3D por *hardware*. Por lo tanto, es una alternativa a considerar, debido a sus especificaciones técnicas. Además, contiene un puerto Ethernet, un puerto HDMI y cuatro puertos USB.

Se trata de un producto con propiedad registrada, pero manteniendo su uso libre. El software es de código abierto. Cuenta con una gran comunidad Open Source, que hace que tenga una compatibilidad y funcionalidad muy extendida gracias a las distintas librerías y programas existentes y también cuenta con una gran cantidad de accesorios debido a su extendida popularidad. Además, el coste de esta plataforma es bajo.

Tienen unas dimensiones muy pequeñas, son ligeras y se pueden obtener por precios muy bajos (Ver Ilustración 4). Además, posee conectividad Bluetooth, lo cual permite comunicarse con cualquier sistema de automatización de forma inalámbrica.

La principal desventaja de estos dispositivos es que no están diseñados para aplicaciones industriales, por lo tanto, no pueden adaptarse a ciertas industrias, a pesar de contar con distintos accesorios y distintas carcasas que les proporcionaban mayor robustez. Además, es necesario un circuito ADC externo, para poder implementarlo, ya que no consta de pines de entrada analógicos.



Ilustración 4. Raspberri Pi3 (Modelo B)

5.2 Arduino

Arduino es una plataforma de creación de electrónica de código abierto, que está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

El hardware libre de esta plataforma, trata de los dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. Esto quiere decir que Arduino ofrece las bases para que cualquier otra persona o empresa pueda crear sus propias placas, pudiendo ser diferentes entre ellas, pero igualmente funcionales al partir de la misma base (Ver ilustración 5). Sin embargo, en cuanto al hardware, la plataforma Arduino contiene especificaciones muy simples, lo que las hace perfectas para aplicaciones sencillas y de bajo consumo; en ningún caso son adecuadas para el ámbito industrial.

El software libre son los programas informáticos cuyo código es accesible por cualquiera para que quien quiera pueda utilizarlo y modificarlo. Arduino ofrece la plataforma Arduino IDE (Entorno de Desarrollo Integrado), que es un entorno de programación con el que cualquiera puede crear aplicaciones para las placas Arduino, de manera que se les puede dar todo tipo de utilidades.

Además, las shields complementan la funcionalidad del modelo de placa empleada, agregando circuitos electrónicos, sensores y módulos de comunicación externos a la placa original.



Ilustración 5. Arduino UNO

5.3 SIMATIC IOT2040

IOT2040 está desarrollado con una carcasa industrial que permite una instalación segura dentro de maquinarias y paneles eléctricos y una fácil conexión con otros dispositivos industriales como PLCs, fuentes de alimentación, relés o unidades inversoras (ver Ilustración 6).

Con respecto a los protocolos de comunicación, IOT2040 puede comunicarse con otros hardwares industriales mediante el uso de protocolos industriales seguros como Modbus, MQTT, etc.

Las placas de desarrollo de código abierto típicas, generalmente no están desarrolladas para un funcionamiento continuo en un entorno industrial y no están diseñadas para conectarse fácilmente con otros dispositivos industriales.

Las características hardware del dispositivo son las siguientes:

- **Procesador:** Inter Quark x1020
- **Memoria RAM:** 1024MB
- **Interfaces Ethernet:** 2 x RJ
- **Puertos serie:** 2 x RS232/485
- **Reloj Interno:** Mediante Pila

Con un módulo de entrada / salida compatible con Arduino específicamente diseñado para IOT2000, es posible agregar entradas y salidas digitales y análogas al IOT2040. Esto permite conectar directamente los sensores locales a la puerta de enlace de IOT2040 y desencadenar procesos no críticos. La placa está diseñada para el funcionamiento industrial.



Ilustración 6. SIMATIC IOT2040

6 Selección de la solución propuesta

En este apartado se valorará cual de las tres opciones planeadas con anterioridad son las más adecuadas para nuestro proyecto, además de los dispositivos y programas que serán necesarios para su elaboración.

En primer lugar, se comenzará descartando el sistema de automatización Arduino, ya que no contiene memoria RAM, ni capacidad de procesamiento, ni es compatible con diferentes sistemas operativos, es incompatible con un largo número de protocolos y no tiene capacidad alguna de información. Además, las plataformas Arduino no están diseñadas para el ámbito industrial lo que limita su potencial para el proyecto que se está realizando.

Por lo tanto, las únicas opciones posibles a considerar parecen ser los dispositivos SIMATIC IOT2040 y Raspberry PI, ya que sus características son similares. Las únicas diferencias apreciables son que el tamaño y peso de la Raspberry PI es menor que SIMATIC IOT2040, pero esta última plataforma dispone de pines analógicas de entrada y salida que la Raspberry PI no tiene. Sin embargo, la principal diferencia entre ambos dispositivos es que la plataforma SIMATIC IOT2040 está especialmente diseñada para el ámbito industrial, mientras que Raspberri PI no lo está. Por tanto, se optará por la utilización del dispositivo de SIEMENS IOT2040.

Por otro lado, para poder llevar a cabo este proyecto se necesitarán los siguientes dispositivos/accesorios y programas:

- **Hardware:**
 - Siemens SIMATIC IOT2040
 - Tarjeta Micro SD
 - Ordenador con puerto Ethernet y con lector de tarjetas Micro SD
 - 2 cables Ethernet RJ-45
 - HUB de CISCO con 8 puertos
 - PLC SIMATIC S7-1500
- **Software:**
 - PuTTY
 - Win32 Disk Imager
 - Tia Portal V15.1

6.1 PLC

PLC SIMATIC S7-1516

Un controlador lógico programable, más conocido como PLC por sus siglas en inglés, es un sistema basado en μ P muy utilizada en el ámbito de la automatización industrial. Los PLC son capaces de soportar condiciones ambientales adversas, como pueden ser altas temperaturas, vibraciones, polvo, ruido eléctrico, etc. Además, disponen de un número elevado de entradas y salidas, lo que les permite controlar las muchas señales que pueda haber en un proceso industrial.

Los PLCs han evolucionado hasta ser capaces de incluir el control de movimiento, control de procesos, sistemas de control distribuido y comunicación por red. Las capacidades de manipulación, almacenamiento, potencia de procesamiento y de comunicación de algunos PLCs modernos son aproximadamente equivalentes a los ordenadores de sobremesa.

A nivel de software, estos autómatas programables tienen un sistema operativo muy estable. Esto unido a que su hardware está diseñado para trabajar en plantas industriales, hace que los PLCs tengan una fiabilidad muy alta.

El PLC utilizado para este trabajo es el controlador SIMATIC S7-1516 (ver Ilustración 7). A día de hoy, S7-1516 es el autómata de más altas prestaciones que ofrece la empresa SIEMENS. S7-1516 ofrece un mayor rendimiento y mayor facilidad de uso que autómatas anteriores de la misma gama. Las principales ventajas de S7-1516 en contraposición con versiones anteriores son:

- Excelente rendimiento del sistema para tiempos de respuesta más cortos y mayor calidad de control.
- Tecnología integrada para una perfecta integración de autómatas a través de funcionalidades de control.
- Seguridad integrada, para la protección de proyectos de gran inversión.
- Diseño innovador y fácil manejo para uso simple y una operación de puesta en marcha segura.



Ilustración 7. PLC SIMATIC S7-1516

TIA Portal V15.1

Para la programación del PLC SIMATIC S7-1516 se optará por utilizar la herramienta de ingeniería TIA Portal (Totally Integrated Automation) de SIEMENS. TIA Portal es el innovador sistema de ingeniería que permite configurar de forma intuitiva y eficiente todos los procesos de planificación y producción. Convence por su funcionalidad probada y por ofrecer un entorno de ingeniería unificado para todas las tareas de control, visualización y accionamiento.

La principal utilidad que TIA Portal ofrece es la posibilidad de integrar distintas aplicaciones de software industrial para procesos de producción en una misma plataforma lo que facilita enormemente el aprendizaje, la interconexión y la operación. Con esta nueva arquitectura de software tanto los usuarios nuevos como los expertos trabajan de una forma efectiva ya que no necesitan operar una amplia variedad de sistemas de diferentes orígenes. TIA Portal incorpora las últimas versiones de software de ingeniería SIMATIC STEP 7 para la planificación, programación y diagnóstico de todos los controladores SIMATIC, pantallas de visualización y accionamientos SINAMICS de última generación.

En este trabajo, se ha dispuesto de la versión v15.1 (ver Ilustración 8). Se trata de una aplicación modular a la que se le pueden ir añadiendo nuevas funcionalidades según las necesidades concretas de cada sector industrial.

Esta última versión de SIEMENS se centra principalmente en tres aspectos fundamentales:

- Las aplicaciones han sido mejoradas con una programación de lenguaje de alto nivel.
- En el ámbito de la digitalización se ha ampliado la cartera con funciones mejoradas para funcionalidades OPC UA.
- Mayor eficiencia de la ingeniería con el trabajo en equipo y el diagnóstico ampliado a máquina y sistemas.

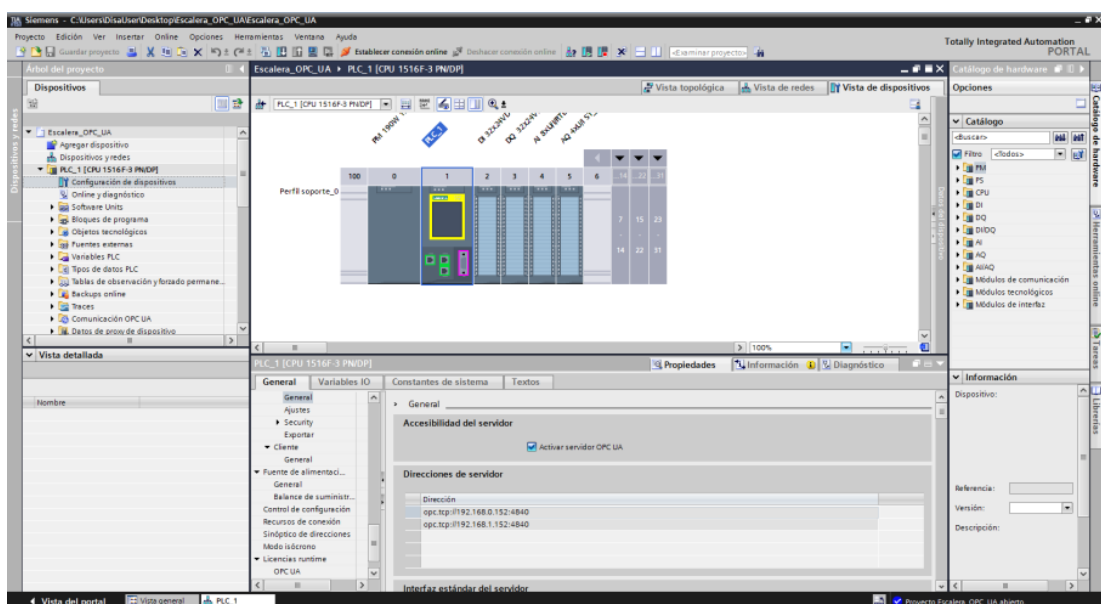


Ilustración 8. TIA Portal V15.1

6.2 IOT2040

Node-RED

Para la comunicación y transferencia de datos a través del SIMATIC IOT2040 se utilizará la herramienta de programación visual Node-RED. Este es un editor de flujo muy ligero, donde se puede añadir o eliminar nodos y conectarlos entre sí, con el fin de establecer una comunicación entre ellos.

El editor de flujos de Node-RED consiste en una sencilla interfaz en HTML, accesible desde cualquier navegador, en la que arrastrando y conectando nodos entre sí, es posible definir un flujo que ofrezca un servicio.

Node-RED es una herramienta de código abierto, estando esté disponible en Github (<https://github.com/node-red/node-red>). Una de las características más notables de Node-RED es la sencillez con la que se pueden crear nuevos nodos e instalarlos, en el siguiente enlace disponemos de toda la documentación necesaria: <http://nodered.org/docs/creating-nodes/>. De este modo, cualquier persona u organización puede crearse sus propios nodos, adaptando el motor de flujo a las necesidades de su negocio.

Los flujos programados en Node-RED se almacenan internamente en formato de código (JSON) y son portables entre distintas instalaciones de Node-RED, siempre que el Node-RED de destino tenga instalados los nodos utilizados en el flujo.

Los nodos pueden clasificarse en tres grupos diferentes:

- Nodos de entrada: Únicamente funcionales con datos de entrada.
- Nodos de salida. Solo admiten datos recibidos mediante diferentes métodos.
- Nodos de entrada y salida: ofrecen la entrada de datos y una o varias salidas.

Al acceder Node-RED se puede comprobar que se trata de una programación visual por bloques, formada por tres ventanas principales (ver Ilustración 9): ventana izquierda de selección de nodos, ventana central de programación de los flujos, y ventana derecha de información, obtención de datos y configuración, entre otras:

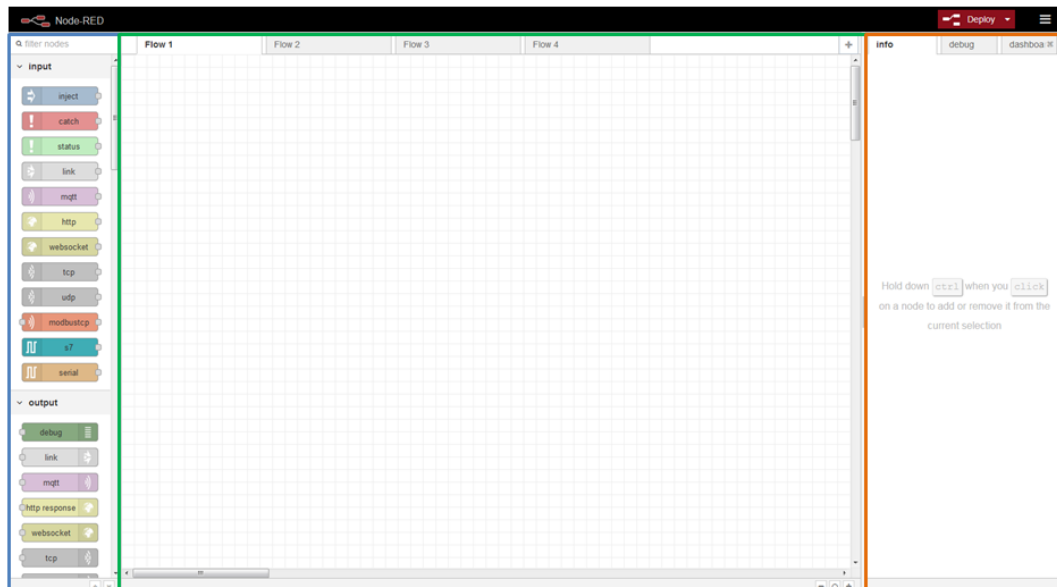


Ilustración 9. Aplicación Node-RED

La parte marcada en azul corresponde a la lista de selección de nodos, donde aparecen todos los nodos de entrada, salida y programación, entre otros, instalados en Node-RED.

La ventana verde es el espacio de programación, donde se arrastran los nodos y se vinculan entre sí para formar flujos y así crear la programación.

La última, de color naranja, muestra información de los nodos, mensajes creados de forma local para el programa Node-Red y configuración de los flujos y nodos.

PuTTY

PuTTY, es un cliente de acceso remoto a máquinas informáticas de cualquier tipo mediante SSH, Telnet o RLogin, para plataformas Windows 32bits y UNIX. Esta herramienta permite el acceso remoto a otros sistemas compatibles o no con el formato que estamos ejecutando.

Su nombre proviene de las siglas PU: Port Unique y TTY: Terminal Type, es decir, puerto único de tipo terminal (ver Ilustración 10).



Ilustración 10. Simbolo PuTTY

Usando esta herramienta es posible acceder a un dispositivo remoto con tan solo conocer su IP y estando conectados mediante un cable Ethernet. Por lo tanto, esta herramienta será de gran ayuda para poder acceder a Node-RED mediante IOT2040.

7 Metodología. Diseño

7.1 Introducción

En este apartado se detallarán los pasos a seguir para la validación de diferentes comunicaciones entre el equipo de adquisición de datos el IOT2040 y un equipo de control como el PLC SIMATIC S7 1516, utilizando protocolos de comunicaciones de uso común en entornos industriales.

En primer lugar, se hará una breve descripción del programa de control en el PLC, que genera los datos que se van a manipular. A continuación, se hará una detallada descripción de las pruebas y procedimientos realizados en cada una de las arquitecturas utilizadas. La aplicación de control en el PLC es común, el control de una escalera mecánica. Además, se deberá poner en servicio el IOT2040 con todas las configuraciones necesarias para poder poner en marcha el dispositivo de captura de datos. Una vez configurado el PLC, se procederá a trabajar con la plataforma IOT2040 para así adquirir datos con los diferentes protocolos de comunicaciones industriales.

7.1.1 Programa de control en el PLC

El programa utilizado para el PLC consiste en el control en tres escaleras mecánicas como la que se muestra en la ilustración 11. En esta ilustración, se muestran los dispositivos utilizados en el control de la misma: un interruptor de puesta en servicio, un sensor de temperatura para detectar la temperatura de los bobinados de del motor, y una célula fotoeléctrica a la entrada de la escalera para detectar el paso de personas. Además, hay dos pulsadores de emergencia para detener la instalación.

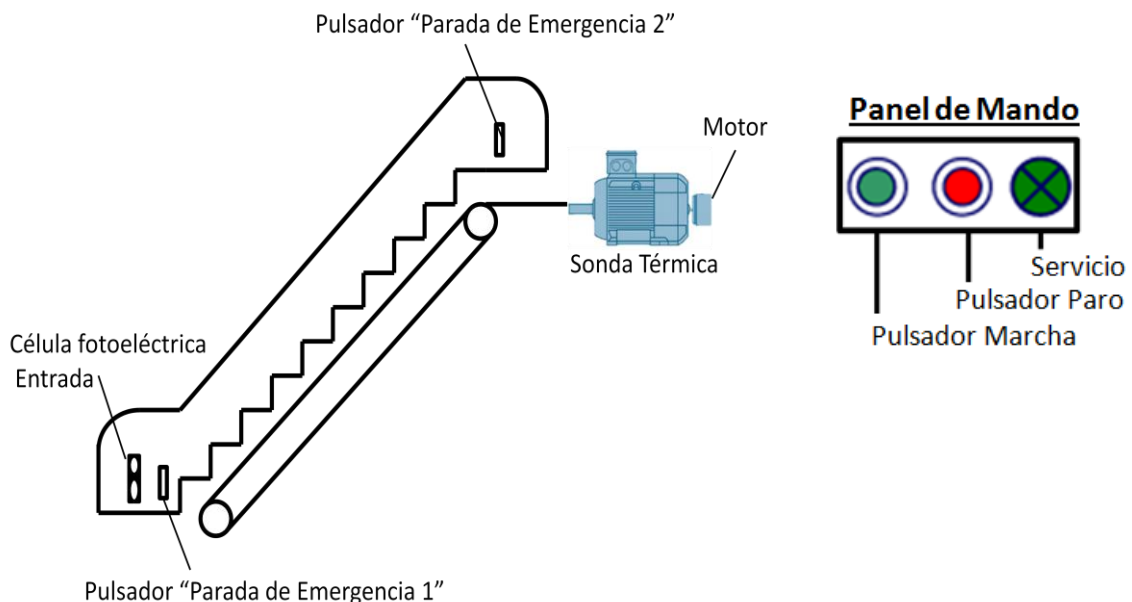


Ilustración 11. Escalera mecánica

En la Tabla1, se muestran cada una de las variables que participan en el programa, la dirección asignada a cada una de ellas y el tipo de dato del que se trata:

Nombre	Tipo de dato	Dirección
PM_E1	Bool	%I0.0
PP_E1	Bool	%I0.1
CF_E1	Bool	%I0.2
ST_E1	Bool	%I0.3
PE_E1	Bool	%I0.4
T1_E1	Timer	%T1
T2_E1	Timer	%T2
Servicio_E1	Bool	%Q0.0
Motor_E1	Bool	%Q0.1
PM_E2	Bool	%I1.0
PP_E2	Bool	%I1.1
CF_E2	Bool	%I1.2
ST_E2	Bool	%I1.3
PE_E2	Bool	%I1.4
T1_E2	Timer	%T10
T2_E2	Timer	%T11
Servicio_E2	Bool	%Q1.0
Motor_E2	Bool	%Q1.1
PM_E3	Bool	%I2.0
PP_E3	Bool	%I2.1
CF_E3	Bool	%I2.2
ST_E3	Bool	%I2.3
PE_E3	Bool	%I2.4
T1_E3	Timer	%T20
T2_E3	Timer	%T21
Servicio_E3	Bool	%Q2.0
Motor_E3	Bool	%Q2.1

Tabla 1. Variables programa PLC

El programa de control está estructurado como se muestra en la Ilustración 12:

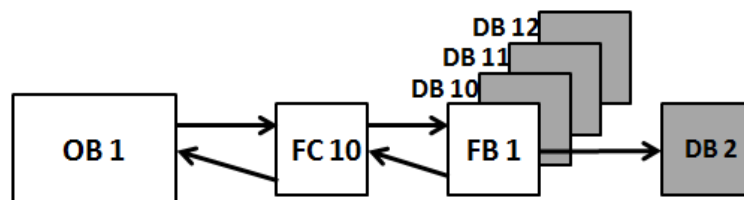


Ilustración 12. Estructura Programa

- La función (FC 10) mantendrá una copia de los parámetros. En la Tabla 2 que se muestra a continuación, se han especificado las variables del FC 10 que serán necesarias para la primera de las escaleras mecánicas, ídem las otras dos.

Variables FB	Variables DB	Dirección
P_Marcha	"PM_E1"	%I 0.0
P_Paro	"PP_E1"	%I 0.1
C_Entrada	"CF_E1"	%I 0.2
S_Termica	"ST_E1"	%I 0.3
P_Emergencia	"PE_E1"	%I 0.4
Temporizador1	"T1_E1"	%T1
Temporizador2	"T2_E1"	%T2
Servicio	"Servicio_E1"	%Q 0.0
Motor	"Motor_E1"	%Q 0.1
N_Personas	"DB20".Personas_E1	%DB20.DBW0
T_Funcionamiento	"DB20".Tiempo_E1	%DB20.DBW2
P_Marcha	"PM_E2"	%I 1.0
P_Paro	"PP_E2"	%I 1.1
C_Entrada	"CF_E2"	%I 1.2
S_Térmica	"ST_E2"	%I 1.3
P_Emergencia	"PE_E2"	%I 1.4
Temporizador1	"T1_E2"	%T10
Temporizador2	"T2_E2"	%T11
Servicio	"Servicio_E2"	%Q 1.0
Motor	"Motor_E2"	%Q1.1
N_Personas	"DB20".Personas_E2	%DB20.DBW4
T_Funcionamiento	"DB20".Tiempo_E2	%DB20.DBW6
P_Marcha	"PM_E3"	%I.0.0
P_Paro	"PP_E3"	%I.0.1
C_Entrada	"CF_E3"	%I.0.2
S_Termica	"ST_E3"	%I.0.3
P_Emergencia	"PE_E3"	%I.0.4
Temporizador1	"T1_E3"	%T20
Temporizador2	"T2_E3"	%T21
Servicio	"Servicio_E3"	%Q2.0
Motor	"Motor_E3"	%Q2.1
N_Personas	"DB20".Personas_E3	%DB20.DBW8
T_Funcionamiento	"DB20".Tiempo_E3	%DB20.DBW10

Tabla 2. Función 10.

- **P_Marcha** : Pulsador que pone la instalación en servicio
- **P_Paro**: Pulsador que deja la instalación fuera de servicio.
- **C_Entrada**: Variable que indica la entrada de un nuevo usuario a la escalera.

- **S_Térmica:** Medida de seguridad que hace desconectar la escalera en caso de ocurrir un fallo.
 - **P_Emergencia:** Pulsador que permite al usuario parar la escalera con efecto inmediato
 - **Temporizador1:** Temporizador de 5 segundos.
 - **Temporizador2:** Temporizador de 1 segundo.
 - **Servicio:** Indica que la instalación se encuentra operativa.
 - **Motor:** Muestra su la escalera está en funcionamiento.
 - **N_Personas:** Contador para las personas que han hecho uso de la escalera.
 - **T_Funcionamiento:** Muestra el tiempo que ha estado la escalera en funcionamiento.
- La programación se lleva a cabo a través de un bloque de funciones (FB 1). Dentro del mismo se distinguen dos partes: La interfaz de bloque (Tabla 3), y el programa (contiene el programa que será ejecutado y desarrollado en AWL). El funcionamiento de la escalera mecánica es el siguiente:

Una vez se haya activado el pulsador marcha y los pulsadores paro y emergencia estén abiertos además del relé térmico (ya que están normalmente cerrados), la escalera mecánica se pondrá en servicio. Mientras que la escalera esté en servicio, se indicará en el panel de control, ilustrado en la figura 11. El PLC que controla la escalera mecánica dispondrá de 2 contadores por escalera mecánica, que llevarán el registro del tiempo de funcionamiento en segundos y del número de personas que acceden a la misma.

El paro de la escalera se puede realizar de dos formas diferentes:

- Tras una conexión del pulsador paro (normalmente cerrado). Su efecto no es inmediato, tarda 5 segundos en desconectarse, el tiempo que necesita una persona en ir desde el comienzo de la escalera hasta su final.
- Tras una conexión del Relé Térmico (normalmente cerrado) o mediante una conexión del pulsador de emergencia (normalmente cerrado).

Todas las variables se instancian en los DBs (DB10, DB11, DB12), en funciona de la escalera mecánica que se está usando.

Nombre	Tipo de dato
Input	
P_Marcha	Bool
P_Paro	Bool
C_Entrada	Bool
S_Termica	Bool
P_Emergencia	Bool
Temporizador	Timer
Temporizador2	Timer
Output	
Servicio	Bool
Motor	Bool
N_Personas	Word

T_Funcionamiento	Word
Input	
Static	
C_Personas	Word
C_Funcionamiento	Word
M1	Bool
M2	Bool
M3	Bool
M4	Bool
M5	Bool
M6	Bool

Tabla 3 Interfaz de bloque

Se ha tenido la necesidad de crear un DB para poder acceder sin ninguna dificultad a las variables tiempo de funcionamiento de las escaleras mecánicas (*T_Funcionamiento*) y a las personas que acceden (*N_Personas*) a las mismas desde Node-RED.

Nombre	Tipo de dato	Offset
Static		
Personas_E1	Word	0.0
Tiempo_E1	Word	2.0
Personas_E2	Word	4.0
Tiempo_E2	Word	6.0
Personas_E3	Word	8.0
Tiempo_E3	Word	10.0

Tabla 4. Variables en el DB2

- Un OB1. Bloque de organización para la ejecución cíclica del programa. Este bloque permite realizar la llamada al bloque de funciones (FC10) (ver ilustración 13).

1	CALL "Bloque_1"	FC10
2		

Ilustración 13. Bloque de organización del programa.

Para poder realizar el servicio de escritura a través de OPC UA, se añadirán dos nuevas variables al DB2 (Booleana1 y Byte1), como se observa en la ilustración 14. En caso de no añadir variables que no son utilizadas por el programa del PLC, estas se solaparían y los valores que se estuvieran insertando desde Node-RED no podrían ser leídos. Así, la configuración del DB2 quedaría de la siguiente forma:

DB2									
	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Escrib...	Visible en ..	
1	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	Personas_E1	Word	...	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Tiempo_E1	Word	...	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Personas_E2	Word	...	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Tiempo_E2	Word	...	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Personas_E3	Word	...	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Tiempo_E3	Word	...	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Booleana1	Bool	...	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Byte1	Byte	...	16#0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Ilustración 14. Configuración DB2 para servicio de escritura en OPC UA

7.1.2 Metodología de arranque

En los siguientes apartados se detallará la metodología a seguir para la puesta en marcha de la plataforma IOT2040. En primer lugar, se procederá a la grabación de la imagen de arranque en una tarjeta de memoria Micro SD, para posteriormente descargar PuTTY, configurar la IP del dispositivo e iniciar sesión en el IOT2040.

7.1.2.1 Grabación de imagen del SIMATIC IOT2040

Para la puesta en marcha del dispositivo tendremos que seguir 4 pasos:

1. Descarga de la imagen de arranque Linux para la carga del sistema operativo del SIMATIC IOT2040. Su descarga se realizará desde la siguiente dirección web:
<https://support.industry.siemens.com/cs/document/109741799/imagen-ejemplo-para-la-sd-card-de-un-simatic-iot2000?dti=0&lc=es-ES>
2. Una vez se tiene la imagen descargada en el PC, hay que instalarla en la tarjeta de memoria para después introducirla en el IOT2040. Para poder instalar la imagen se necesita el programa Win32DiskImager, disponible para descargar en el enlace.
<https://sourceforge.net/projects/win32diskimager/>
3. Se grabará la imagen de arranque en una tarjeta Micro SD, procediendo así a la configuración del dispositivo para poder alojar los programas utilizados posteriormente. La configuración tratará de seleccionar el archivo .wic, la letra de la unidad en la que está la tarjeta Micro SD y hacer clic en *Write* (ver Ilustración 15).

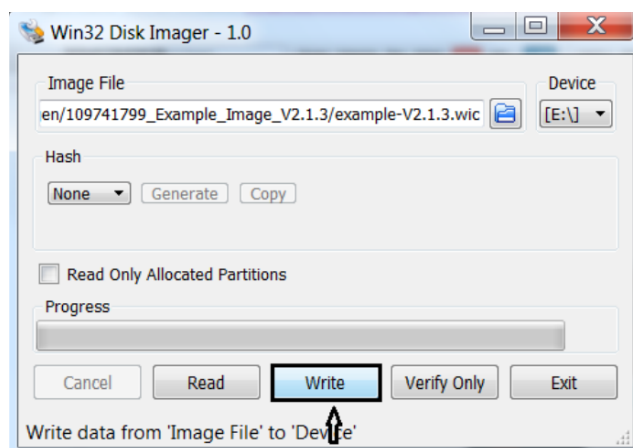


Ilustración 15. Grabación imagen de arranque

4. Inserción de la tarjeta Micro SD en el IOT2040 (ver Ilustración 16).

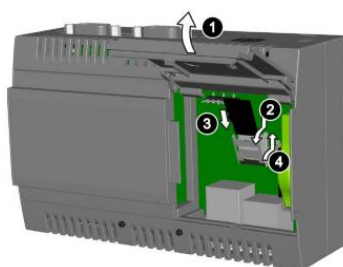


Ilustración 16. Inserción tarjeta de memoria Micro SD en IOT2040

7.1.2.2 Descarga de PuTTY, Configuración IP del dispositivo e inicio de sesión en SIMATIC IOT2040.

La comunicación se llevará a cabo a través del puerto X1 del IOT2040. Para establecer conexión entre el dispositivo SIMATIC IOT2040 y el PC se utilizará la herramienta de programación PuTTY. Su descarga se realizará desde la siguiente dirección web, <http://www.putty.org>.

A continuación, se procederá a la configuración para las comunicaciones con el ordenador y el almacenamiento de los programas, tal y como se muestra en la ilustración 17:

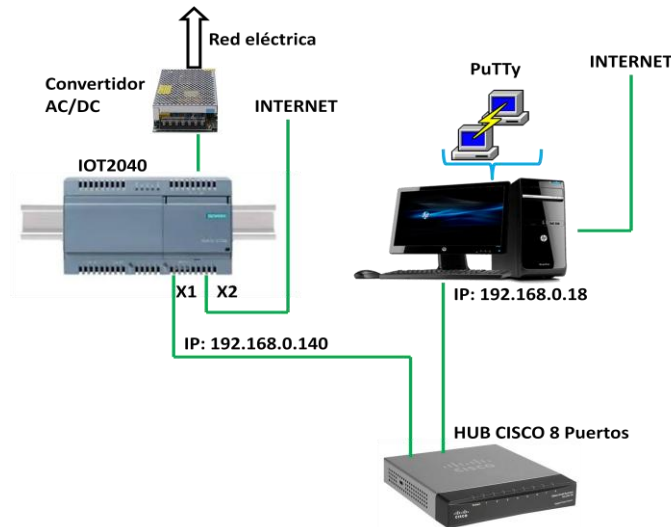


Ilustración 17. Demostrador para inicio de sesión en SIMATIC IOT2040

1. Se conecta el IOT2040 a una fuente de alimentación y a un ordenador a través de un cable Ethernet.
2. Se ejecuta en el PC el programa PuTTY descargado. Para ello, el dispositivo IOT2040 deberá estar conectado con el cable Ethernet al PC, pudiendo así introducir la dirección IP (ver Ilustración 18). En este caso, la dirección IP del dispositivo es 192.168.200.1, ya que es la que tiene asignada por defecto. Una vez introducida la dirección IP, se hace clic en Open, donde se abrirá un terminal SSH (Secure *SHell*) que permite establecer la conexión con el terminal Linux del IOT2040.

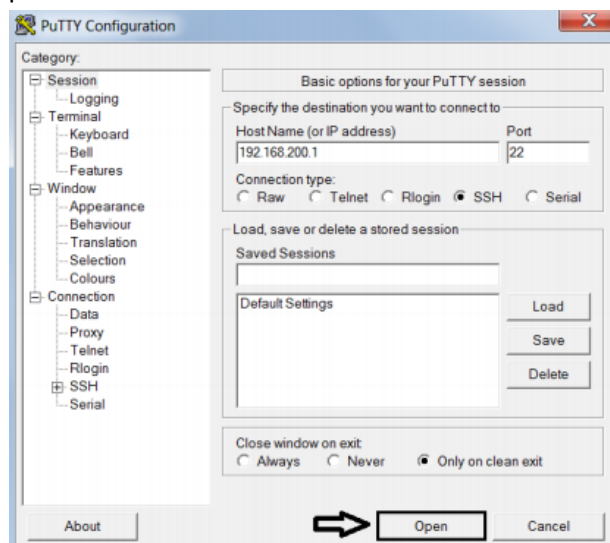


Ilustración 18. Configuración por SSH al dispositivo.

3. En dicha interfaz, se introduce el nombre de usuario *root* para iniciar sesión. El dispositivo no tiene configurada ninguna contraseña de acceso por lo que directamente se podría configurar el terminal de Linux del dispositivo. Para configuraciones adicionales tales como usuario/contraseña, interfaces, direcciones IP, etc. Se realizarán, a través del comando *iot2000setup* que es un menú que permite realizar estos ajustes.
4. Se procederá a cambiar la IP 192.168.200.1 del dispositivo IOT2040 por la dirección IP 192.168.0.140 (por defecto, está asignado el protocolo dhcp para el puerto Ethernet 2). Para ello, introducir el comando 'iot2000setup', se accede a "Networking"-> "Configure Interfaces" y seguir las instrucciones que se indican (ver Ilustración 19):

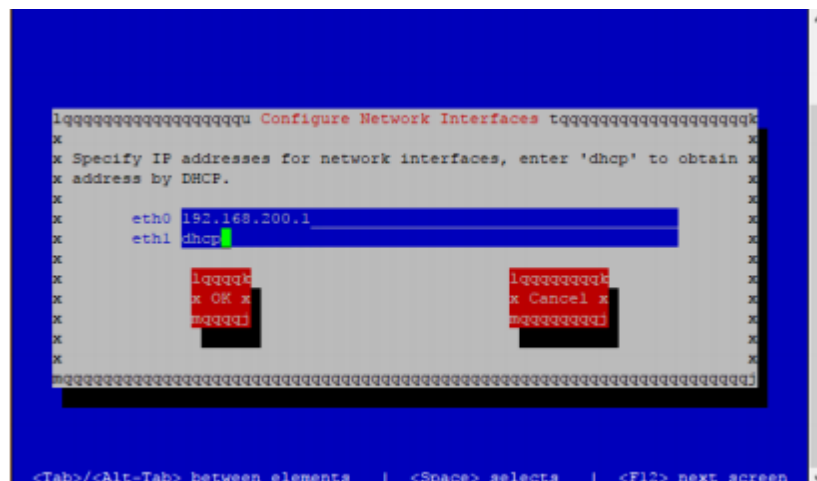


Ilustración 19. Cambio de dirección IP por defecto mediante PuTTY

5. Se volverá a ejecutar el programa PuTTY, pero esta vez con la nueva dirección IP (ver Ilustración 20).

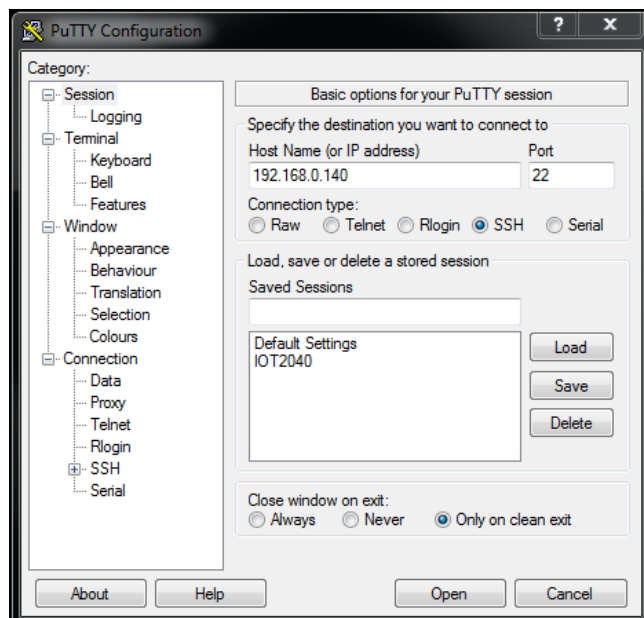
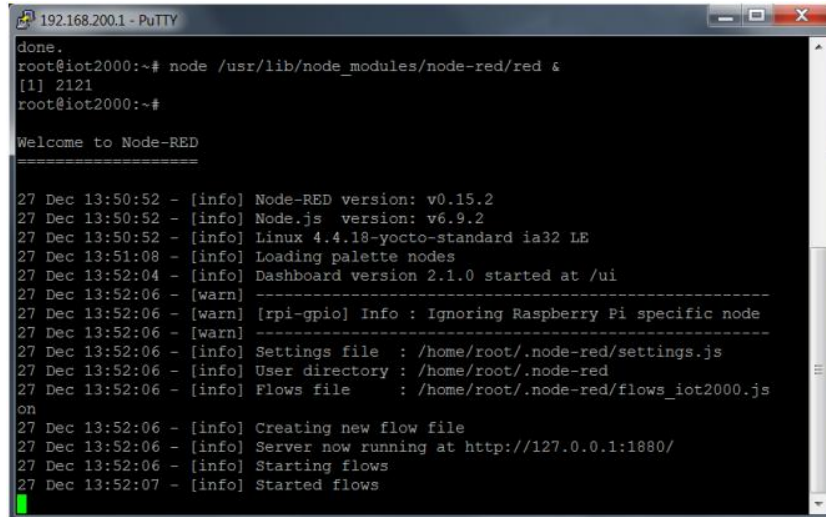


Ilustración 20. Conexión a IOT2040 con IP modificada

6. Para poder acceder al entorno de programación de Node-RED se tiene que escribir el siguiente comando a través de PuTTY: `node /usr/lib/node_modules/node-red/red &`.
7. Una vez Node-RED se ha iniciado, tal como se muestra en la ilustración 21, abrimos un explorador y navegamos a la dirección web de la dirección IP del dispositivo IOT añadiéndole el puerto 1080 (o la dirección IP asignada al puerto Ethernet que se tiene conectada), (ver Ilustración 22).



```
192.168.200.1 - PuTTY
done.
root@iot2000:~# node /usr/lib/node_modules/node-red/red &
[1] 2121
root@iot2000:~#

Welcome to Node-RED
=====

27 Dec 13:50:52 - [info] Node-RED version: v0.15.2
27 Dec 13:50:52 - [info] Node.js version: v6.9.2
27 Dec 13:50:52 - [info] Linux 4.4.18-yocto-standard ia32 LE
27 Dec 13:51:08 - [info] Loading palette nodes
27 Dec 13:52:04 - [info] Dashboard version 2.1.0 started at /ui
27 Dec 13:52:06 - [warn] -----
27 Dec 13:52:06 - [warn] [rpi-gpio] Info : Ignoring Raspberry Pi specific node
27 Dec 13:52:06 - [warn] -----
27 Dec 13:52:06 - [info] Settings file   : /home/root/.node-red/settings.js
27 Dec 13:52:06 - [info] User directory  : /home/root/.node-red
27 Dec 13:52:06 - [info] Flows file      : /home/root/.node-red/flows_iot2000.js
on
27 Dec 13:52:06 - [info] Creating new flow file
27 Dec 13:52:06 - [info] Server now running at http://127.0.0.1:1880/
27 Dec 13:52:06 - [info] Starting flows
27 Dec 13:52:07 - [info] Started flows
```

Ilustración 21. Paleta de nodos cargada a través de PuTTY



Ilustración 22. Acceso a Node-RED a través del navegador

Una vez realizado el arranque del dispositivo IOT2040 y habiendo descrito el programa utilizado en el PLC consistente en el control de tres escaleras mecánicas, se procederá a adquirir datos con los diferentes protocolos de comunicación industriales: Modbus/TCP y OPC UA.

7.2 Protocolo de comunicación Modbus/TCP

Modbus/TCP es un protocolo de comunicación de nivel 7 del modelo OSI, que proporciona comunicación cliente/servidor entre diferentes dispositivos conectados a Ethernet. En la actualidad, el protocolo Modbus es muy utilizado en entornos industriales, ya que es un protocolo que permite a los dispositivos comunicarse utilizando un modelo de comunicación (cliente-servidor), en la que un dispositivo puede iniciar transacciones (consultas). Los otros dispositivos (servidor) responden suministrando los datos solicitados por el cliente, o tomando la acción solicitando en la consulta.

- El *cliente* puede ser una interfaz humano-máquina (HMI) o sistema SCADA
- Un *servidor* es cualquier dispositivo periférico (transductor de E/S, válvula, unidad de red u otro dispositivo de medición) que procesa información y envía su salida al cliente.

Modbus se utiliza para la conexión de un ordenador con una unidad remota (RTU), en sistemas de supervisión de adquisición de datos (SCADA). Además, existen versiones del protocolo Modbus para puertos series y Ethernet.

En su origen este protocolo estaba diseñado para ser conectados a través de líneas serie como RS-232 o RS-485. De este protocolo surgieron varias versiones que permitieran adaptarse a las nuevas formas de comunicación que iban surgiendo. Así fue como se originó el protocolo conocido como Modbus/TCP, que mantenía la forma de enviar información de Modbus en un entorno en el que se emplean protocolos TCP/IP.

Por lo tanto, la especificación Modbus/TCP define un estándar interoperable en el campo de la automatización industrial, que es simple de implementar para cualquier dispositivo que soporte TCP/IP. Todas las solicitudes son enviadas vía TCP sobre el puerto registrado 502. La combinación de una red física versátil y escalable como Ethernet con el estándar universal de interredes TCP/IP y una representación de datos independiente del fabricante como MODBUS, proporciona una red abierta para el intercambio de datos de cualquier proceso.

La comunicación mediante Modbus/TCP siempre se produce en pares, un dispositivo debe iniciar una solicitud y luego esperar una respuesta. Por lo tanto, la comunicación Cliente/Servidor empleando Modbus/TCP está basada en cuatro tipos de mensajes (ver Ilustración 23).

- **Petición (Request):** Mensaje enviado por el cliente para iniciar la comunicación.
- **Indicación (Indication):** Mensaje de petición recibido por el servidor.
- **Respuesta (Response):** Respuesta enviada por el servidor.
- **Confirmación (Confirmation):** Mensaje de respuesta recibido por el cliente.



Ilustración 23. Comunicación cliente/servidor en Modbus/TCP

Modbus/TCP básicamente encapsula una trama MODBUS dentro de una trama TCP en una manera simple, como se muestra en la Ilustración 24:

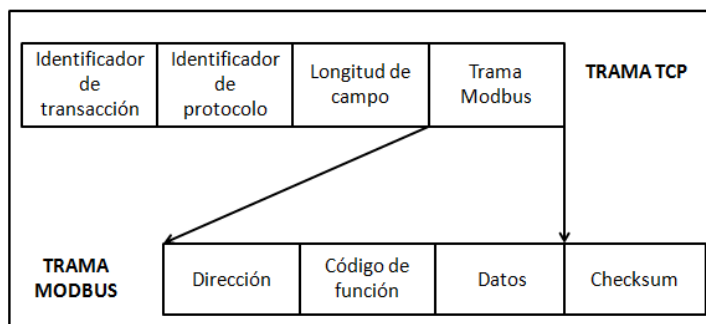


Ilustración 24. Encapsulamiento de la trama Modbus/TCP

De esta forma, un mensaje Modbus/TCP completo posee la siguiente estructura:

- **Identificador de transacción** (*Bytes 0 y 1*): Se trata de un número que no tiene valor en el servidor, pero es copiado literalmente desde la solicitud a la respuesta a conveniencia del cliente. Este campo se utiliza para que un cliente Modbus/TCP pueda establecer simultáneamente múltiples conexiones con diferentes servidores y pueda identificar cada una de las transacciones.
- **Identificador de protocolo** (*Byte 2 y 3*): Es un número que debe ser establecido a cero.
- **Longitud de campo** (*Byte 4 (Byte bajo) y 5 (Byte alto)*): Especifica el número de bytes que siguen. La longitud es una cantidad de dos bytes, pero el byte alto tiene el valor de cero ya que los mensajes son más pequeños de 256 bits.

Dentro de la trama Modbus se distinguen 3 zonas diferentes, precisamente estas serán las que posteriormente se podrán configurar a través de Node-RED:

- **Dirección** (*Byte 6*): La identificación de los equipos se realiza en el campo de dirección que dispone de 8 bits, lo que nos da un conjunto de 256 valores distintos para identificar el hardware involucrado en la comunicación. Se deben tener direcciones unívocas entre los equipos, es decir, no pueden existir dos direcciones iguales entre equipos.
- **Código de función** (*Byte 7*): Los registros Modbus de un dispositivo están organizados en torno a los cuatro datos básicos tipos de referencia mencionados anteriormente. Además, este tipo de datos se identifica con más detalle por el número inicial de referencia de la siguiente manera:

Descripción	Lectura	Escritura simple	Escritura múltiple
Salidas discretas	FC 01	FC 05	FC 15
Entradas discretas	FC 02	-	-
Registros de entrada analógica	FC 04	-	-
Registros de retención de salida analógica	FC 03	FC 06	FC 16

Tabla 5. Funciones en Modbus/TCP

Las funciones que aparecen en la *Tabla 5* anterior corresponden con las funciones que se nos van a permitir configurar en Node-RED.

- **Datos** (*Byte 8 o más*): Son los datos que se han de transportar.

7.2.1 Demostrador

En la ilustración 25 se presentan los dispositivos que serán utilizados en el demostrador para poder probar y validar la comunicación entre el IOT2040 y el analizador de redes eléctrica que medirá las magnitudes eléctricas de las tres diferentes fases eléctricas que alimentan los motores eléctricos encargados de mover las escaleras mecánicas, utilizando el protocolo Modbus /TCP. Como se puede observar en la ilustración 25, se ha dispuesto de un equipo IOT2040, un analizador de redes, un HUB de Cisco de 8 puertos y el PC.

El dispositivo IOT2040 tiene el puerto X2 conectado a internet y el Puerto X1 conectado al HUB. El puerto conectado a Internet se ha parametrizado para que se le asigne la dirección IP mediante el protocolo DHCP y al puerto conectado al HUB se la asigna de forma manual la dirección IP, 192.168.0.140.

El analizador de redes tiene conectado el puerto con dirección IP 192.168.0.200 al HUB.

El dispositivo PC, tiene el puerto de la placa base conectado a Internet parametrizado para asignación dinámica de dirección IP. Se le ha instalado una tarjeta de red Ethernet, cuyo puerto se ha conectado al HUB, a la que se le ha asignado de forma manual la dirección IP, 192.168.0.18.

En relación con el software instalado en este equipo indicar los siguientes:

- PuTTY: Aplicación de código abierto que utiliza protocolos de red para habilitar una sesión remota en una computadora.
- Node-RED: Entorno gráfico de desarrollo de aplicaciones que se cargan y ejecutan en el equipo IOT2040.

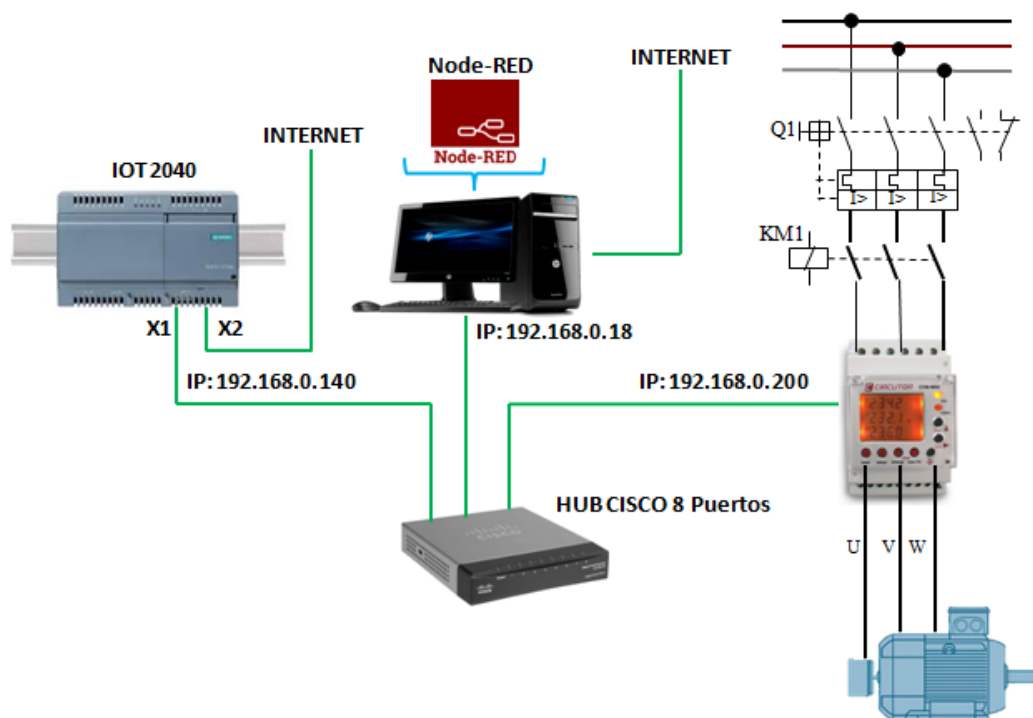


Ilustración 25. Demostrador para Modbus/TCP

7.2.2 Analizador de redes eléctricas

Con este protocolo, se va a realizar el servicio de lectura de un dispositivo denominado analizador de redes y su función es realizar medidas de magnitudes eléctricas. Este tipo de dispositivos son muy utilizados en la actualidad para realizar un seguimiento del consumo eléctrico de una instalación para mejorar la eficiencia energética.

Para poder llevar a cabo esta aplicación, se contará con un analizador de redes tipo CVM-MINI (CIRCUITOR), que es un instrumento de medición configurable. Este mide, analiza y visualiza los principales parámetros eléctricos de redes industriales trifásicas equilibradas o desequilibradas. En el caso a tratar, el analizador será conectado a la red eléctrica que alimentará al conjunto de escaleras mecánicas automatizadas en el apartado 7.2. Por lo tanto, se tratará de una red trifásica equilibrada. En la ilustración 26 se muestra el modelo CVM-MINI y su conexión a la red eléctrica.

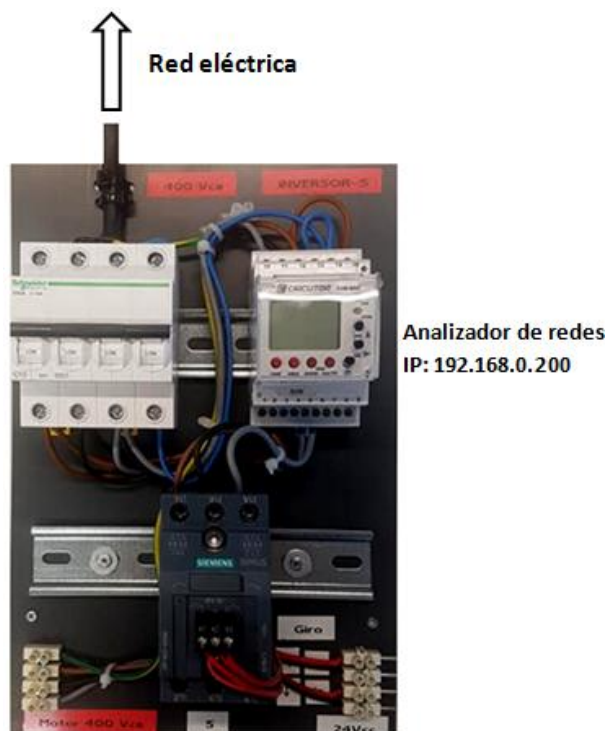


Ilustración 26. Maqueta analizador de redes

El CVM-MINI permite la visualización de todos los parámetros eléctricos (tensión, corriente, potencias y factor de potencia), mediante su display LCD retroiluminado, visualizando tres parámetros eléctricos instantáneos, máximos o mínimos en cada salto de pantalla.

El analizador de redes tipo CVM-MINI se comunica utilizando protocolo Modbus/TCP. Dentro del protocolo MODBUS se utiliza el modo RTU (Remote terminal Unit).

Cuando se alimenta el CVM-MINI mediante la red eléctrica, el equipo inicializa su software interno. Ante una ausencia de programación previa, como es nuestro caso, el analizador mostrará por pantalla la tensión entre fase y neutro de las tres fases (L1, L2 y L3).

En la Tabla 6, se muestran las magnitudes correspondientes a las tres fases del sistema eléctrico (a excepción de la temperatura). En cada una de las fases se distinguen: La tensión fase, la corriente, las potencias activas, reactivas y aparentes y el factor de potencia.

MAGNITUD	SIMBOLO	Instantáneo	Máximo	Mínimo	Uds.
Tensión Fase	V L1	00-01	60-61	C0-C1	V x10
Corriente	A L1	02-03	62-63	C2-C3	mA
Potencia Activa	kW L1	04-05	64-65	C4-C5	w
Potencia Reactiva	Kvar L1	06-07	66-67	C6-C7	w
Potencia Aparente	kV·A L1	4A-4B	AA-AB	10A-10B	
Factor de Potencia	PF L1	08-09	68-69	C8-C9	x 100
Tensión Fase	V L2	0A-0B	6A-6B	CA-CB	V x10
Corriente	A L2	0C-0D	6C-6D	CC-CD	mA
Potencia Activa	kW L2	0E-0F	6E-6F	CE-CF	w
Potencia Reactiva	Kvar L2	10-11	70-71	D0-D1	w
Potencia Aparente	kV·A L2	4C-4D	AC-AD	10C-10D	
Factor de Potencia	PF L2	12-13	72-73	D2-D3	x 100
Tensión Fase	V L3	14-15	74-75	D4-D5	V x10
Corriente	A L3	16-17	76-77	D6-D7	mA
Potencia Activa	kW L3	18-19	78-79	D8-D9	W
Potencia Reactiva	Kvar L3	1A-1B	7A-7B	DA-DB	W
Potencia Aparente	kV·A L3	4E-4F	AE-AF	10E-10F	w
Factor de Potencia	PF L3	1C-1D	7C-7D	DC-DD	x 100
Temperatura	°C	50-51	B0-B1	110-111	°C x 10

Tabla 6. Mapa de memoria CVM-MINI

En la columna *instantáneo* se representa un código de números que sigue el siguiente esquema:

- 8 bit binario, hexadecimal 0-9, A-F
- 2 caracteres hexadecimales contenidos en cada campo de 8-bit del mensaje.

7.2.3 Desarrollo de la aplicación en Node-RED para Modbus/TCP

Para poder empezar a trabajar con este protocolo se necesitará instalar el paquete de nodos “*node-red-contrib-modbustcp*” en la aplicación Node-RED. La manera más fácil de realizarlo es haciendo clic en “*Manage palette*” desde el menú de configuración en la esquina superior derecha del monitor (ver Ilustración 27).

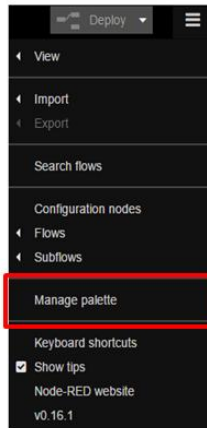


Ilustración 27. Menú de configuración de Node-red

A continuación, se mostrará una barra lateral de administración de nodos en el lado izquierdo del editor. Se hace clic en la pestaña instalar y se procede a buscar “*node-red-contrib-modbustcp*” (ver Ilustración 28).

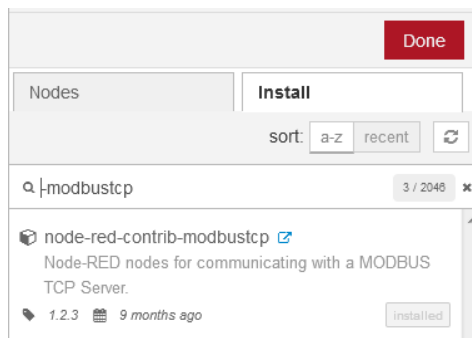


Ilustración 28. Instalación de paquete de nodos modbustcp

Haciendo clic en el botón de instalación se procederá a la instalación del nodo. El nodo input descargado será el responsable de la lectura desde el cliente (IOT2040) hacia el servidor y el nodo output será el responsable de la escritura desde el cliente hacia el servidor (ver Ilustración 29).

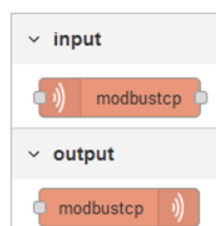


Ilustración 29. Nodos instalados para protocolo modbustcp

7.5.3.1 Servicio de lectura de datos

Para la realización de la lectura de datos desde Node-RED únicamente se tendrán que escribir los valores instantáneos correspondientes a las tensiones fase. Para el servicio de lectura se necesita un nodo input [modbustcp] y un nodo output [debug]. Estos se encontrarán en el lado izquierdo de la aplicación Node-RED (ver Ilustración 30).

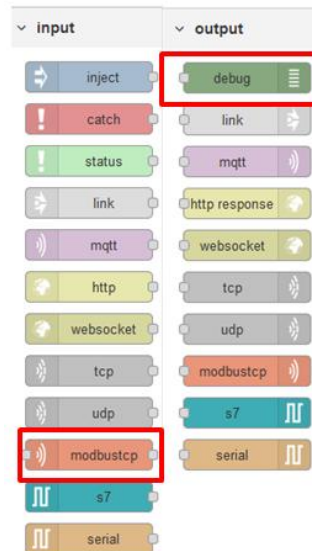


Ilustración 30. Nodos utilizados para el servicio de lectura

Para realizar la lectura, el nodo input [modbustcp] será el encargado de entregar los datos provenientes del dispositivo que se está empleando. La configuración de este nodo consta de los apartados que se muestran en la ilustración 31:

The image shows the 'Edit modbustcp node' configuration window. It has a title bar 'Edit modbustcp node' and buttons for 'Delete', 'Cancel', and 'Done'. The configuration fields are: Name (TENSIÓN FASE 1), Topic (topic), FC (FC 3: Read Holding Registers), Address (0), Quantity (2), Poll Rate (3 second(s)), Server (modbustcp@192.168.0.200:502), IEEE-754 Type (<off>), and Endian (empty).

Ilustración 31. Configuración nodo input modbustcp

Tanto el apartado *Name* como *Topic* sirven para nombrar y categorizar el nodo, a efectos prácticos no tienen ninguna relevancia en la comunicación.

Los apartados *FC*, *Address* y *Quantity*, que serán definidos a continuación, permiten esencialmente definir la trama de Modbus que se utilizan para solicitar el servicio de lectura.

FC hace referencia al tipo de función que realizará el nodo, que han sido explicados detalladamente en la *Tabla 5*:

- FC1: Read Coils
- FC2: Read Discrete Inputs
- FC3: Read Holding Registers
- FC4: Read Input Registers

En este caso, se selecciona la opción FC3 ya que se pretende leer datos del registro del CVM-MINI.

Los siguientes dos apartados, *Address* y *Quantity*, sirven para definir que variable del registro al que se está acudiendo se pretende leer. *Address* define la dirección de la variable a leer en el registro, mientras que *Quantity* es la longitud de esta variable en bytes.

Para acceder a la dirección de la variable que será leída en el registro, se acude a la tabla suministrada por el fabricante que se ha adjuntado anteriormente (*Tabla 6*). Los valores de las fases eléctricas se deberán escribir en formato decimal por lo tanto los valores 0A-0B y 14-15 de la *Tabla 6* tendrán ahora los valores 10-11 y 20-21.

En el caso que nos ocupa, se pretende realizar una lectura de la tensión en la fase 1 del CVM-MINI, que se encuentra en el byte 0 del registro y ocupa 2 bytes. La posición del byte 0 en formato hexadecimal se escribe de la misma forma que en decimal, pero si se hubiese querido leer la tensión de la fase 2, que se encuentra en la posición 0A-0B, se tendría que haber escrito en formato decimal.

El apartado *Poll rate* permite variar el tiempo de muestreo de los datos provenientes del analizador.

Por último, mediante la pestaña *server* se puede seleccionar el servidor modbustcp con el que se quiere establecer conexión. Desde esa misma pestaña se pueden definir los parámetros de este servidor, si todavía no están establecidos o editarlos en caso de que se precise. Solamente habría que hacer clic en el símbolo de editar que se sitúa a la derecha.

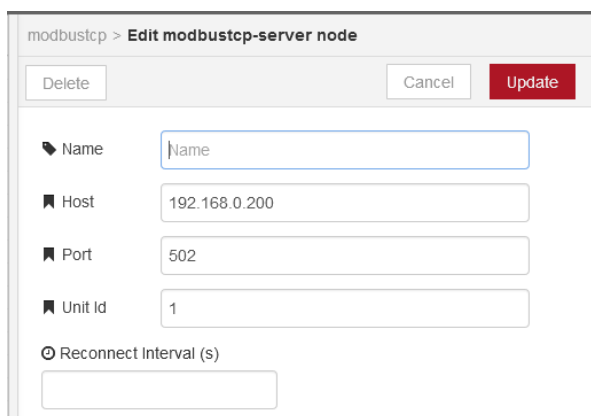


Ilustración 32. Configuración Modbustcp-server

La ilustración 32 muestra la interfaz de configuración de un nuevo servidor modbustcp en Node-RED. Solamente sería necesario definir la IP del dispositivo con el que se pretende establecer una conexión (IP: 192.168.0.200) y el puerto empleado, que en este caso será el 502, como se mencionó anteriormente. Además, se muestra la opción de darle un nombre al servidor (*Name*) para identificarlo con mayor facilidad o incluso definir el tiempo del intervalo (*Reconnect Interval*) en segundos para realizar la reconexión, en caso de que proceda.

Una vez configurado el nodo input de [modbustcp], solo habría que definir la finalidad de los datos que se reciben desde el dispositivo remoto del que se está recibiendo información. En este caso, la visualización de la información recibida se efectúa de la forma más simple, mediante el *debug* del propio node-red. Así, la disposición de nodos sería la que se muestra en la Ilustración 33.



Ilustración 33. Disposición de nodos para lectura en Modbustcp-server

Por último, se muestra en pantalla el valor de la tensión de la fase 1 (ver Ilustración 34).

```
7/6/2019 8:59:55 node: f55ae9f5.d31a18
msg.payload : buffer[2]
▶ [ 0, 229 ]
```

Ilustración 34. Valor de las tensiones leídas desde Node-RED

7.5.3.2 Servicio de escritura de datos

Los dispositivos que trabajan con Modbus suelen disponer de un espacio destinado a lectura y otro a escritura. En este caso, el analizador de redes no dispone de ningún registro de escritura ya que está pensado para captar información de la red eléctrica, no para recibirla. Es por ello, por lo que realizar una función de escritura no tendría sentido alguno en este caso. Sin embargo, a modo didáctico, se abordará el procedimiento a seguir si se quisiese enviar información a un dispositivo externo vía Modbus/TCP que soportarse servicio de escritura de datos.

En este apartado, se mostrará la realización de la escritura del valor de 200V de tensión en la fase 1 del CVM-MINI.

Si bien en el anterior apartado de escritura se utilizaba el nodo [modbustcp] input, en este se utilizará el nodo [modbustcp] output. Como se puede observar, este nodo tiene mucha semejanza con el nodo que se configuró anteriormente para realizar la función de escritura (ver Ilustración 35).

Ilustración 35. Configuración nodo output modbustcp

Primeramente y al igual que antes, se dispone de la opción de dar nombre al nodo (*Name*) que se está utilizando y categorizarlo mediante *Topic*. En la pestaña *Type* se encuentran las diferentes funciones de escritura que ofrece, que han sido explicados detalladamente en la *Tabla 5*:

- FC 5: Write Single Coil
- FC 6: Write Single Holding Register
- FC 15: Write Multiple Coils
- FC 16: Write Multiple Holding Registers

En el ejemplo que se está mostrando, si el analizador de redes permitiese realizar escrituras en sus registros, se seleccionaría el FC6, ya que se estaría escribiendo en uno de sus registros destinados a la escritura.

En *Address* se introduce la dirección a la que de la variable sobre la que se pretende realizar la escritura. Esta información suele ser entregada por el fabricante.

Para la configuración del servidor [modbustcp], el procedimiento seguido es el mismo que el mencionado anteriormente para la lectura. En caso de que ya se tuviera configurado el servidor solo habría que seleccionarlo, de no ser así, se tendría que configurar un nuevo servidor con sus respectivos parámetros y direcciones IP. Además, se deberá incluir el puerto con el valor 502 para lograr una comunicación efectiva.

Una vez configurado el nodo, faltaría por programar la parte correspondiente al envío de datos con el que se realizará la escritura. La forma más simple es la que se ha empleado en la ilustración siguiente, mediante un simple nodo [inject] donde únicamente se definirá el valor que se le pretende asignar. A continuación, en la ilustración 36 se muestra la disposición de los nodos que participan en la escritura:

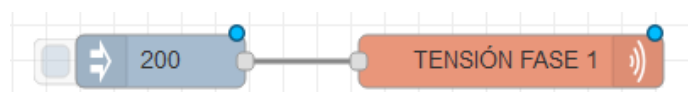


Ilustración 36. Disposición de nodos para escritura en Modbustcp-server

Por último, se observa en la ilustración 37 la estructura de un programa más completo, que utilizando como base lo mencionado anteriormente, se ha realizado las lecturas de las tres primeras fases, la frecuencia de la red y la escritura de distintos valores de tensión: 200, 300, 400 (suponiendo que se pudieran realizar).

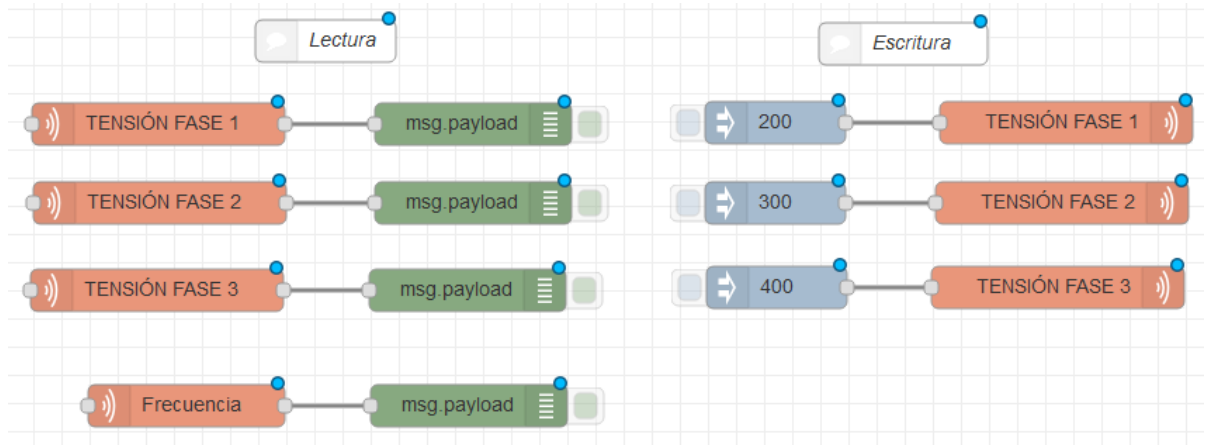


Ilustración 37. Disposición de nodos para lectura y escritura en Modbus/TCP

7.3 Protocolo OPC UA

OPC UA (*Open Platform Communications Unified Architecture*) es el estándar de interoperabilidad para el intercambio seguro y confiable de datos en el espacio de automatización industrial. OPC UA, a diferencia del estándar OPC, es independiente de la plataforma y asegura un flujo continuo de información entre dispositivos de distintos fabricantes. Una arquitectura OPC incluye uno o varios Clientes OPC y Servidores OPC comunicándose entre sí.

- El Servidor OPC hace de interfaz comunicando por un lado con una o más fuentes de datos (PLCs, DCSs, básculas, módulos I/O, controladores, etc.)
- Los clientes OPC son típicamente SCADAs, HMIs, generadores de informes, generadores de gráficos, aplicaciones de cálculos, etc.

Inicialmente, el estándar OPC se limitaba su uso al utilizarlo sobre el sistema operativo Windows. Sin embargo, con la introducción de arquitecturas orientadas a servicios en sistemas de fabricación, vinieron nuevos retos en la seguridad y el modelado de datos. Así, la Fundación OPC desarrolló las especificaciones OPC UA para satisfacer estas necesidades y, al mismo tiempo, proporcionó una arquitectura de plataforma abierta, que estaba preparada para el futuro, escalable y extensible.

OPC UA dispone de seguridad incorporada, un requisito que se ha vuelto cada vez más importante en entornos donde desde la red de la oficina se debe acceder a los datos de la planta. Además, OPC UA reúne las distintas especificaciones de la OPC clásica y ofrece un punto único de acceso a un sistema que proporciona acceso a los datos actuales, alarmas y sucesos, junto con el histórico de ambos. A diferencia de la OPC clásica, OPC UA proporciona un conjunto reducido y único de accesos genéricos de servicio a toda la información.

OPC UA funciona no solo en sistemas Linux, sino también en los controladores y dispositivos inteligentes que tienen sistemas de funcionamiento en tiempo real. Una de las principales ventajas de usar OPC UA respecto del OPC clásico, es que permite la modelización de la información y facilita muchas operaciones. Debido a que OPC UA define un sencillo conjunto de tipos básicos que pueden ampliarse con modelos de información.

Por otro lado, Node-RED permite la comunicación entre el dispositivo SIMATIC IOT2040 de SIEMENS y el PLC SIMATIC S7-1500 de SIEMENS. Node-RED se puede utilizar para proporcionar una solución fácil y confiable para procesar y monitorizar datos desde un servidor OPC UA.

7.3.1 Demostrador

En la ilustración 38 se presentan los dispositivos utilizados en el demostrador para probar y validar la comunicación entre el IOT2040 y el PLC utilizando diferentes protocolos. Como se puede observar, se ha dispuesto un equipo IOT2040, un PLC SIMATIC S7-1516, un HUB de Cisco de 8 puertos y un PC.

- El dispositivo IOT2040 tiene el puerto X2 conectado a Internet y el puerto X1 conectado al HUB. El puerto conectado a Internet se ha parametrizado de forma que asigne la dirección IP mediante el protocolo DHCP y al puerto conectado al HUB se le ha asignado la dirección IP 192.168.0.140 de forma manual. El software de este dispositivo es compatible con Yocto Linux, que proporciona un conjunto flexible de herramientas un espacio donde los desarrolladores integrados pueden compartir tecnologías para crear imágenes de Linux para dispositivos IOT o integrados.
- El dispositivo SIMATIC S7-1516, tiene conectado el puerto X1, con dirección IP 192.168.0.152, a uno de los puertos del HUB. El PLC utilizado en el trabajo no contiene ningún software instalado con anterioridad.

El PC tiene el puerto de la placa base conectada a Internet para la asignación dinámica de la IP. Se le ha instalado una tarjeta de red Ethernet, cuyo puerto se ha conectado al HUB, a la que se le ha asignado de forma manual la dirección IP 192.168.0.18.

El software instalado en este equipo contiene los siguientes programas:

- TIA Portal V15.1. Herramienta de ingeniería para el desarrollo de las aplicaciones para el PLC.
- PuTTY: Aplicación de código abierto que utiliza protocolos de red para habilitar una sesión remota en una computadora.
- Node-RED: Entorno gráfico de desarrollo de aplicaciones que se cargan y ejecutan en el equipo IOT2040.

El desarrollo de la aplicación va a tener 2 partes:

- Desarrollo de un proyecto de automatización en TIA Portal para configurar el PLC para la comunicación con el protocolo OPC-UA.
- Desarrollo de una aplicación Node-RED, que realice las operaciones de lectura y escritura en el PLC, una vez cargada en el equipo IOT2040.

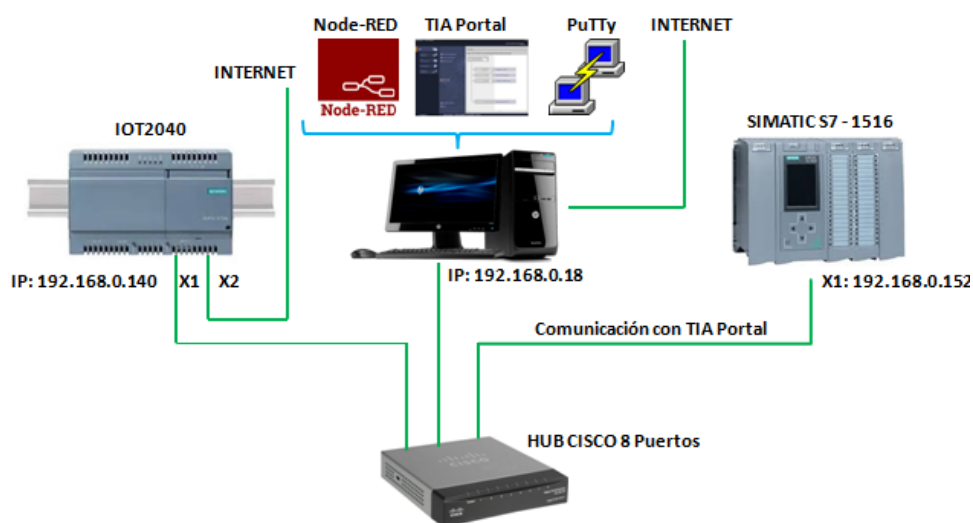


Ilustración 38. Demostrador para OPC UA.

7.3.2 Proyecto TIA Portal para el PLC

La primera tarea a desarrollar es un proyecto de automatización en TIA Portal que configurara el PLC con el servidor OPC UA. La funcionalidad de este proyecto de automatización será el control de tres escaleras mecánica descritas en el apartado 7.1.1. Por lo tanto, en este apartado se indicarán las configuraciones y parametrizaciones necesarias con TIA Portal para la configuración del servidor OPC UA integrado en un PLC S7-1516 (ver Ilustración 39). Asimismo, la configuración realizada es la mínima necesaria en el servidor OPC-UA para lograr la comunicación con un cliente OPC-UA proporcionado por Siemens funcionando en modo DA (Data Access). En la configuración se seguirán los siguientes pasos:

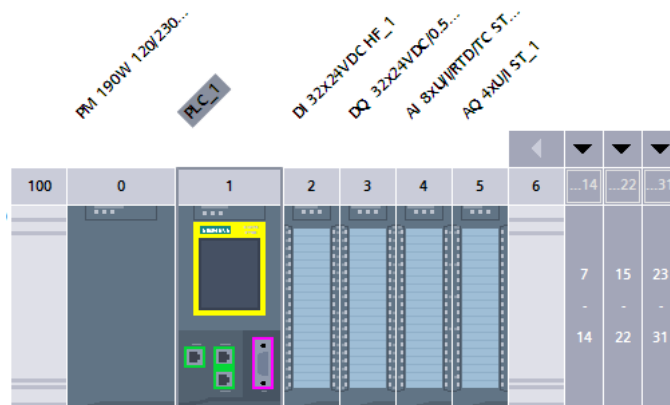


Ilustración 39. Configuración Hardware.

1. Se completa la configuración de Hardware del PLC respetando las referencias del hardware del equipo del laboratorio, asignando a la CPU la dirección IP 192.168.0.152 (ver Ilustración 40).

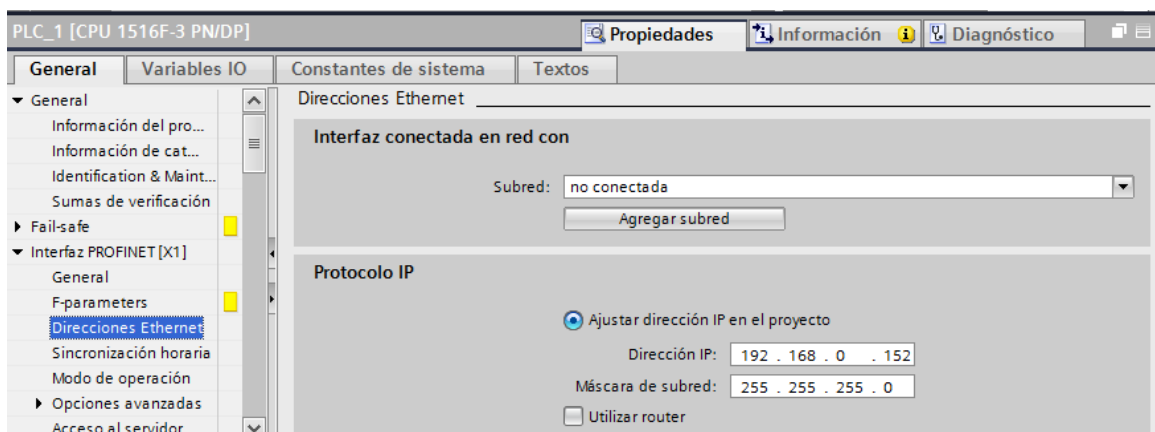


Ilustración 40. IP del PLC

2. Se selecciona el tipo de licencia del servidor OPC UA integrado en el PLC. En el menú de parametrización de la CPU, en el apartado “Licencias runtime”, se selecciona el sub apartado “OPC UA”. A continuación, se elige el servidor “SIMATIC OPC-UA S7-1500 medium”, que está disponible en este PLC. (Ver ilustración 41)

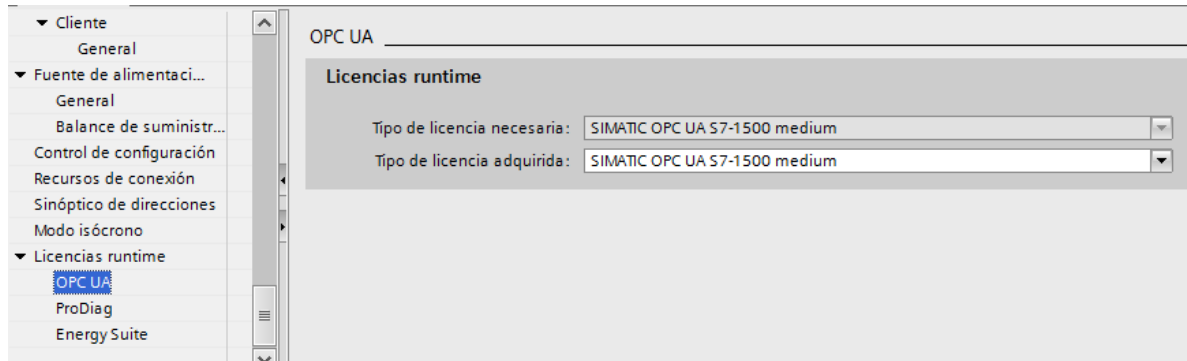


Ilustración 41. Licencia servidor OPC UA

3. Activación del servidor OPC UA. Esto se realiza en el apartado “OPC UA”, servidor, activando la opción correspondiente (ver Ilustración 42).

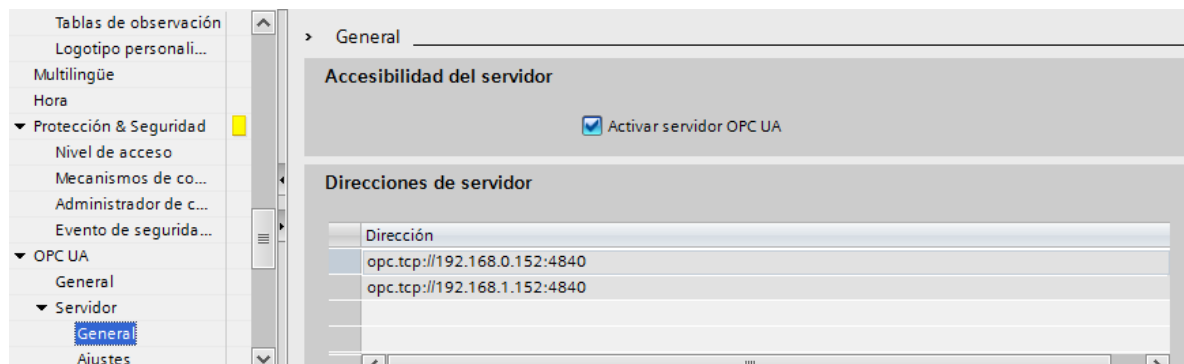


Ilustración 42. Activación servidor OPC UA

4. El apartado General permite modificar el nombre del servidor OPC UA. Es precisamente con este nombre con el que se podrá identificar el servidor al realizar la configuración del cliente y es muy importante cuando puedan coexistir diferentes servidores (ver Ilustración 43). Se optará por utilizar la primera de las direcciones IP, ya que la subred a la que va a ser conectada a de ser la misma que la del IOT2040.

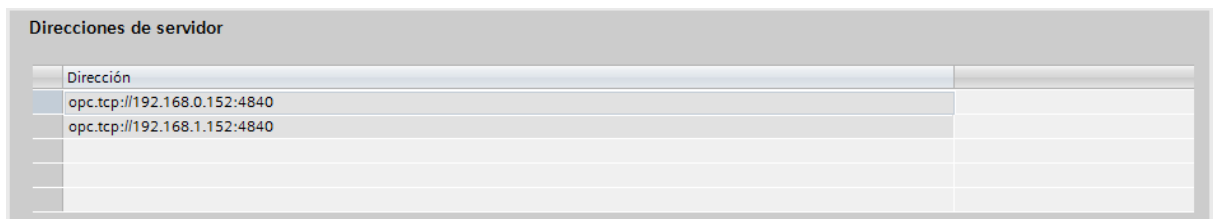


Ilustración 43. Activación de la dirección del OPC UA

Una vez realizado estos ajustes en el proyecto en TIA Portal, se carga el proyecto en el PLC y al arranca el PLC, el Servidor OPC UA ya está en condiciones de poder recibir peticiones de conexión desde Clientes OPC UA.

Para el servicio de lectura de datos desarrollado desde la aplicación Node-RED para el IOT2040, se deberá exportar el archivo XML OPC UA de la interfaz estándar del servidor SIMATIC para configurar correctamente los nodos desde Node-RED. A continuación se muestran los pasos necesarios para la exportación del archivo XML OPC UA.

1. Se accede al servidor OPC UA desde TIA Portal, en el se hará clic sobre “Exportar” y se obtendrá la siguiente pestaña (ver Ilustración 44). En dicha pantalla, se seleccionará “Exportar archivo XML OPC UA”.

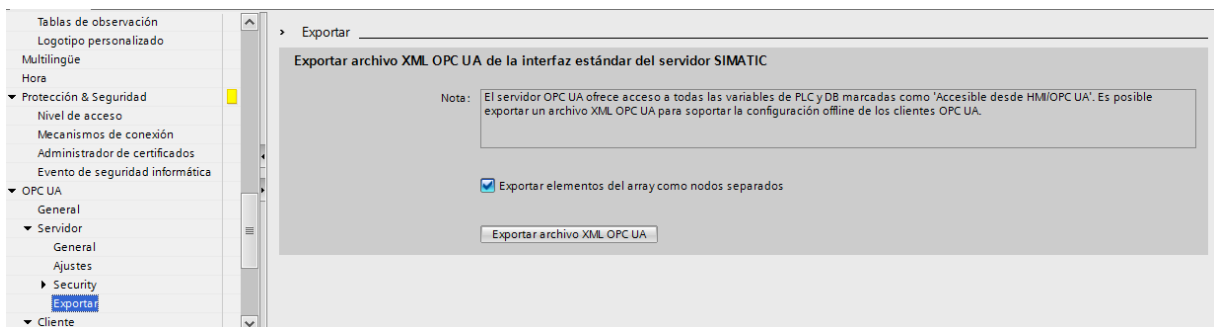


Ilustración 44. Exportación archivo XML OPC UA

2. Extraído el archivo, se muestra la opción de cambiar su nombre. Así, se creará un archivo, como el que se muestra a continuación, en el escritorio (ver Ilustración 45).

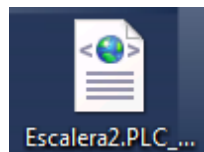


Ilustración 45. Archivo XML OPC UA en escritorio

3. Dentro de este archivo se encuentra un código de programa en lenguaje. En él, se buscará la variable que queramos leer desde Node-RED, por ejemplo, si se quisiera leer el número de personas que suben a través de la primera de las escalera (*Personas_E1*), se debería buscar el código que hace referencia a tal variable (ver Ilustración 46): *ns=3; s"DB2"." Person*_{s_E1}*". datatype=Unit16*. Siendo *datatype=Unit16* el tipo de dato a agregar. Cabe destacar que se ha escrito *Unit16*, puesto que la variable *Personas_E1* es de tipo Word.

```
- <UAVariable NodeId="ns=3;s="DB2"."Personas_E1" BrowseName="3:Personas_E1" ParentNodeId="ns=3;s="DB2"" DataType="WORD" AccessLevel="3">
  <DisplayName>Personas_E1</DisplayName>
```

Ilustración 46. Variable *Personas_E1*. Archivo XML OPC UA.

7.3.3 Desarrollo de la aplicación en Node-RED para IOT2040

Para poder empezar a trabajar con este protocolo se necesitará instalar el siguiente paquete de nodos *node-red-contrib-opcu* en el entorno Node-RED. La manera más sencilla de realizarlo es haciendo clic sobre “*Manage palette*” desde el menú de configuración en la esquina superior derecha de Node-RED (Ver Ilustración 47).

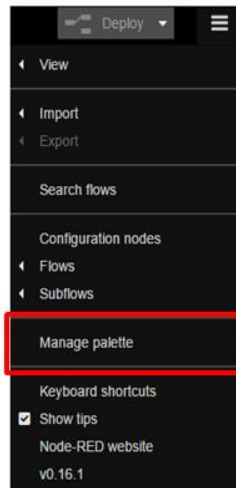


Ilustración 47. Menú de configuración de Node-RED

A continuación, dentro del menú que se muestra en la izquierda, se accede a la pestaña “*Install*”. Se busca el paquete de nodos denominado “*node-red-contrib-opcu*”, para así instalarlo (Ver Ilustración 48).

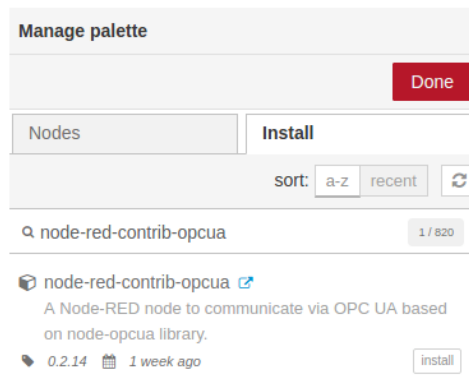


Ilustración 48. Instalación de nodo

Los diferentes nodos recién instalados en Node-RED se muestran justo al resto de nodos instalados.

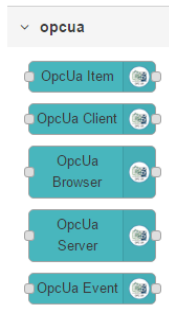


Ilustración 49. Nodos instalados para protocolo OPC UA

7.3.3.1 Servicio de lectura de datos

Para la lectura se comenzará arrastrando un nodo de inyección [Inject] delante del nodo [OpcUa Client], y se configurará el nodo de inyección (ver ilustración 50). El nodo [Inject] presenta diversos campos a rellenar y se completan de la siguiente forma:

- **Payload:** En este campo se introduce el tipo de dato que queremos leer en Node-RED. Pueden ser de diferentes tipos, por ejemplo: String, JSON, Number ...
- **Topic:** Este campo se rellenará con la información que obtenemos del TIA Portal (formato XML.). Debe seguir el formato que comienza con “ns=xxx”, y agregar el tipo de dato correspondiente a los datos que está obteniendo.
- **Repeat:** Sirve para definir el periodo con el que se realizará el servicio de lectura de datos. En este caso lo configuraremos de forma que los datos se actualicen con un periodo de 5 segundos.
- **Name:** En este campo se definirá el nombre con el que se desea nombrar el nodo.

Ilustración 50. Configuración nodo [Inject]

A continuación, se utilizará un nodo [OpcUa-Client](ver Ilustración 51). Se configurarán los siguientes apartados.

- **Endpoint:** Se introduce la dirección del servidor OPC UA descrito en la ilustración 38. Obsérvese que se debe introducir el puerto de OPC UA con valor 4840.
- **Action:** Se selecciona la acción que se desea realizar, en este caso READ, puesto que queremos realizar la operación de lectura.
- **Name:** Únicamente nos permitirá poner un nombre al nodo que se está utilizando.

La pestaña de configuración de este nodo es como se muestra en la ilustración 51.

Ilustración 51. Configuración nodo OpcUa-Client para lectura.

Segunda operación de lectura

Se procederá a realizar la misma configuración que ha hecho anteriormente, pero en vez de introducir los datos referidos al número de personas que entran en la escalera, se escribirá la referencia en el campo *Topic* relativo al tiempo de funcionamiento de la escalera (ver Ilustración 52).

Ilustración 52. Configuración nodo inject. Tiempo de funcionamiento

Los nodos se conectarán de la forma que se indica en la Ilustración 53 para poder realizar el servicio de lectura correctamente.

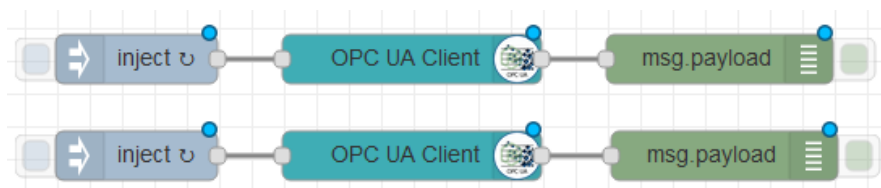


Ilustración 53. Disposición nodos lectura en OPC UA

Finalmente, se introducirán dos nodos [debug] para que se muestre en pantalla (ver Ilustración 54) los valores relativos al tiempo de funcionamiento y al número de personas que cuenta el PLC.

```

10/6/2019 10:00:58 node: 224e2000.6ae9d8
ns=3;s="DB2"."Personas_E1" : msg.payload : number
15

10/6/2019 10:00:59 node: 3531aab4.18ab2e
ns=3;s="DB2"."Tiempo_E1" : msg.payload : number
9

```

Ilustración 54. Número de personas y tiempo de funcionamiento en Node-RED

7.3.3.2 Servicio de escritura de datos

En este caso, se seguirá un procedimiento similar al del apartado anterior, pero se añadirá un nuevo nodo (*Item*) para la especificación del valor que se quiere escribir en el propio programa. La disposición de los nodos para la escritura del número de personas que acceden a la escalera se muestra en la Ilustración 55:

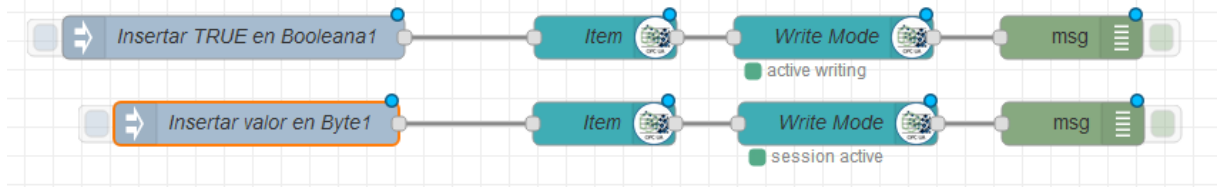


Ilustración 55. Disposición nodos para servicio de escritura en OPC UA

Se insertará el valor 10 desde un nodo tipo [inject] desde Node-RED (ver Ilustración 56).

Ilustración 56. Configuración nodo [Inject] para servicio de escritura en OPC UA

A continuación, se insertará el nodo [OpcUa Item] para definir la variable que se quiere escribir junto con el tipo de dato. Se seguirá el mismo procedimiento que se ha realizado para la lectura, es decir, en el archivo XML exportado se identifica la variable a escribir en el campo *Item*. La variable tipo Byte sobre la que se realizará el servicio de escritura es la variable *Byte1* del DB2, que se definió en el apartado 7.1.1. Por ello, al campo *Type* se le asignará la opción Byte (ver Ilustración 57).

Ilustración 57. Configuración nodo OpcUa-Item

Posteriormente, se configura el nodo [OpcUa-Client] para que realice el servicio de escritura. Su configuración es igual que el nodo configurado para la lectura, a excepción de la acción que realizará. Es decir, para la realización de la lectura se ha elegido la opción *READ*, mientras que para la escritura se elegirá *WRITE* (Ver Ilustración 58).

Ilustración 58. Configuración nodo OpcUa-Client para escritura.

Por último, para que los resultados se nos muestren en pantalla, se introducirá un nodo [debug].

En el DB2 del programa del PLC, se visualiza el valor introducido desde Node-RED, en valor hexadecimal. En este caso, como se ha introducido el valor 10 en formato decimal desde Node-RED, se visualiza el valor 0A en TIA Portal (ver Ilustración 59).

DB2				
Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación
Static				
Byte1	Byte	13.0	16#0	16#0A

Ilustración 59. Escritura 10 en DB2

En caso de que se quisiese otro tipo de variable (Bool), el procedimiento sería similar, únicamente se debería de cambiar el tipo de dato en los nodos [Inject] y [OpcUa Client] (ver Ilustración 60 y 61).

Ilustración 60. Configuración [Inject] para escritura OPC UA (Bool)

The image shows a dialog box titled "Edit OpcUa-Item node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons, there are four labeled input fields:

- Item:** A text box containing the string "ns=3;s='DB2'.\"Booleana1\"".
- Type:** A dropdown menu with "Boolean" selected.
- Value:** An empty text box.
- Name:** A text box containing the string "Item".

Ilustración 61. Configuración [OpcUa-Item] para escritura OPC UA (Bool)

En el DB2 del programa del PLC se visualiza el valor introducido desde Node-RED, en este caso el *Valor de observación* ha cambiado a TRUE (ver ilustración 62). Por lo tanto, podemos concluir que la configuración de nodos previamente realizada es correcta.

DB2					
	Nombre	Tipo de datos	Offset	Valor de arranq...	Valor de observación
Static					
Booleana 1		Bool	12.0	false	TRUE

Ilustración 62. Escritura TRUE en DB2

7.3.3.3 Servicio de suscripción de datos

El servicio de suscripción hace posible agrupar información del servidor. Este servicio únicamente funciona definiendo los denominados “elementos monitorizados” de los que el servidor administra la información con la que se trabaja y se realiza el envío al cliente, mientras el elemento monitorizado y la suscripción están permanentemente abiertos. Por lo tanto, la funcionalidad de suscribir consiste en monitorizar elementos y analizar su evolución temporal.

Para realizar esta función se han utilizado los mismos nodos en la misma secuencia a excepción del último de ellos, donde en lugar de usar dos nodos [Debug] se han introducido dos nodos [Chart] (ver Ilustración 63), para que así se puedan mostrar las variables que se están analizando: tiempo de funcionamiento y número de personas que acceden a la escalera. Todo ello se mostrará en un histórico.

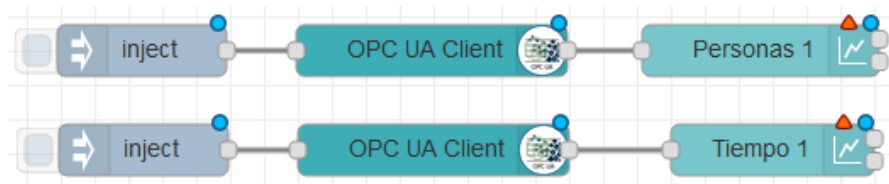


Ilustración 63. Disposición nodos para servicio de escritura en OPC UA

Asimismo, se tendrá que elegir la acción *SUBSCRIBE* del nodo [OpcUa-Client] (ver Ilustración 64), para poder realizar el servicio de suscripción de datos.

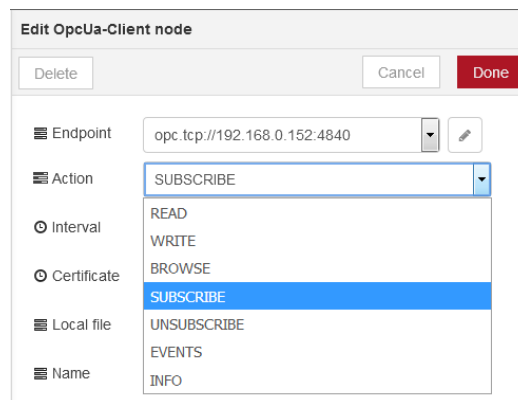


Ilustración 64. Edición nodo OpcUa-Client para suscripción

Además, se debe configurar el intervalo de actualización de datos en función de las características del proyecto. En este caso, se ha elegido un intervalo de 500 milisegundos (0,5 segundos) (ver Ilustración 65).

Ilustración 65. Edición nodo OpcUa-Client para suscripción

Los nodos [Chart] se pueden configurar de diferentes formas: Diferentes colores, interpolación, leyenda... En el caso que nos ocupa se ha optado por la opción más sencilla (ver Ilustración 66). Únicamente, se ha modificado la etiqueta que da nombre a las diferentes gráficas que posteriormente se verán, con los nombres *Personas 1* (Relativo al número de personas y a la escalera que estamos haciendo referencia).

Ilustración 66. Edición nodo OpcUa-Client para suscripción

Una vez realizado la disposición de nodos que se muestra en la ilustración 63, se deberá acceder a la siguiente dirección web <http://192.168.0.140:1880/ui/#/0>. En ella, se muestra la evolución temporal de las dos variables que estamos analizando (ver Ilustración 67)

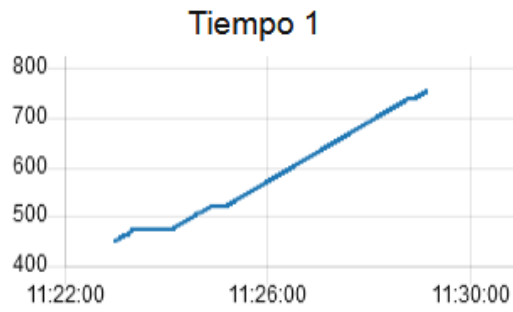


Ilustración 67. Graficas suscripción.

Se ha de tener en cuenta que la actualización de los gráficos solo se realiza ante cambios en las variables, es decir si una variable contiene un mismo valor a lo largo del tiempo, las gráficas no se actualizarán. Sin embargo, cuando las gráficas vuelvan a actualizarse se obtendrá una línea recta en las gráficas indicativo que no ha cambiado el valor a lo largo del tiempo.

8 Descripción de tareas. Diagrama de Gantt

Las tareas implicadas durante el presente Trabajo de Fin de Grado son las siguientes:

1. Análisis de alternativas

Se procederá al análisis de diferentes opciones disponibles en el mercado para la realización del proyecto, tanto a nivel de hardware, software y conectividad, ajustándose así a las necesidades del proyecto que se llevará a cabo. Esta fase tendrá una duración de 11 días.

2. Establecer objetivos

Esta es una etapa central en la realización del proyecto, donde se detalla el contenido del proyecto, su finalidad y el alcance del mismo. El establecimiento de objetivos es una labor esencial para poder llevar a cabo las etapas posteriores de forma rápida y ordenada. Esta fase del proyecto tendrá una duración aproximada de 6 días.

3. Estudio de viabilidad

Se deberá realizar este estudio, desde un punto de vista académico y económico, para saber si la viabilidad del proyecto es posible. Para ello, se analizará la documentación existente de los protocolos de comunicación y dispositivos a emplear. Además, se deberá tener en cuenta los conocimientos necesarios para llevar a cabo el proyecto. Esta fase del proyecto tendrá una duración aproximada de 6 días.

4. Creación del programa PLC

Se procederá a la creación del programa del PLC que será el controlará el funcionamiento de tres escaleras mecánicas, y tendrá un registro con el número de personas que se incorporan a las escaleras y el tiempo de funcionamiento de las mismas. Esta fase del proyecto tendrá una duración aproximada de 16 días.

5. Puesta en marcha del sistema de automatización

Primeramente, se descargarán los archivos del sistema operativo. Una vez descargados, se grabarán en la tarjeta Micro SD, para poder poner en marcha el sistema. Esta fase del proyecto tendrá una duración aproximada de 11 días.

6. Configuración de Node-RED para la transferencia de datos

Habiendo puesto en marcha el sistema de automatización a emplear, se procederá a la configuración de los diferentes nodos que ofrece Node-RED para poder realizar la lectura y escritura de los diferentes datos, a través de los protocolos OPC UA y Modbus/TCP. Esta fase del proyecto tendrá una duración aproximada de 41 días.

7. Comprobación de resultados

Una vez finalizado la configuración de Node-RED se hará la comprobación de resultados que han sido desarrolladas previamente. Durante esta etapa se espera que los valores de escritura y lectura que llevados a cabo con los diferentes protocolos correspondan con los valores que se pueden observar en los diferentes módulos de datos del programa PLC (y con el contador trifásico electrónico instalado). Esta fase del proyecto tendrá una duración aproximada de 11 días.

8. Redacción de la memoria

La última parte del Trabajo de Fin de Grado consistirá en la redacción de la memoria del mismo. En él se detallarán todos los elementos utilizados durante su realización, la información y la metodología empleada. Esta fase del proyecto tendrá una duración aproximada de 16 días.

El Trabajo de Fin de Grado lleva un total de 120 días laborables para su ejecución, lo cual supone 180 horas de trabajo considerando 1.5 horas por día de trabajo.

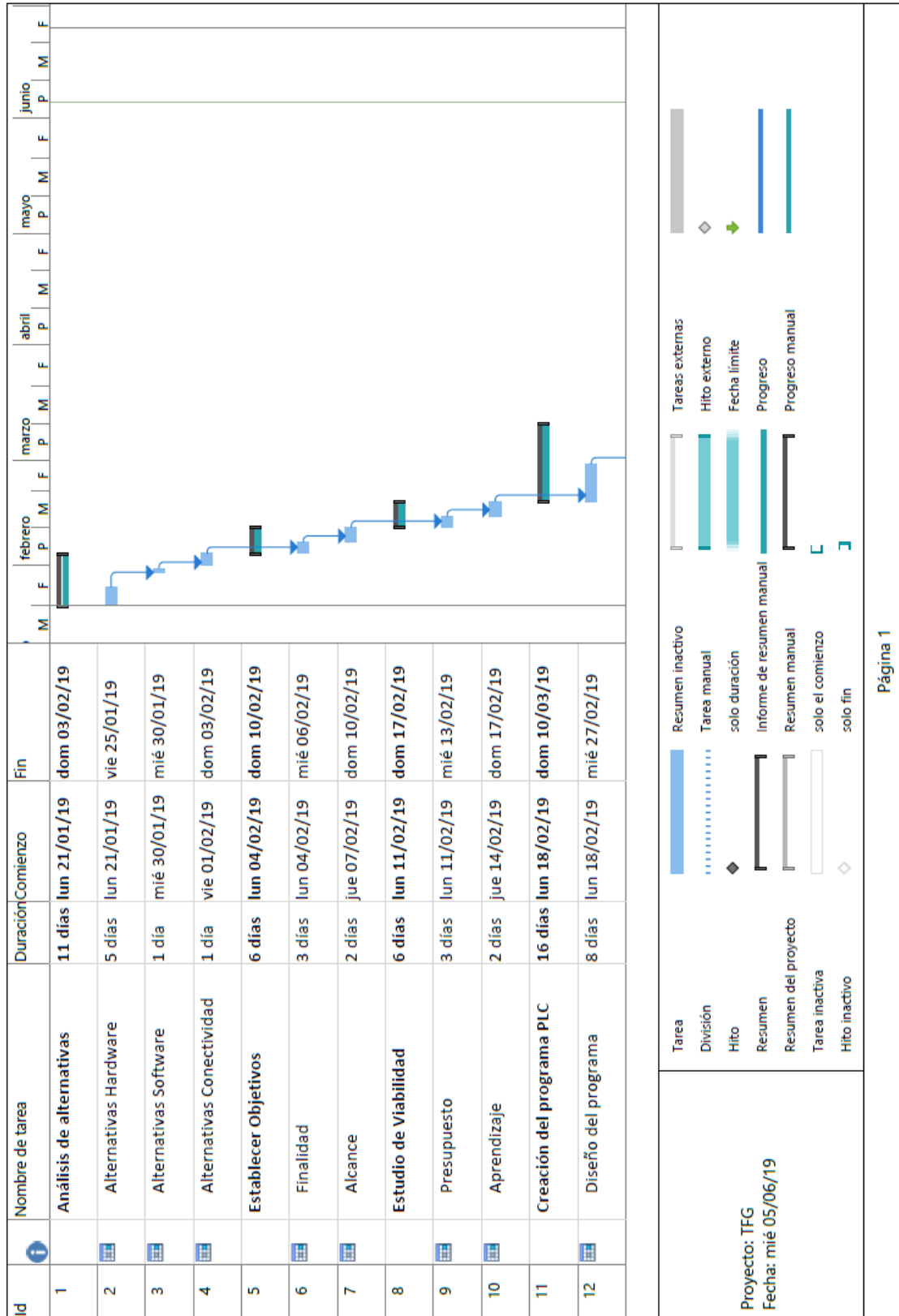


Ilustración 68. Diagrama de Gantt (1 de 3)

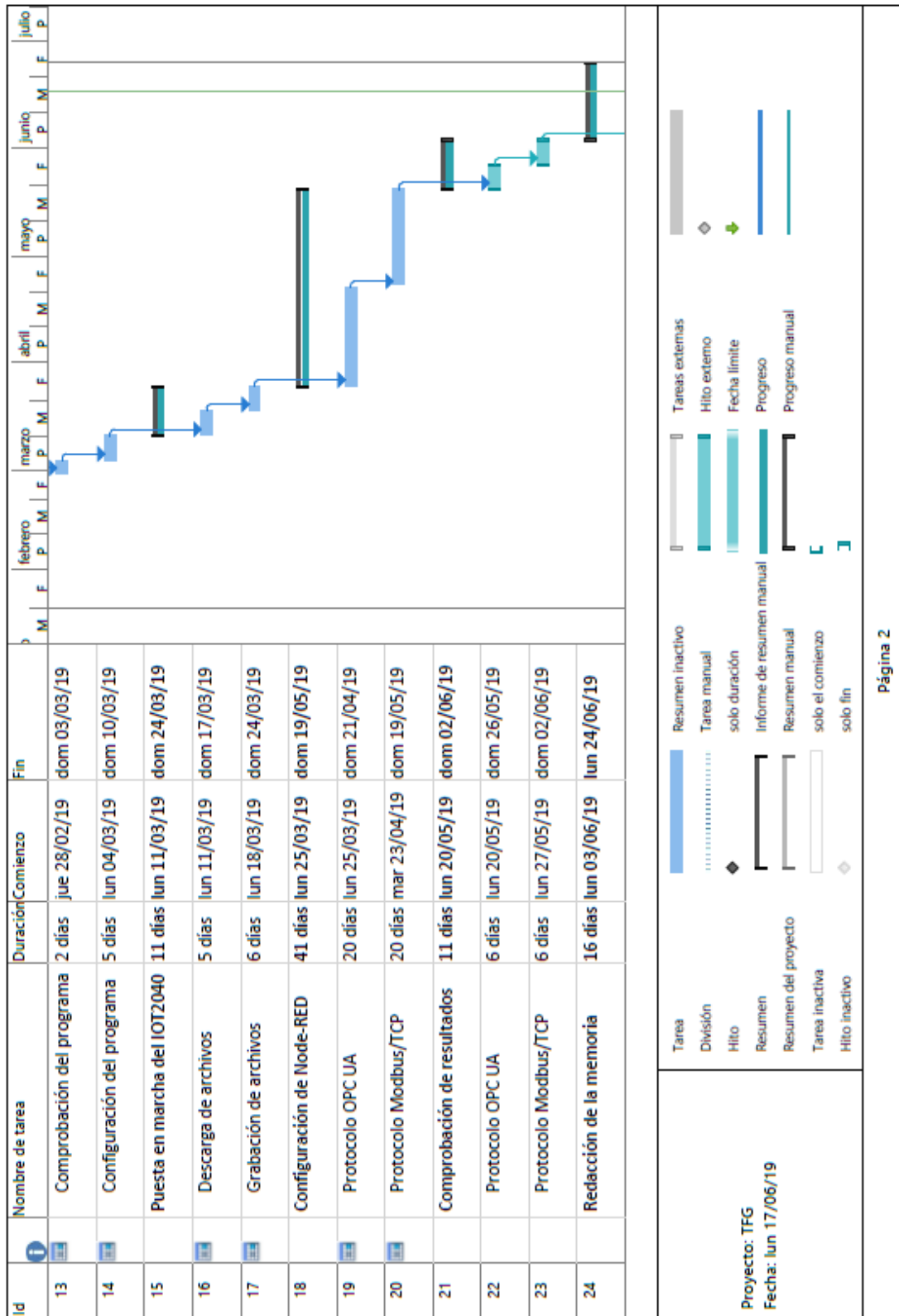


Ilustración 69. Diagrama de Gantt (2 de 3)

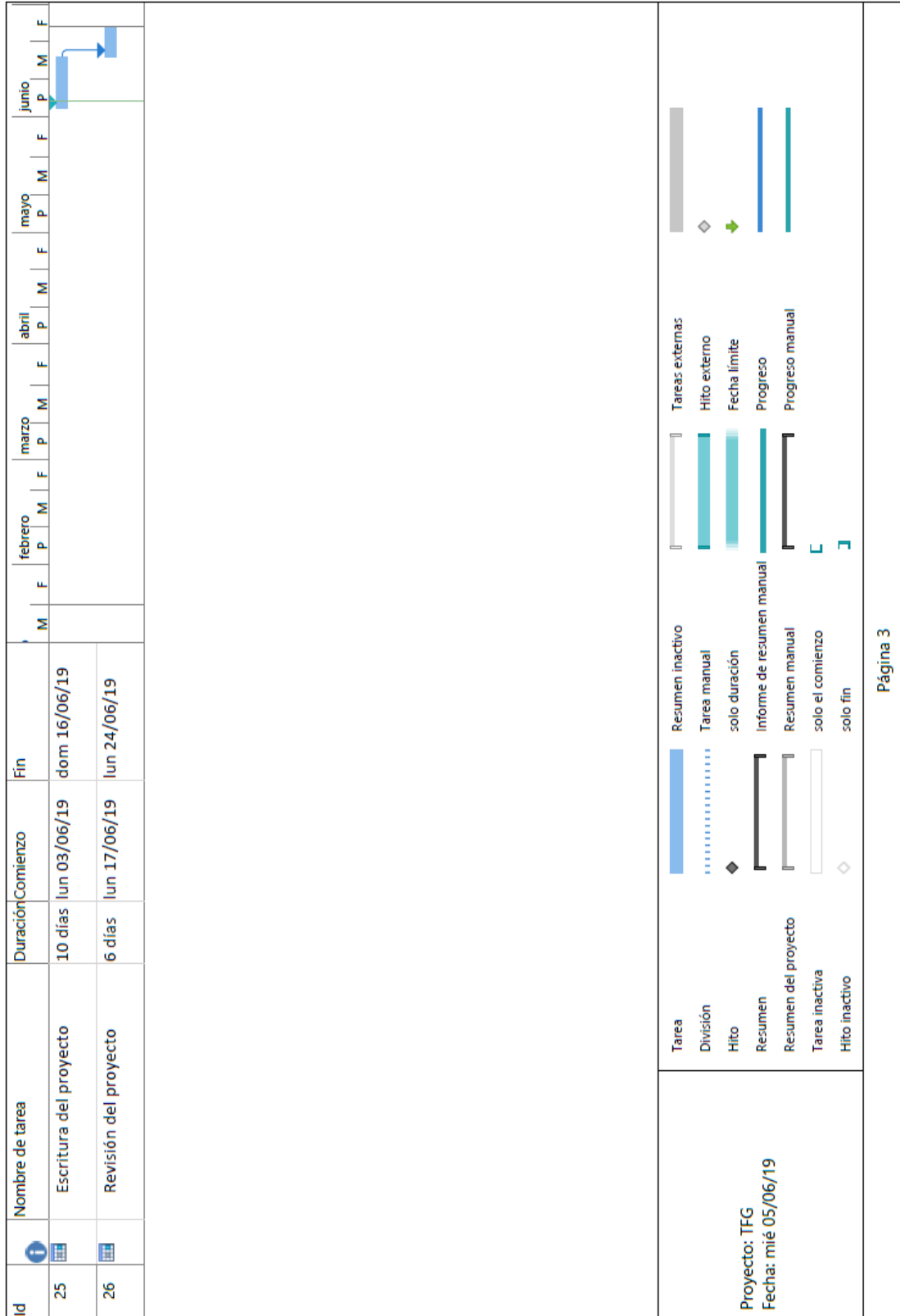


Ilustración 70. Diagrama de Gantt (3 de 3)

9 Aspectos económicos. Descripción del presupuesto

En la elaboración del proyecto, se ha decidido clasificarlo en tres partidas: horas internas, amortizaciones, y gastos. Asimismo, se ha añadido un resumen incluyendo en el mismo los costes indirectos que se derivan. En el importe final presupuestado se ha incluido una partida relativa a los costes indirectos asociados a la realización de proyecto como la electricidad o la conexión a Internet, entre otros.

Horas internas			
Concepto	Coste Unitario (€/h)	Número de unidades (h)	Coste(€)
Ingeniero Superior	30	180	5400
TOTAL			5400

Tabla 7. Horas internas

Amortizaciones				
Concepto	Precio de adquisición (€/h)	Vida útil (años)	Utilización(h)	Coste (€)
Ordenador	800	5 (9125h/año)	180	15,78
TOTAL				15,78

Tabla 8. Presupuesto. Amortizaciones

Gastos			
Concepto	Coste Unitario(€/ud)	Número de unidades	Coste (€)
Entrenador SIMATIC S7-1516	3275	1	3275
Switch 8 puertos	23,45	1	23,45
Arduino Ethernet Shield	12,96	1	12,96
SIEMENS SIMATIC IOT2040	222,6	1	222,6
Cables Ethernet RJ45	9,31	2	18,62
TOTAL			3729.33

Tabla 9. Presupuesto. Gastos

Resumen del presupuesto	
Concepto	Total (€)
Horas internas	5400
Amortizaciones	15,78
Gastos	3729,33
SUBTOTAL	9145,11
Costes Indirectos (5%)	457,26
TOTAL	9602,37

Tabla 10. Presupuesto. Resumen

En la Ilustración 71 se muestra la ponderación de cada partida del presupuesto:

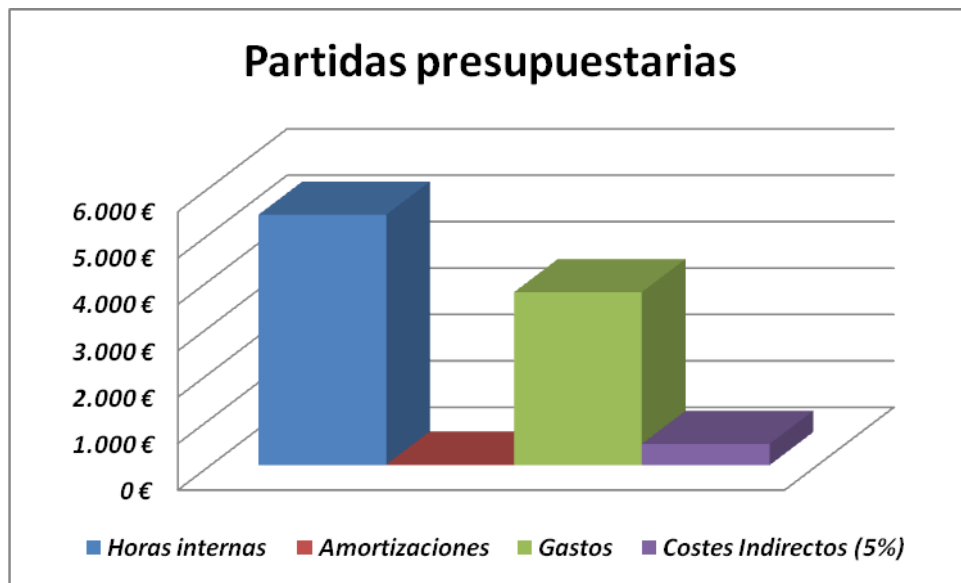


Ilustración 71. Partidas Presupuestarias.

Tal y como se puede observar en la ilustración anterior, las mayores partidas son debidas al pago de horas internas por parte del ingeniero superior, que se necesita para llevar a cabo el proyecto, y al gasto de los equipos necesarios para su realización. Entre ellos, cabe destacar el gasto que se debe realizar debido al Entrenador SIMATIC S7-1516 que tiene un precio de 3275€. El dispositivo SIMATIC IOT2040 tiene un precio moderadamente bajo, tal y como se ha comentado en apartados anteriores.

Por otro lado, se observa que las amortizaciones y los costes indirectos (5% del presupuesto) únicamente suponen un gasto muy pequeño en comparación con las partidas mencionadas en el párrafo anterior.

Finalmente, la inversión inicial necesaria para la realización de este proyecto es de 9602,37€.

10 Conclusiones

En este documento, se ha detallado el proceso necesario para intercambiar datos desde la plataforma SIMATIC IOT2040 de Siemens con los protocolos industriales: Modbus/TCP y OPC UA. Así se posibilita el acceso a datos de cualquier proceso, acercando el concepto de la industria tradicional y actual hacia la cuarta revolución industrial, conocida con el nombre de *Industria 4.0*, ya que el objetivo de un equipo como el IOT2040 es adquirir datos de los controladores (PLCs) y subirlos a la nube (Cloud).

El uso de protocolos de comunicación industrial ha permitido incrementar la flexibilidad del sistema de manera que se pueda cumplir con los requerimientos para adaptarlo a industrias que incorporan estas tecnologías.

El bajo coste del sistema, y su diseño para el ámbito industrial, hacen que la implementación del dispositivo SIMATIC IOT2040 en Node-RED sea idónea para la aplicación de diferentes protocolos industriales como han sido OPC UA y Modbus/TCP. Cabe destacar, que el bajo coste de este sistema no solo se ha debido precisamente al bajo precio del dispositivo, sino también a la no necesidad de licencias para su puesta en marcha.

Un sistema de adquisición de datos de este tipo, sirve para negocios a nivel industrial y además, resulta de gran utilidad en otros campos como podría ser la docencia. El abaratamiento de los ordenadores y el desarrollo de tecnologías de redes que trabajan con una mayor velocidad, permite acelerar su funcionamiento y su implantación para cualquier otra aplicación donde haya que adquirir datos con diversos protocolos.

Por último, para seguir siendo competitivas en un entorno globalizado, las empresas necesitan evolucionar constantemente, mediante sistemas de producción más eficientes y poder adaptarse a las demandas cambiantes de los mercados. Por tanto, deberán incorporar plataformas modulares como SIMATIC IOT2040 de Siemens, la cual se ha utilizado en el presente Trabajo de Fin de Grado para facilitar la incorporación a la *Industria 4.0*.

11 Bibliografía

- [1] Siemens. *SIMATIC IOT2040. Primera instalación y primer programa en node-red.*
<https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/IoT2040-Primeros-pasos-programacion.pdf>
- [2] Siemens. SIMATIC IOT2000 support forums.
<https://support.industry.siemens.com/tf/ww/en/conf/60/>
- [3] Node-RED. Library
https://flows.nodered.org/?num_pages=1
- [4] Trabajo de Fin de Grado. Puesta en marcha de un sistema de automatización SIEMENS SIMATIC IOT2040
<https://addi.ehu.es/bitstream/handle/10810/26011/TRABAJO%20DE%20FIN%20DE%20GRADO3.pdf?sequence=1>
- [5] PDAControl. Instalación nodo Modbus TCP en Node-red
<http://pdacontroles.com/instalacion-nodo-modbus-tcp-en-node-red/>
- [6] Sensetecnic Developer – Hosted Node-RED. Tutorial: Using OPC-UA with FRED (Cloud Node-RED)
<http://developers.sensetecnic.com/article/tutorial-using-opc-ua-with-fred/>
- [7] Siemens. Getting started with SIMATIC IOT2000.
<https://support.industry.siemens.com/tf/ww/en/posts/getting-started-with-simaticiot2000/155643>
- [8] Siemens. Setting up the SIMATIC IOT2000
<https://support.industry.siemens.com/tf/ww/en/posts/setting-up-the-simaticiot2000/155642>
- [9] Wikipedia. Revolución industrial etapa cuatro.
https://es.wikipedia.org/wiki/Revoluci%C3%B3n_industrial_etapa_cuatro
- [10] Wolfgang Mahnke, Stefan-Helmut Leitner, Matthias Damm (2009). OPC Unified Architecture. Springer